

BASIC PROGRAMMING C

Editors:

Masitah Ghazali

Dayang Norhayati Abang Jawawi

Edited with permission from the original material by Norazah Yusof, Dayang Norhayati Abang Jawawi, Noraniah Mohd Yassin, Radziah Mohamad, Paridah Samsuri and Rohayanti Hassan, *Programming Technique I (C++)*, First Edition, 2007.

Copyright © 2010

All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, or transmitted in any form or by any means, electronics, mechanical, photocopying or otherwise, without the prior written permission of the authors.

Disclaimer

This workbook and its contents are intended solely for the use of the subject Basic Programming C to be taught at Fakulti Sains Komputer dan Sistem Maklumat, Universiti Teknologi Malaysia, Skudai. The algorithms and programs in this workbook have been included for their instructional value. They have been tested but are not guaranteed for any particular purpose. The authors do not offer any warranties or representations nor do they accept any liabilities with respect to their use. The authors reserve the right to revise this publication and make changes from time to time in its content without notice.

**First Edition
November 2010**

CONTENTS

LAB	TOPIC	PAGE
	INTRODUCTION	v
	TERMS	vii
1	RUNNING THE C PROGRAM	1
	Objectives For Students	1
	Assumptions	1
	Lab Exercises :	
	Exercise 1	1
	Exercise 2	10
	Exercise 3	12
2	DESIGNING PSEUDO CODE AND FLOW CHART USING MOROSOFT WORD	16
	Objectives For Students	16
	Assumptions	16
	Lab Exercises :	
	Exercise 1	16
	Exercise 2	19
	Exercise 3	22
	Exercise 4	25
	Exercise 5	27
	Exercise 6	29
3	DESIGN PSEUDO CODE / FLOW CHART USING MICROSOFT	30
	Objectives For Students	30
	Assumptions	30
	Lab Exercises :	
	Exercise 1	30
	Exercise 2	35
	Exercise 3	38
4	CONVERTING ALGORITHM TO C CODE	40
	Objectives For Students	40
	Assumptions	40
	Lab Exercise :	
	Exercise 1	40
	Exercise 2	41
	Exercise 3	42

5	WORKING WITH DATA	44
	Objectives For Students	44
	Assumptions	44
	Lab Exercise :	
	Exercise 1	44
	Exercise 2	45
	Exercise 3	46
	Exercise 4	47
	Exercise 5	48
	Exercise 6	49
6	INPUT & OUTPUT	50
	Objectives For Students	50
	Assumptions	50
	Lab Exercise :	
	Exercise 1	50
	Exercise 2	51
	Exercise 3	52
	Exercise 4	53
	Exercise 5	54
	Exercise 6	55
7	SELECTION OR DECISION	57
	Objectives For Students	57
	Assumptions	57
	Lab Exercise :	
	Exercise 1	57
	Exercise 2	58
	Exercise 3	60
	Exercise 4	61
	Exercise 5	63
	Exercise 6	64
	Exercise 7	65
8	LOOP STATEMENTS	68
	Objectives For Students	68
	Assumptions	68
	Lab Exercise :	
	Exercise 1	69
	Exercise 2	70
	Exercise 3	72
	Exercise 4	73
	Exercise 5	74
	Exercise 6	75
	Exercise 7	77
	Exercise 8	78
	Exercise 9	79

	Exercise 10	79
9	NESTED LOOP	82
	Objectives For Students	82
	Assumptions	82
	Lab Exercise :	
	Exercise 1	82
	Exercise 2	84
	Exercise 3	85
	Exercise 4	86
10	PREDEFINED FUNCTIONS	88
	Objectives For Students	88
	Assumptions	88
	Lab Exercise :	
	Exercise 1	88
	Exercise 2	89
	Exercise 3	90
	Exercise 4	90
11	USER-DEFINED FUNCTION	91
	Objectives For Students	91
	Assumptions	91
	Lab Exercise :	
	Exercise 1	91
	Exercise 2	92
	Exercise 3	96
	Exercise 4	98
	Exercise 5	101
	Exercise 6	102
12	ONE-DIMENSIONAL ARRAY	105
	Objectives For Students	105
	Assumptions	105
	Lab Exercises :	
	Exercise 1	105
	Exercise 2	109
	Exercise 3	111
13	TWO-DIMENSIONAL ARRAY	115
	Objectives For Students	115
	Assumptions	115
	Lab Exercises :	
	Exercise 1	115
	Exercise 2	118

INTRODUCTION

OBJECTIVE

The laboratory for Basic Programming C is intended to introduce the students to programming as a practical problem-solving activity, in which they will have the opportunity to develop their skills.

CONTENT OF THE LABORATORY

Of many possible alternatives, you will learn to program in C, the high-level programming language. This language is popular because of its portability and is used for developing a wide variety of computer applications.

Programming is an activity that requires problem solving skill. This activity requires logical thinking, attention to detail, careful planning and elements of creativity and arts. Basic Programming C subject needs the students to put in some 200 hours' effort. 26 of these hours will be lectures and 26 in laboratory or tutorial. That leaves 148 hours split between reading the textbook and lectures notes, working on programming assignments in their own time, and preparing for the practical and written examinations. After the fourth week or so, they need to work in developing programs independently. In order to ensure their skill at an acceptable stage a short assessment on problem solving will be given to them after the fourth week. It is expected that the students can achieve certain level of problem solving skill before they can pursue to more complex programming techniques. Special workshop will be set for the students who cannot meet the standard.

LABORATORY RESOURCES

The FSKSM laboratories are equipped with the C programming language which Borland C++ software of version 5.02 is used for developing programs. This Borland C++ version 5.02 software have integrated development environment (IDE), which consists of a text editor, compiler, debugger, and other utilities integrated into one package with a single set of menus. Preprocessing, compiling, linking, and even executing a program is done with a single click of a button, or by selecting a single item from a menu.

Besides the programming language software, some of the programming materials of this work book can also be accessed from the UTM's e-learning system. Students who have problem downloading the materials may need to consult his/her lecturer or teaching assistant for further assistance.

LABORATORY SESSION AND ASSESSMENT

The attendance of the laboratory session is compulsory. If a student missed a session, he/she will be expected to make up the work in his/her own time.

The Basic Programming C course is assessed in the following way:

○ Mid Semester Exam	20%
○ Assignments (at least 3)	25%
○ Quiz (at least 3)	15%
○ Laboratory work (at least 4)	10%
○ Final Exam	30%
TOTAL	100%

For the Laboratory works, the students are expected to attempt a specified number of small exercises in each lab. Do note that, some of the lab exercises are expected to be submitted at the end of the session through UTM e-learning system. Normally, no allowance will be made for absence. At least four of the lab works (as assigned by the lecturer) will be assessed and counted 10% to the overall marks.

Further details about the assessment will be announced by the lecturer in the class.

ACKNOWLEDGEMENT

The authors gratefully acknowledge all contributors to this workbook in term of ideas, selected questions and comments. Thanks to the Software Engineering Department and Faculty of Computer Science and Information Systems for their support in preparing this material.

REFERENCES

1. Tony Gaddis, Barret Krupnow, (2007), *Starting out with C++*, 5th edition Brief version. Pearson Addison-Wesley.
2. W. Savitch, (2006), *Problem Solving with C++*. 6th Edition. Addison Wesley.
3. H.M Deitel, P.J Deitel, (2005), *C++ How to Program*. 5th edition. Prentice Hall.
4. Behrouz A.Forouzan, Richard F.Gilberg, (2004) *Computer Science: A Structured Approach Using C++*, 2nd edition. Brooks/Cole Thomson Learning.

TERMS

ENGLISH	MALAY
Algorithm	<i>Algoritma</i>
Argument	<i>Argumen</i>
Arithmetic operation	<i>Operasi aritmetik</i>
Array	<i>Tatasusunan</i>
Assembly stage	<i>Peringkat penghimpun</i>
Assign	<i>Umpuk</i>
Backup	<i>Bakup</i>
Boot	<i>But atau memulakan</i>
Buffer	<i>Penimbal</i>
Character	<i>Aksara</i>
Code segment	<i>Segmen aturcara</i>
Column	<i>Lajur</i>
Command	<i>Arahan</i>
Compile	<i>Kompil</i>
Constant	<i>Pemalar</i>
Control structure	<i>Struktur kawalan</i>
Copy	<i>Salin</i>
Current directory	<i>Direktori semasa</i>
Cursor	<i>Kursor</i>
Cut	<i>Potong</i>
Decision	<i>Keputusan</i>
Declare	<i>Isytihar</i>
Decrement	<i>Susutan</i>
Define	<i>Takrif</i>
Demonstrate	<i>Menunjuk cara</i>
Device	<i>Peranti</i>
Directory	<i>Direktori</i>
Directory tree	<i>Pepohon direktori</i>
Disk	<i>Cakera</i>
Distance	<i>Jarak</i>
Download	<i>Muat turun</i>
Drive C	<i>Pemacu C</i>
Editor	<i>Penyunting</i>
Error	<i>Ralat</i>
Executable file	<i>Fail boleh-laksana</i>
Expression	<i>Ungkapan</i>
Extension	<i>Sambungan</i>
Field	<i>Medan</i>

File	<i>Fail</i>
File operation	<i>Operasi fail</i>
File stream object	<i>Objek aliran fail</i>
File system path	<i>Laluan fail sistem</i>
Floppy drive	<i>Pemacu liut</i>
Flowchart	<i>Carta alir</i>
Folder	<i>Direktori</i>
Function	<i>Fungsi</i>
Function prototype	<i>Prototaip fungsi</i>
Global variable	<i>Pembolehubah sejagat</i>
Hardware	<i>Perkakasan</i>
Increment	<i>Tokokan</i>
Index	<i>Indeks</i>
Inline function	<i>Fungsi dalam baris</i>
Invalid	<i>Tidak sah</i>
Keyboard	<i>Papan kekunci</i>
Load	<i>Muat</i>
Local variable	<i>Pembolehubah setempat</i>
Logic error	<i>Ralat logik</i>
Loop	<i>Gelung</i>
Main function	<i>Fungsi utama</i>
Memory layout	<i>Hampan ingatan</i>
Menu bar	<i>Bar menu</i>
Multiple	<i>Berbilang</i>
Nested	<i>Tersarang</i>
Nested loop	<i>Gelung tersarang</i>
One dimensional array	<i>Tatasusunan satu dimensi</i>
Open source	<i>Sumber terbuka</i>
Operand	<i>Operan</i>
Operating system	<i>Sistem pengoperasian</i>
Overloaded function	<i>Fungsi sarat</i>
Parameter	<i>Parameter</i>
Pass by reference	<i>Hantar secara rujukan</i>
Pass by value	<i>Hantar secara nilai</i>
Paste	<i>Tampal</i>
Pen drive	<i>Pemacu pen</i>
Personal Computer (PC)	<i>Komputer peribadi</i>
Pointer	<i>Penuding</i>
Predefined function	<i>Fungsi takrifan piawai</i>
Pre-processor	<i>Prapemproses</i>
Program	<i>Aturcara</i>
Program header	<i>Kepala aturcara</i>
Program segment	<i>Segmen aturcara</i>
Prompt	<i>Penggesa</i>
Pseudo code	<i>Kod sudo</i>

Random Access Memory (RAM)	<i>Ingatan capaian rawak</i>
Read	<i>Baca</i>
Read Only Memory (ROM)	<i>Ingatan baca sahaja</i>
Repetition	<i>Pengulangan</i>
Return	<i>Memulangkan</i>
Row	<i>Baris</i>
Run	<i>Laksanakan</i>
Runtime error	<i>Ralat masa larian</i>
Save	<i>Simpan</i>
Search text	<i>Carian teks</i>
Selection	<i>Pilihan</i>
Semantic	<i>Semantik</i>
Shortcuts	<i>Capaian ringkas</i>
Software	<i>Perisian</i>
Statement	<i>Pernyataan</i>
Stream	<i>Aliran</i>
String	<i>Rentetan</i>
Structured chart	<i>Carta berstruktur</i>
Structured data	<i>Data berstruktur</i>
Subscript	<i>Subskrip</i>
Syntax	<i>Sintaksis</i>
Syntax error	<i>Ralat sintaks</i>
Text file	<i>Fail teks</i>
Trace	<i>Jejak</i>
Two-dimensional array	<i>Tatasusunan dua dimensi</i>
Upload	<i>Muat naik</i>
User defined function	<i>Fungsi takrifan pengguna</i>
Valid	<i>Sah</i>
Variable	<i>Pembolehubah</i>
Velocity	<i>Halaju</i>
Write	<i>Tulis</i>

LAB 1: RUNNING THE C PROGRAM

OBJECTIVES FOR THE STUDENT

1. To use the Borland C++ software version 5.02 to enter C source code, compile, link and run the C program.
[Untuk menggunakan perisian Borland C++ versi 5.02 untuk memasukkan kod sumber C, mengkompil, memaut, dan melaksanakan aturcara C]
2. To identify the C program structure.
[Untuk mengenal pasti struktur aturcara C]
3. To load, edit and recompile the C program.
[Untuk memuat, mengedit dan mengkompil semula aturcara C]

ASSUMPTIONS

1. Students should have a basic understanding of computer hardware and software.
[Pelajar sepatutnya mempunyai pengetahuan asas dalam perkakasan dan perisian komputer]

LAB EXERCISES

EXERCISE 1:

[LATIHAN 1]

1. Starting the Computer.
[Menghidupkan komputer.]
 - i. Booting is the term used to start up the PC (personal computer) by turning on the power. The boot instructions are placed in ROM by the computer manufacturer and cannot be altered by the user. These instructions first perform a quick check of the basic computer hardware.
[But (booting) ialah istilah yang digunakan untuk menghidupkan komputer peribadi (PC) dengan cara memasang kuasa elektrik. Arahan but terletak dalam ROM oleh pembuat komputer dan arahan ini tidak boleh diubahsuai oleh pengguna. Pada permulaannya, arahan ini melaksanakan pemeriksaan pantas terhadap perkakasan komputer yang asas.]
 - ii. Next, the ROM BIOS checks to see if there is a system disk in drive A. If it is not found, it then checks to see if the system files are on drive C.

Once the system disk is found, a short set of instructions called the boot record is loaded into RAM. The boot record contains the information necessary to load the I/O handler into RAM.

[Seterusnya, ROM BIOS memeriksa sama ada terdapat cakera sistem pada pemacu A. Jika ia tidak dijumpai, ia akan memeriksa sama ada fail sistem berada di pemacu C. Sebaik sahaja cakera sistem ditemui, satu set arahan pendek yang dinamakan rekod but dimuatkan ke dalam RAM. But rekod ini mengandungi maklumat penting untuk memuatkan pengendali I/O ke dalam RAM.]

- iii. After the I/O handler is loaded, the system disk is then checked for the presence of the operating system files (i.e. CONFIG.SYS, COMMAND.COM and AUTOEXEC.BAT) and loads them into RAM.
[Setelah pengendali I/O dimuatkan, cakera system akan diperiksa bagi kehadiran fail-fail sistem pengoperasian (seperti CONFIG.SYS, COMMAND.COM and AUTOEXEC.BAT) dan memuatkannya ke dalam RAM.]
 - iv. The PC that uses Windows 98, 2000 or XP, needs the user to enter a password in order to access any program/files in the computer. For the computer in the FSKSM lab, please get the password from the technician in charge.
[Komputer peribadi yang menggunakan sistem pengoperasian Windows 98, 2000 atau XP, memerlukan penggunaannya memasukkan kata laluan tertentu sebelum boleh mencapai mana-mana aturcara/fail yang terdapat di dalam komputer tersebut. Bagi komputer di makmal FSKSM, sila dapatkan kata laluan daripada juruteknik yang bertugas.]
 - v. To boot and enter the password:
[Untuk but dan memasukkan kata laluan:]
 - vi. Turn on all the switches to start your PC.
[Pasang semua suis untuk menghidupkan komputer anda]
 - vii. Wait until a dialog box appears for you to enter the password.
[Tunggu sehingga kekotak dialog dipaparkan untuk anda masukkan kata laluan.]
 - viii. Enter the appropriate password.
[Taipkan kata laluan yang bersesuaian.]
 - ix. Press the Enter key.
[Tekan kekunci Enter.]
2. Loading the Borland C++ software.
[Memuatkan perisian Borland C++.]
- i. All programs to be run by the computer need to be loaded into the computer's memory, i.e RAM.

[Semua aturcara yang hendak dilaksanakan oleh komputer mesti dimuatkan ke dalam ingatan utama komputer, iaitu RAM.]

- ii. To load the Borland C++ software:
[Untuk memuatkan perisian Borland C++:]
 - a. Click on the Start menu that is located at the bottom left of your screen.
[Klik pada menu Start yang berada di bahagian bawah sebelah kiri skrin.]
 - b. Choose All Programs.
[Pilih pada pilihan All Programs.]
 - c. Choose Borland C++ 5.02
[Pilih pada pilihan Borland C++ 5.02]
 - d. Click on the Borland C++ program.
[Klik pada aturcara Borland C++]
- iii. The Borland C++ software version 5.02 will be presented as below:
[Perisian Borland C++ versi 5.02 akan dipamerkan seperti berikut:]

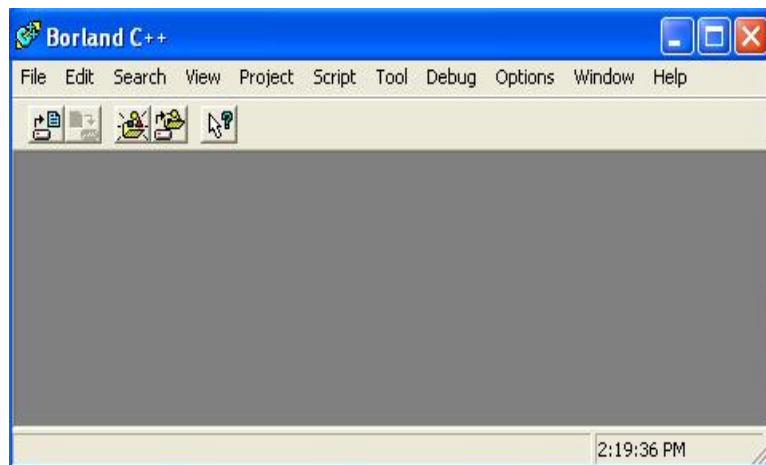


Figure 1.1 :

3. Writing the C source code.
[Menulis kod sumber C.]
 - i. The Borland C++ software consists of an editor that permits the programmer to type, edit, and save programs in C, as well as C++ language.
[Perisian Borland C++ mempunyai penyunting yang membenarkan pengaturcara menaip, menyunting/membaiki dan menyimpan aturcara dalam bahasa C dan juga C++]
 - ii. To write a new program, click on the **File** menu, choose **New**, and **Text Edit** option as follows:

[Untuk menulis aturcara yang baru, klik pada menu **File**, pilih **New** dan **Text Edit** seperti berikut:]

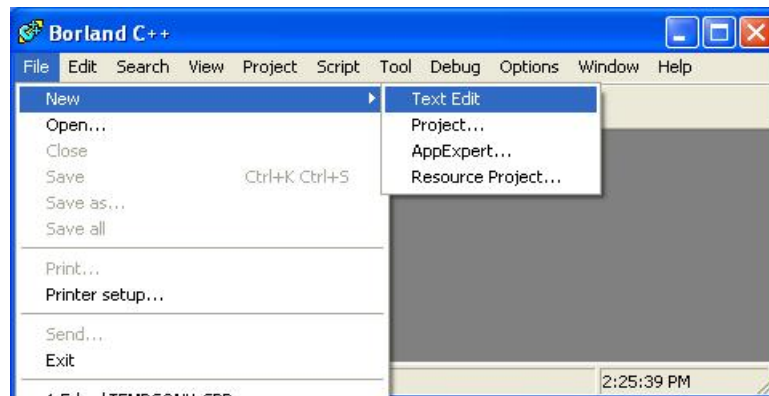


Figure 1.2 :

- iii. The following window will appear.
 [Tetingkap berikut akan dipamerkan.]

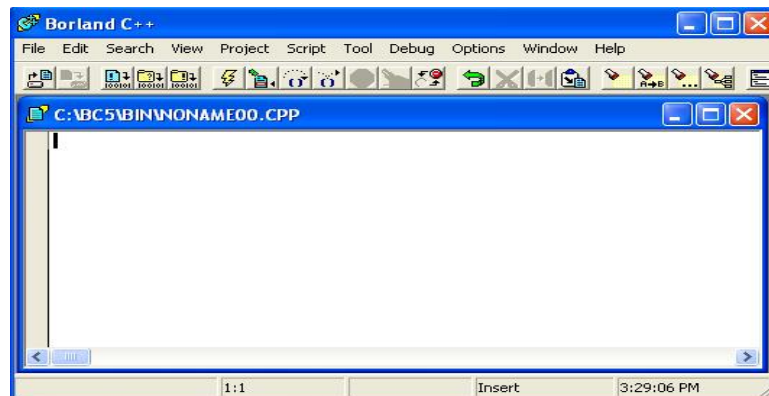


Figure 1.3 :

- iv. Type the following C statements. Do not type the line number.
 [Taipkan pernyataan C seperti berikut. Jangan taipkan nombor baris.]

```

1 //Program 1.1
2 //Nama: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?");
10    getch();
11    return 0;
    }
    
```

4. Saving the C source file.
[Menyimpan fail sumber C.]
- i. The program that you just wrote using the C editor is called the source program. This source program resides in the computer's main memory (RAM). You need to save the source program into the secondary storage (such as the hard disk, diskette, or pen drive) as a backup so that you can use the program file at other time.
[Aturcara yang telah ditulis menggunakan editor C dikenali sebagai aturcara sumber. Aturcara sumber ini berada di dalam ingatan utama computer (RAM). Anda perlu menyimpan aturcara sumber ini ke dalam storan sekunder (seperti cakera keras, disket, atau pemacu pen) sebagai sokongan supaya anda boleh menggunakan fail aturcara ini pada masa yang lain.]
 - ii. To save a source program into a new folder in the hard disk.
[Untuk menyimpan aturcara sumber ke dalam folder baru di cakera keras:]
 - a. Click at the **File** menu. Choose **Save As**.
[Klik pada menu **File**. Pilih pilihan **Save As**.]

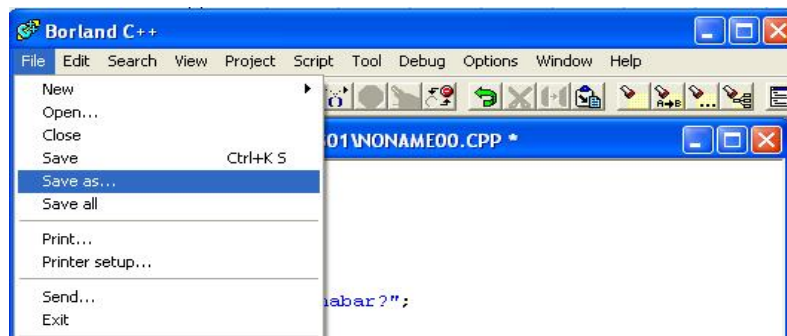


Figure 1.4 :

- b. To create a new folder so that it will be located under the Bin folder, click on the icon **Create New Folder** (as circled) in the **Save File As** dialog box.
[Untuk mencipta folder baru supaya berada di bawah folder Bin, klik pada ikon **Create New Folder** seperti yang dibulatkan pada bahagian kotak dialog **Save File As** berikut:]

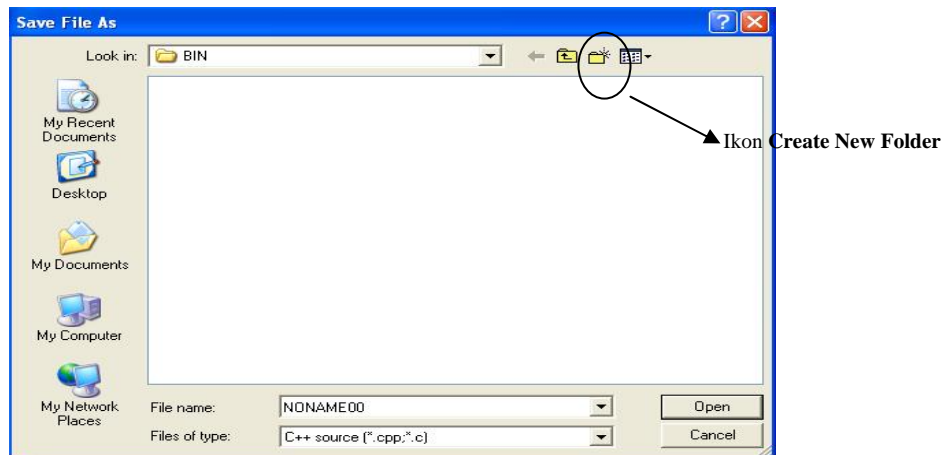


Figure 1.5 :

- c. In the **New Folder**, type the folder name keep all of your program files in that folder.
*[Di bagian **New Folder**, taipkan nama folder yang akan menyimpan fail-fail aturcara anda]*
- d. Click twice at your folder and type the name of your source program at the File name input box, as Hello1.c.
*[Klik dua kali pada folder anda itu dan kemudian taipkan nama fail aturcara sumber anda sebagai Hello1.c dan kotak input **File Name**]*

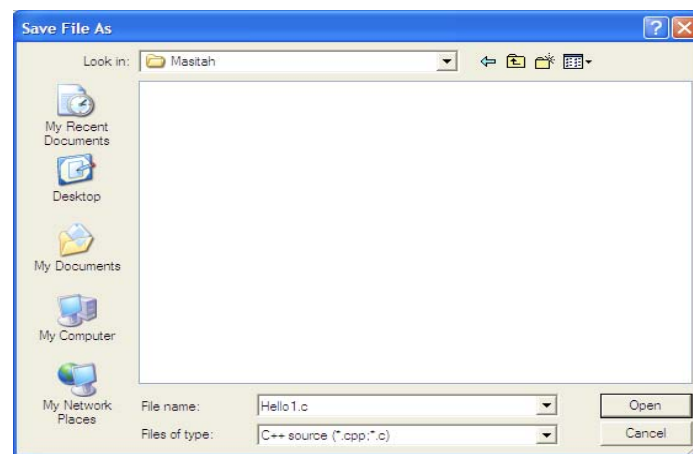


Figure 1.6 :

- e. Click on the button **Open** to save the file.
*[Klik pada bebutang **Open** untuk menyimpan fail.]*
5. Compiling the C program.
[Mengkompil aturcara C.]
- i. After you created a program, you can compile it so that the source code will be translated into object code. The C compiler will try to detect any syntax errors in your program.

[Setelah siap menaip aturcara, anda boleh mengkompil aturcara tersebut untuk menterjemahkan kod sumber kepada kod objek. Pengkompil bahasa C akan cuba mengesan sebarang ralat sintak yang terdapat di dalam aturcara.]

- ii. To compile a C program:
[Untuk mengkompil aturcara C]
 - a. Click on the **Project** menu and select the **Compile** option, or simultaneously press the **Alt** key and the **F9** function key. The following screen will appear:
[Klik pada menu **Project** dan pilih pilihan **Compile** atau tekan pada bebutang **Alt** dan kekunci fungsi secara serentak skrin berikut akan dipaparkan.]

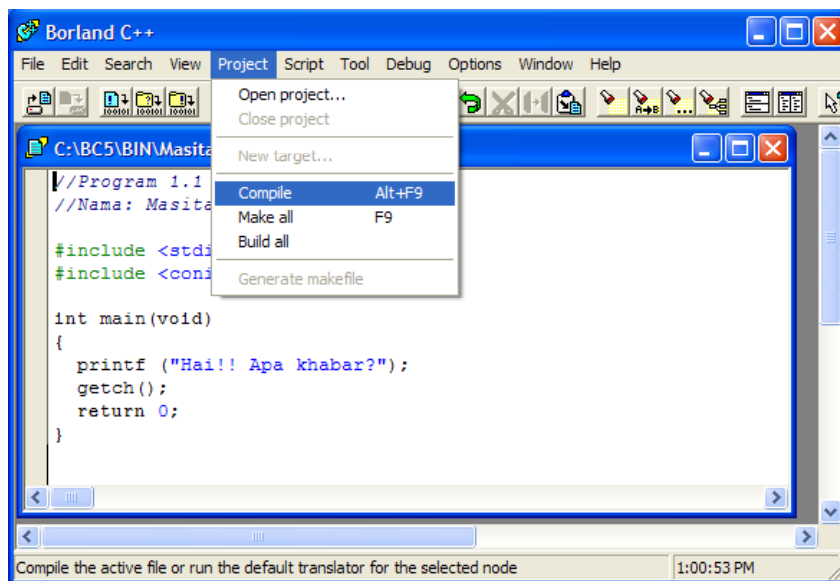


Figure 1.7 :

- b. If there is no syntax error, the compilation process is considered successful.
[Sekiranya tiada ralat sintaks ditemui, proses pengkompilan dianggap berjaya]

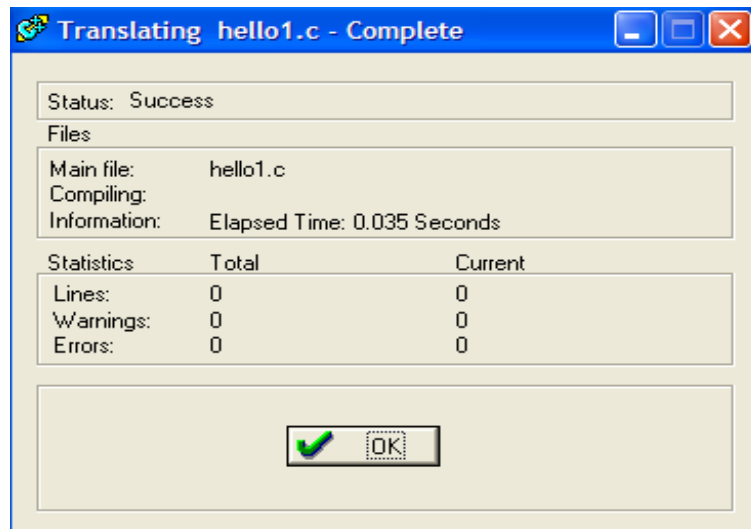


Figure 1.8 :

- c. If there are syntax errors, an error message will be displayed and the programmer will edit the code with errors and recompile the program until successful.
[Jika terdapat ralat sintaks, mesej ralat akan dipamerkan dan pengaturcara perlu memperbaiki arahan yang terdapat ralat dan mengkompil semula aturcara tersebut sehingga berjaya.]
- iii. Click on the **OK** button.
*[Klik pada bebutang **OK**]*
6. Running the C program.
[Melaksanakan aturcara C.]
- i. The program that has successfully been compiled, can be executed (run) so that the result will appear.
[Aturcara yang telah berjaya dikompil perlu dilaksanakan (run) supaya boleh menghasilkan keputusan.]
- ii. However, if error occurs, an error message will be displayed and the programmer need to edit and recompile, as well as rerun the program to produce the output.
[Bagaimanapun, jika terdapat ralat, mesej ralat akan dipamerkan dan pengaturcara perlu membaikinya dan mengkompil semula serta melaksana semula aturcara tersebut untuk menghasilkan output.]
- iii. To run the program:
[Untuk melaksanakan aturcara:]
- a. Click on the **Debug** menu and choose **Run**, or press the **Ctrl** and the **F9** function key simultaneously.
*[Klik pada menu **Debug** dan pilih **Run**, atau tekan pada bebutang **Ctrl** dan kekunci fungsi **F9** secara serentak]*

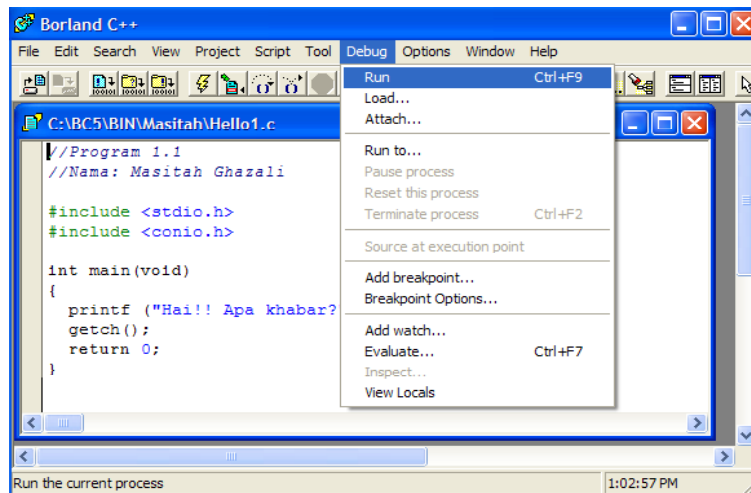


Figure 1.9 :

- b. The output is displayed in a DOS window, as shown below:
[Output akan dipaparkan pada tettingkap DOS seperti berikut:]

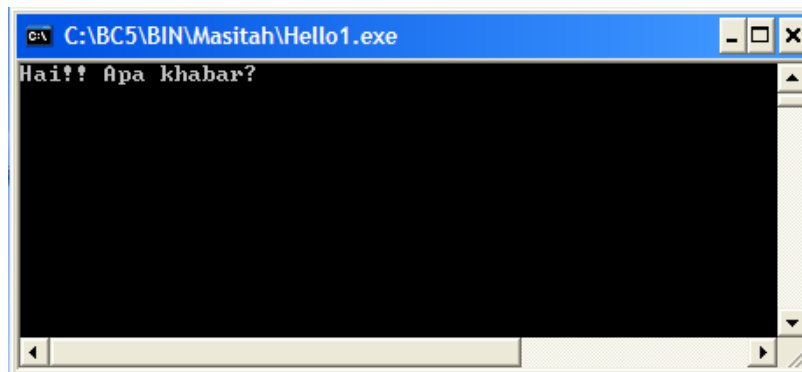


Figure 1.10 :

7. Close the C program.
[Tutup aturcara C.]
- i. The opened program source can be closed and reopened whenever required.
[Aturcara sumber yang sedang dibuka boleh ditutup dan boleh membukanya semula apabila diperlukan.]
 - ii. Step for closing the opened source program:
[Cara untuk menutup aturcara sumber yang sedang dibuka:]
 - a. Click the **File** menu and chose **Close**, or click the cross image at the top right hand corner the window.
*[Klik pada menu **File** dan pilih **Close**, ataupun klik pada tanda pangkah di bahagian atas sebelah kanan tettingkap aturcara sumber tersebut.]*

EXERCISE 2:**[LATIHAN 2]**

Often you need to load existing C source file, edit the content and compile it.

[*Seringkali anda perlu memuatkan fail sumber C sedia ada, mengedit kandungannya serta menkompilnya.*]

1. Load the Borland C++ software as in Exercise 1 Part 2 (see page 3).
[*Muatkan perisian Borland C++ seperti Latihan 1 Bahagian 2. (Lihat halaman 3)*]
2. Open the existing C source file as follows:
[*Buka sumber fail C yang wujud seperti berikut:*]
 - i. Click on the **File** menu and choose **Open** or click on the **Open a file** icon as shown in figure 1.11 below.
[*Buka fail sumber C sedia ada seperti berikut pada menu File dan memilih Open atau dengan mengklik pada ikon Open a file seperti yang ditunjukkan oleh rajah 1.11 berikut.*]

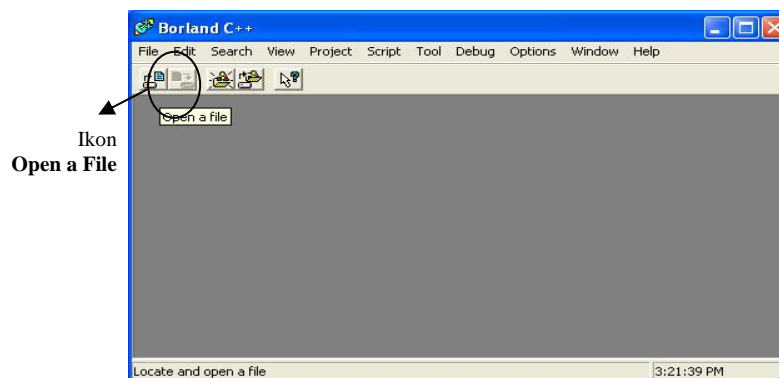


Figure 1.11 :

- ii. Once the dialog box appears, type the appropriate file name as shown in figure 1.12 below.
[*Apabila kekotak dialog dipamerkan, taipkan nama fail seperti tertera dalam tajah 1.12 berikut.*]

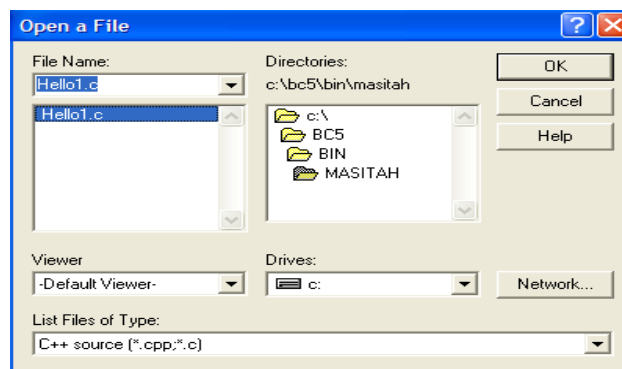


Figure 1.12 :

- iii. Add a new line of code at line 9:
[*Tambah satu barisan kod baru pada baris 9:*]

```
1 //Program 1.2
2 //Nama pengaturcara: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?\n");
10    printf ("Ucapan dari Masitah Ghazali\n");
11    getch();
12    return 0;
}
```

3. Compile the C program as in Exercise 1 Part 5. (See page 7)
[*Mengkompil aturcara C seperti Latihan 1 Bahagian 5. (Lihat halaman 7)*]
4. Run the C program as in Exercise 1 Part 6. (See page 9)
[*Melaksana aturcara C seperti Latihan 1 Bahagian 6. (Lihat halaman 9)*]
5. Show the output of program 1.2 above.
[*Tunjukkan output bagi aturcara 1.2 di atas.*]
6. Close the C program as in Exercise 1 Part 7. (See page 10)
[*Tutup aturcara C seperti Latihan 1 Bahagian 7. (Lihat halaman 10)*]
7. Shut down the computer system.
[*Tutup sistem komputer.*]
 - i. Close all application programs before shutting down the computer.
[*Pastikan semua perisian telah ditutup sebelum menutup komputer anda.*]
 - ii. Click on the **Start** menu and choose **Turn Off Computer**.
[*Klik pada menu **Start** dan pilih **Turn Off Computer**.*]
 - iii. Click pada ikon **Turn Off** as in figure 1.13.
[*Klik pada ikon **Turn Off** seperti rajah 1.13.*]



Figure 1.13 :

EXERCISE 3:**[LATIHAN 3]**

1. The annotation of program 1.1 is as in page 5 will be as follows:
[Penerangan bagi aturcara 1.1 seperti di halaman 5 adalah seperti berikut:]

```

1 //Program 1.1
2 //Nama: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio.h>
6
7 int main(void)
8 {
9     printf ("Hai!! Apa khabar?\n");
10    getch();
11    return 0;
12 }
```

- i. Line 1 and 2 are **comments**. A **comment** is a piece of descriptive text (letters, number or special symbols) which explains some aspect of a program. Program comments are considered as **non-executable** statements that are totally ignored by the compiler and are only intended for human readers. In program 1.1 above, the symbol // are used to begin the comment in a line.
[Baris 1 dan 2 merupakan suatu **komen**. **Komen** teks (huruf, nombor atau simbol-simbol) yang menerangkan mengenai aspek-aspek aturcara. Komen dalam aturcara merupakan penyata **bukan pelaksana** (non-executable statement) yang akan diabaikan oleh pengkompil C dan ditunjukkan hanya untuk pembaca aturcara sahaja. Pada aturcara 1.1 di atas, penyata komen dimulai dengan simbol //]
- ii. Lines 4 and 5 are the **preprocessor directive** #include that include the contents of the library file, namedly `stdio.h` and `conio`, in the program. `stdio.h` is a standard C header file and contains definitions for input and output, such as `printf` and `scanf`. While

the `conio` is another header file that contains functions, such as `getch()`, that accepts an input data from the keyboard.

[Baris 4 dan 5 ialah **arahan prapemproses** yang akan menyertakan kandungan bagi fail perpustakaan bernama `stdio.h` dan `conio`, ke dalam aturcara. `stdio.h` ialah fail kepala C yang piawai dan mengandungi penakrifan input dan output seperti `printf` dan `scanf`. Manakala `conio` pula ialah satu lagi fail kepala yang mengandungi fungsi-fungsi, seperti `getch()`, yang menerima data masukan daripada papan kekunci.]

- iii. Line 7 defines a function called **main** that begins the program execution. The brackets without any word indicate the main function has no parameters. The return type for main is `int` (i.e., an integer number).

[Baris 7 menakrifkan satu fungsi bernama **main** yang memulakan pelaksanaan aturcara. Tanda dua kurungan tanpa sebarang perkataan di antaranya menunjukkan fungsi main tidak mempunyai sebarang parameter. Jenis data yang dikembalikan oleh fungsi main ialah `int` (iaitu, suatu nombor integer)]

- iv. Line 8 is an open brace that marks the beginning of the body of main function.

[Baris 8 merupakan pembuka kurungan berkeluk yang menandakan permulaan badan fungsi main.]

- v. Line 9 is a **statement**. The end of a statement is always marked with a semicolon (;). This statement causes the **string** "Hai!! Apa khabar?" to be sent to the `printf` **output stream**. A string is any sequence of characters enclosed in double-quotes. The last character in this string (`\n`) is a newline character which returns the cursor to the new line. A stream is an object which performs input or output. `printf` is the standard output stream in C (standard output usually means your computer monitor screen).

[Baris 9 ialah suatu pernyataan. Pernyataan lazimnya diakhiri dengan satu tanda semikolon (;). Pernyataan ini menyebabkan rentetan "Hai!! Apa khabar?" dihantar kepada aliran (stream) output `cout`. Suatu rentetan ialah sebarang turutan aksara yang dikurungkan dengan tanda petik berkembar "quote". Aksara terakhir pada rentetan ini (`\n`) merupakan satu aksara baris baru yang akan membawa kursor kembali ke baris baru. Aliran (stream) ialah suatu objek yang akan melaksanakan input atau output. `printf` ialah aliran output piawai bagi C. Dalam kes ini, kesan operasi ini ialah rentetan "Hai!! Apa khabar?\n" dihantar kepada `printf`, menyebabkannya dipaparkan ke skrin computer].

- vi. Line 10 is another statement that is a function `getch()` from the library `conio`. `getch()` will prompt and wait for the user enters any key at the keyboard.

[Baris 10 adalah satu lagi pernyataan iaitu fungsi bernama `getch()` daripada perpustakaan `conio`. `getch()` akan menggesa dan menunggu sehingga pengguna menekan sebarang kekunci pada papan kekunci]

- vii. Line 11 is the **return** statement that the control of the program to the operating system. It also determines the value of the return type that match the main function.

[Baris 11 ini ialah pernyataan **return** yang akan mengalihkan kawalan daripada aturcara kepada sistem pengoperasian. Ia juga akan menentukan nilai bagi fungsi yang akan dipulangkan dan perlu berpadanan dengan jenis data pulangan bagi fungsi main.]

- viii. Line 12 is a close brace that marks the end of the body of main.

[Baris 12 merupakan penutup kurungan berkeluk yang menandakan akhir bahagian badan fungsi main.]

2. Given program 1.3 below:

[Diberi aturcara 1.3 berikut:]

```
1  /* Program 1.3
2  Nama pengaturcara: Masitah Ghazali
3  Tujuan aturcara : Masukkan nama dan tahun lahir
4  serta kira usia pengguna pada tahun ini */
5
6  #include <stdio.h>      //Pre-processor directive
7  #include <conio.h>     //Pre-processor directive
8
9  int main()
10 {
11     const int TahunIni=2010;
12     char nama[20];
13     int tahunlahir, umur;
14
15     printf ("Masukkan nama anda: ");
16     scanf ("%s", &nama);
17
18     printf ("Masukkan tahun anda dilahirkan: ");
19     scanf ("%d", &tahunlahir);
20
21     umur = TahunIni - tahunlahir;
22
23     printf ("Nama anda: %s", nama);
24     printf ("\nUsia anda: %d", umur, "tahun.");
25     getch();
26     return 0;
27 }
```

- i. Type, save, compile, and run program 1.3 above. What will be the output of the program for the given input?

[*Taip, simpan, kompil, dan laksanakan aturcara tersebut. Apakah output aturcara di atas bagi input yang dimasukkan?*]

- ii. Briefly annotate what each line of code means.
[*Terangkan secara ringkas maksud setiap baris kod.*]

3. Given program 1.4 below:
[*Diberi aturcara 1.4 berikut:*]

```
1 //Program 1.4
2 //Nama pengaturcara: Masitah Ghazali
3
4 #include <stdio.h>
5 #include <conio>
6
7 int main (void)
8 {
9     int workDays;
10    float workHours, payRate, weeklyPay;
11
12    workDays = 5;
13    workHours = 6.5;
14    payRate = 20.50;
15    weeklyPay = workDays * workHours * payRate;
16    printf ("Weekly Pay = %f", weeklyPay);
17    printf ("\n");
18    getch();
19    return 0;
20 }
```

- i. Type, save, compile, and run the program. What will be the output of the program?
[*Taip, simpan, kompil, dan laksanakan aturcara tersebut. Apakah output bagi aturcara di atas?*]
- ii. Briefly annotate what each line of code means.
[*Terangkan secara ringkas maksud setiap baris kodnya.*]

LAB 2: DESIGNING PSEUDOCODE AND FLOWCHART USING MICROSOFT WORD

OBJECTIVES FOR STUDENTS

1. Develop pseudocode using Microsoft Word software.
[*Membina kod sudo menggunakan perisian Microsoft Word.*]
2. Develop flowchart using Microsoft Word software.
[*Membina carta alir menggunakan perisian Microsoft Word.*]
3. Trace pseudocode and flowchart in order to determine the contents of the variables and the output.
[*Menjejak kod sudo dan carta alir untuk menentukan kandungan pemboleh ubah dan output algoritma.*]
4. Design a pseudocode or a flowchart with given problem.
[*Mereka bentuk kod sudo dan carta alir bagi masalah yang diberikan.*]

ASSUMPTIONS



5. Students have the basic experience in starting (booting) the computer.
[*Pelajar mempunyai pengalaman asas memulakan (but) komputer.*]
6. Students have the basic experience in loading and opening particular software in Microsoft Windows.
[*Pelajar mempunyai pengalaman asas memuat dan membuka perisian tertentu dalam Microsoft Windows.*]

LAB EXERCISES

EXERCISE 1: DEVELOP THE PSEUDOCODE USING MICROSOFT WORD [*LATIHAN 1: MEMBINA KOD SUDO MENGGUNAKAN MICROSOFT WORD*]

7. Loading the Microsoft Word software.
[*Memuatkan perisian Microsoft Word.*]
 - i. Load the Microsoft Word software into the main memory of the computer (i.e. RAM).
[*Muatkan perisian Microsoft Word ke dalam ingatan utama komputer, iaitu RAM.*]
 - ii. To load the Microsoft Word software:

[Untuk memuatkan perisian Microsoft Word:]

- a. Click on the **Start** menu at the bottom-left of the screen.
[Klik pada menu **Start** yang berada di bahagian bawah sebelah kiri skrin.]
- b. Choose **Microsoft Word** represented by the  icon
[Pilih **Microsoft Word** yang diwakili oleh ikon ]

The Microsoft Word software is displayed as in Figure 2.1 below:
[Perisian Microsoft Word dipamerkan seperti dalam Rajah 2.1 berikut:]

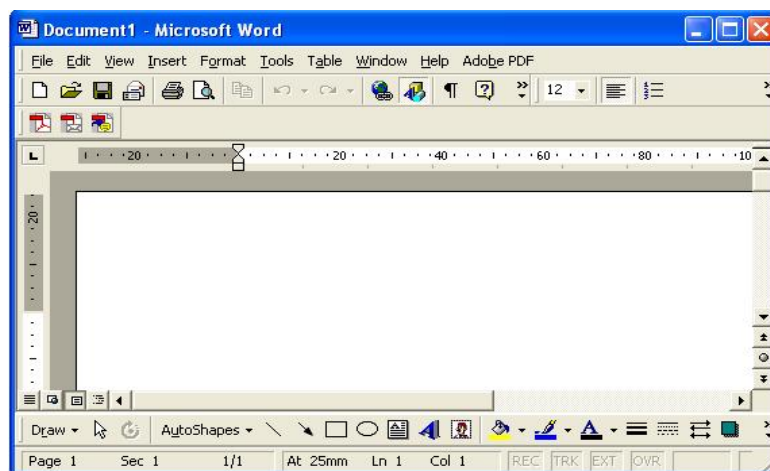


Figure 2.1

8. Type the following pseudocode.
[Taipkan kod sudo berikut.]

```
Algorithm 2.1 : To calculate the total of three numbers
1. Start
2. Set TOTAL to 0
3. Read NUMBER1
4. Read NUMBER2
5. Read NUMBER3
6. Add NUMBER1, NUMBER2 dan NUMBER3 and store in TOTAL
7. Display TOTAL
8. End
```

OR
[Atau]

Algoritma 2.2 : Mengira hasil tambah bagi tiga nombor

1. Mula
2. Setkan HASILTAMBAH bersamaan 0
3. Baca NOMBOR1
4. Baca NOMBOR2
5. Baca NOMBOR3
6. Tambah NOMBOR1, NOMBOR2 dan NOMBOR3 dan simpan dlm HASILTAMBAH
7. Paparkan HASILTAMBAH
8. Tamat

9. Save the pseudocode file.

[Simpan fail kod sudo.]

iii. To save a word document:

[Untuk menyimpan dokumen word:]

a. Click on the **File** menu. Select **Save As** as shown in Figure 2.2.

[Klik pada menu **File**. Pilih pilihan **Save As** seperti Rajah 2.2.]

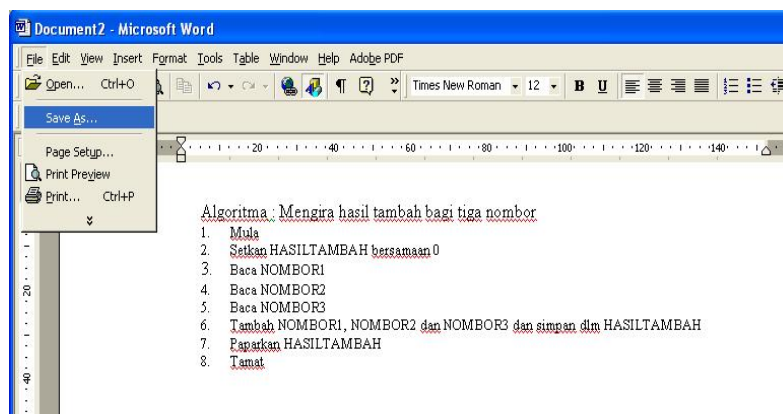


Figure 2.2

b. Click twice on the folder that contains your files. Then type your word document filename. For example, **KodSudoKira3Nom** as in Figure 2.3 below:

[Klik dua kali pada folder yang mengandungi fail-fail anda. Kemudian taipkan nama fail dokumen word. Sebagai contoh, **KodSudoKira3Nom** seperti Rajah 2.3 berikut.]

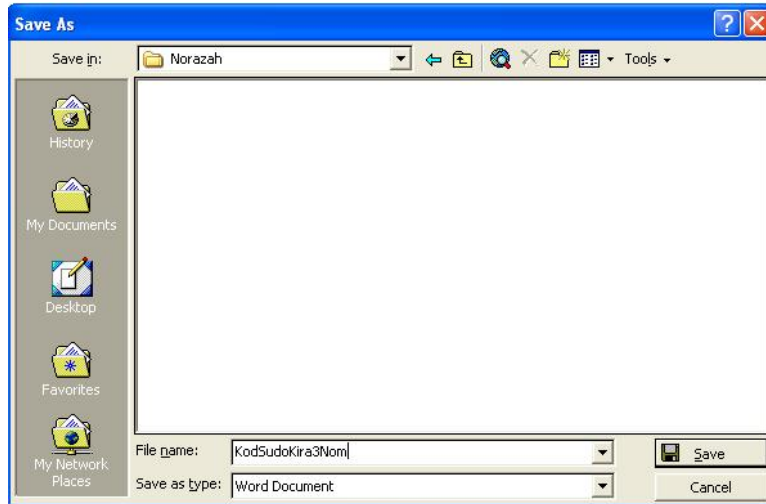


Figure 2.3

- c. Click on the **Save** button
[Klik bebutang *Save*]

EXERCISE 2: DEVELOP THE FLOWCHART USING MICROSOFT WINDOWS

[LATIHAN 2: MEMBINA CARTA ALIR MENGGUNAKAN MICROSOFT WINDOWS]

1. Move cursor to new page.
[Gerak kursor ke halaman baru.]
 - i. Apply the **Page Break** function:
[Pilih fungsi **Page Break**.]
 - a. On the **Insert** menu, select **Break** as shown in Figure 2.4.
[Pada menu **Insert**, pilih pilihan **Break** seperti yang ditunjukkan dalam Rajah 2.4.]

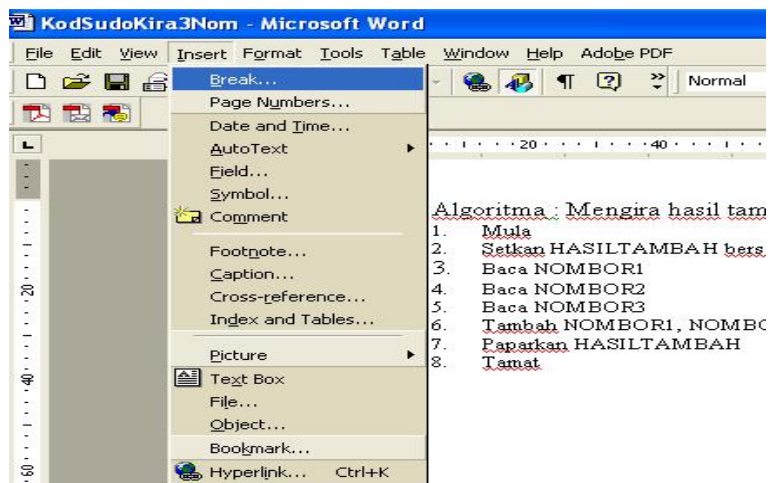


Figure 2.4

- b. On the following dialogue box, select **Page Break** and click on the **OK** button. See Figure 2.5.

[Pada kekotak dialog berikut, pilih pada **Page Break** dan klik pada bebutang **OK**. Lihat Rajah 2.5.]

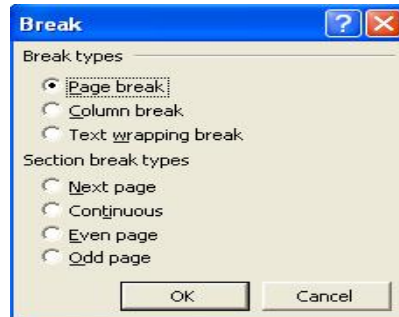


Figure 2.5

2. Activate the **Drawing** submenu.
[Aktifkan submenu **Drawing**.]
3. Make sure that the **Drawing** submenu is activated. If it is not active, on the **View** menu, select **Toolbars** and select **Drawing** as Figure 2.6 below:
[Pastikan bahagian **Drawing** diaktifkan. Jika tidak aktif, pada menu **View**, pilih **Toolbars** dan pilih **Drawing** seperti Rajah 2.6 berikut:]

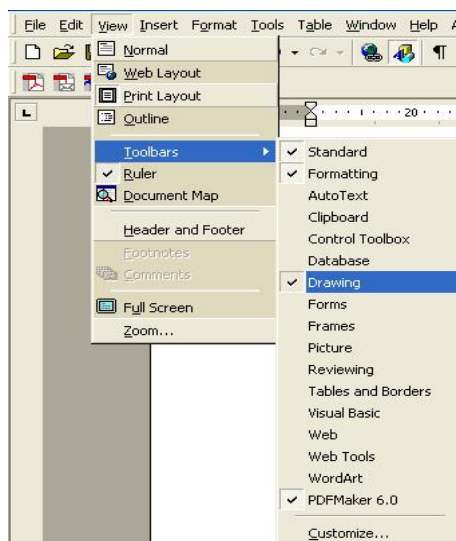


Figure 2.6

If active, the **Drawing** submenu is normally found as a panel located at the bottom of the screen.

[Sekiranya aktif, submenu **Drawing** lazimnya terdapat sebagai panel di bahagian bawah skrin.]

4. On the **Drawing** panel, click on the text box icon. See Figure 2.7.
[Pada panel **Drawing**, klik pada ikon kekotak teks. Lihat Rajah 2.7.]



Figure 2.7

5. Draw a rectangular shape and label it with **Start** as below:
 [Lukis satu kawasan segi empat dan labelkannya sebagai **Start** seperti contoh berikut:]



6. Change its shape to a terminal symbol by clicking on the **Draw** menu, select **Change AutoShape**, select **Flowchart**, select the **terminator** symbol. See Figure 2.8.
 [Ubah rupa supaya berbentuk simbol terminal dengan cara, klik pada menu **Draw**, pilih **Change AutoShape**, pilih **Flowchart**, pilih simbol **terminator**. Lihat Rajah 2.8.]

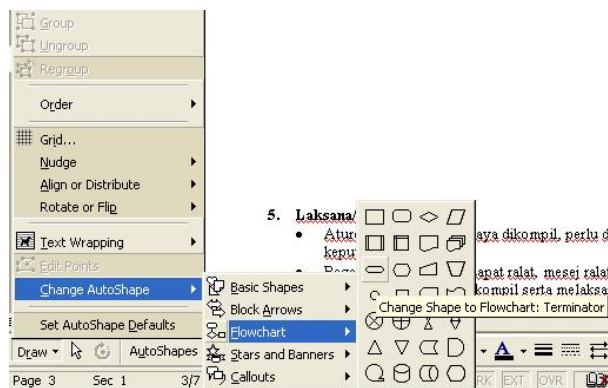


Figure 2.8

7. Draw the flowchart as in Figure 2.9:
 [Bina carta alir seperti Rajah 2.9:]

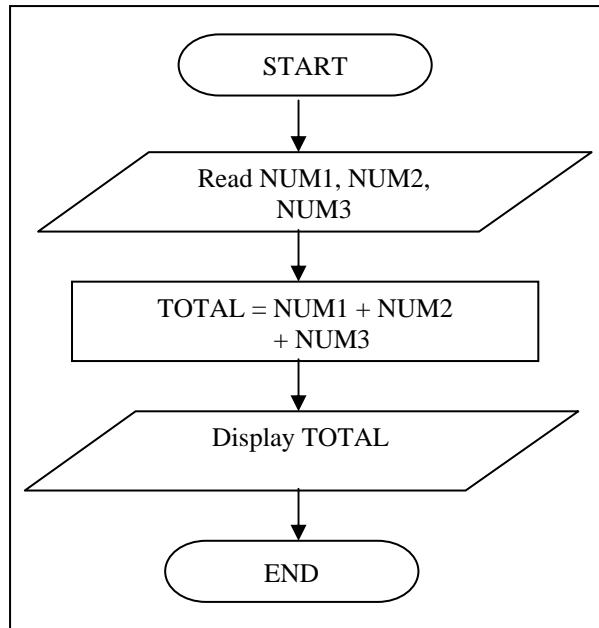


Figure 2.9

This flowchart is a conversion from the pseudocode done in Exercise 1.
 [Carta alir ini ditukar daripada kod sudo yang telah di bina dalam Latihan 1.]

8. Save the file as **CartaAlirKira3Nom**.
 [Simpan fail sebagai **CartaAlirKira3Nom**.]

EXERCISE 3:
 [LATIHAN 3]

1. Using the Microsoft Word software, develop the pseudocode as in Algorithm 2.3 that converts the distance in mile to kilometer. Then trace the content of the variables and determine the output of Algorithm 2.3.
 [Dengan menggunakan perisian Microsoft Word, bina kod sudo seperti Algoritma 2.3 yang menukarkan jarak dalam unit batu kepada kilometer. Kemudian jejak kandungan pembolehubah dan tentukan output Algoritma 2.3 tersebut.]

```

Algorithm 2.3: Convert the distance in mile to kilometer
1. Start
2. Set KM_PER_MILE to 1.609
3. Set Distance_Mile to 100
4. Convert the distance in mile to kilometers using the formula:
   Distance_Kilo = KM_PER_MILE x Distance_Mile
5. Display the distance in kilometer
6. End
  
```

2. Using the Microsoft Word software, develop a pseudocode that assigns values to variables as in Algorithm 2.4. Then trace the content of the variables and determine the output of the Algorithm 2.4.

[Dengan menggunakan perisian Microsoft Word, bina kod sudo yang mengumpukkan nilai kepada pembolehubah seperti Algoritma 2.4. Kemudian jejak kandungan pembolehubah dan tentukan output bagi Algoritma 2.4 tersebut.]

Algorithm 2.4: Determine the content of variables

1. Start
2. Set Data1 = 5
3. Set Data2 = 2.5
4. Data3 = Data1 + Data2
5. Data2 = Data3 x Data1
6. Display Data1, Data2, Data3
7. End

3. Using the Microsoft Word software, develop a pseudocode that reads the radius of a circle, and computes the area of the circle as in Algorithm 2.5.

[Dengan menggunakan perisian Microsoft Word, bina kod sudo seperti Algoritma 2.5 yang membaca jejari bagi suatu bulatan dan mengira luas kawasan bulatan tersebut.]

Algorithm 2.5: Compute the area of a circle

1. Start
2. Set PI = 3.14159
3. Read the Radius
4. Calculate the area of a circle using the formula:
Area = Radius x Radius x PI
5. Display Area
6. End

- i. Trace the content of the variables and determine the output of the Algorithm 2.5 if the input for Radius is:

[Jejak kandungan pembolehubah serta tentukan output Algoritma 2.5 sekiranya input jejari (Radius) ialah:]

- a. 3
- b. 10
- c. 150

4. Using the Microsoft Word software, develop a pseudocode that reads the money collection of a user. The algorithm computes the number of Ringgit and cents (50, 20, 10 5, and 1 cent) that can be extracted from the collection. Tips: You can use the **Modulus** operator that obtains the remainder from a division.

[Dengan menggunakan perisian Microsoft Word, bina kod sudo yang membaca koleksi duit pengguna. Algoritma tersebut mengira bilangan Ringgit dan syiling (50, 20, 10, 5, dan 1 sen) yang boleh diasing-asingkan daripada koleksi tersebut. Panduan: Anda boleh menggunakan operator **Modulus** untuk mendapatkan baki daripada hasil bahagi.]

- i. Trace the content of the variables and determine the output of Algorithm 2.6 if the input for the money collection is as follows:
[Jejak kandungan pembolehubah serta tentukan output bagi Algoritma 2.6 sekiranya input koleksi adalah seperti berikut:]
 - a. 927
 - b. 1274
 - c. 6785
 - d. 35565

Algorithm 2.6: Compute Money Change from Collection to Ringgit and cents

1. Start
2. Read the Collection
3. Determine the total for ringgit using the formula:
 $\text{Ringgit} = \text{Collection} / 100$
4. Determine the remaining amount using the formula:
 $\text{Remain_Amt} = \text{Collection} \text{ Modulus } 100$
5. Determine the total 50 cents in the remaining amount using the formula:
 $\text{Fifty_Cent} = \text{Remain_Amt} / 50$
6. Determine the remaining amount using the formula:
 $\text{Remain_Amt} = \text{Remain_Amt} \text{ Modulus } 50$
7. Determine the total 20 cents in the remaining amount using the formula:
 $\text{Twenty_Cent} = \text{Remain_Amt} / 20$
8. Determine the remaining amount using the formula:
 $\text{Remain_Amt} = \text{Remain_Amt} \text{ Modulus } 20$
9. Determine the total 10 cents in the remaining amount using the formula:
 $\text{Ten_Cent} = \text{Remain_Amt} / 10$
10. Determine the remaining amount using the formula:
 $\text{Remain_Amt} = \text{Remain_Amt} \text{ Modulus } 10$
11. Determine the total 5 cents in the remaining amount using the formula:
 $\text{Five_Cent} = \text{Remain_Amt} / 5$
12. Determine the remaining amount using the formula:
 $\text{Remain_Amt} = \text{Remain_Amt} \text{ Modulus } 5$
13. Determine the total 1 cents in the remaining amount using the formula:
 $\text{One_Cent} = \text{Remain_Amt}$
14. Display Collection, Ringgit, Fifty_Cent, Twenty_Cent, Ten_Cent, Five_Cent, One_Cent
15. End

EXERCISE 4:**[LATIHAN 4]**

1. Using the Microsoft Word software, draw a flowchart as in Figure 2.10 that calculates the payment for normal and overtime working hours of an employee. Then trace the content of the variables and determine the output displayed from the algorithm.

[Dengan menggunakan perisian Microsoft Word, lukiskan carta alir seperti Rajah 2.10 yang mengira bayaran untuk jam kerja normal dan lebih masa seseorang pekerja. Kemudian jejak kandungan pembolehubah dan tentukan output algoritma tersebut.]

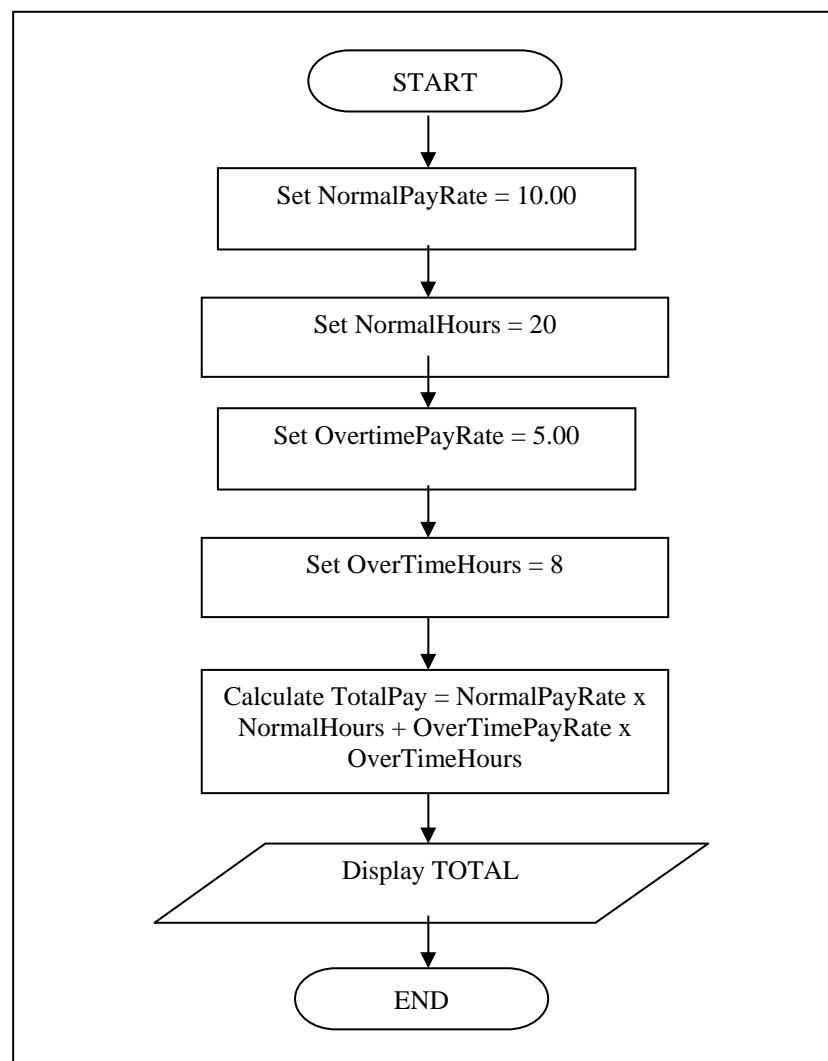


Figure 2.10

2. Using the Microsoft Word software, draw a flowchart as in Figure 2.11 that calculates the total cost of item purchased.
[Dengan menggunakan perisian Microsoft Word, lukiskan carta alir seperti Rajah 2.11 yang mengira jumlah harga sesuatu pembelian.]

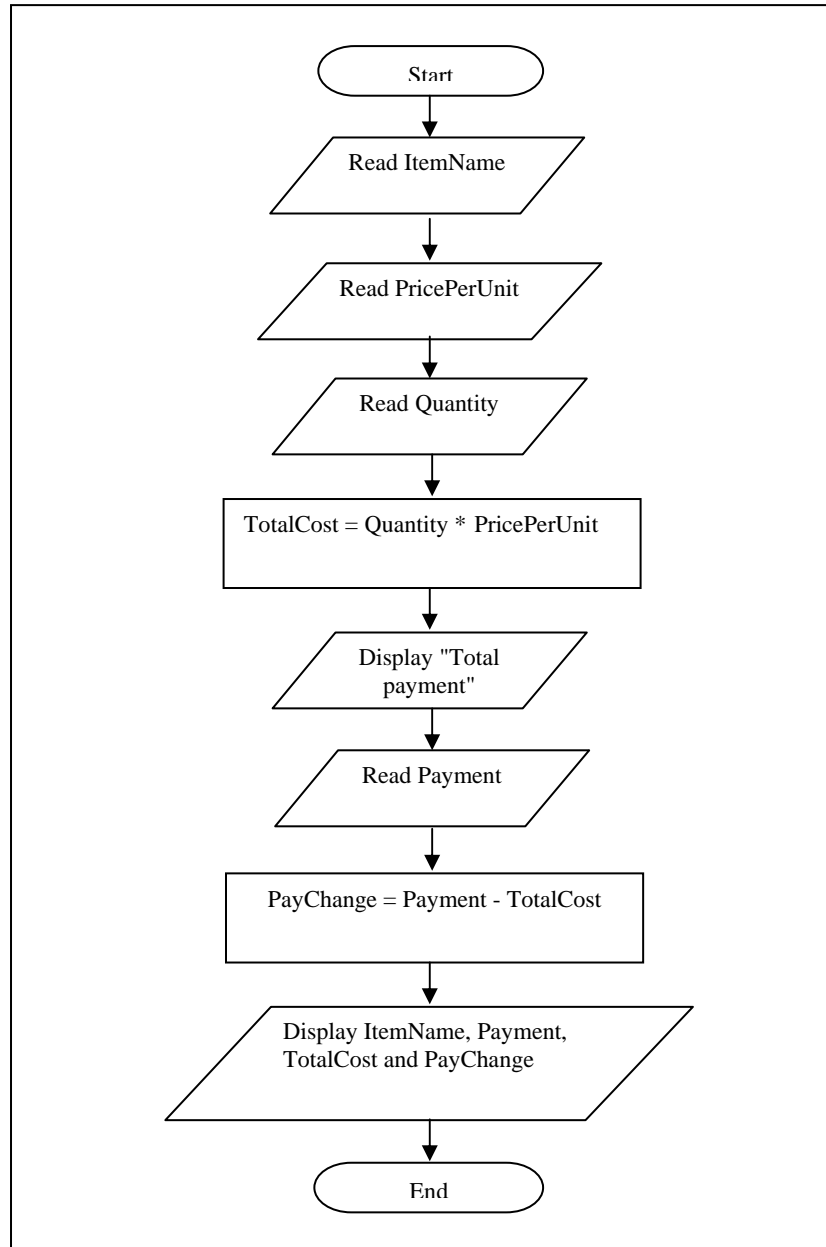


Figure 2.11

- i. Trace the content of the variables and determine the output of the algorithm if the input values are as follows:
[*Jejak kandungan pembolehubah dan tentukan output algoritma tersebut sekiranya input adalah seperti berikut:*]
 - a. ItemName: **Pencil**
PricePerUnit: **0.50**
Quantity: **15**
Payment: **10.00**
 - b. ItemName: **Note book**
PricePerUnit: **2.00**
Quantity: **5**
Payment: **50.00**
 - c. ItemName: **Pencil**
PricePerUnit: **3**
Quantity: **1**
Payment: **5.00**

EXERCISE 5:

[*LATIHAN 5*]

Given the following set of problems. Analyze each problem and identify the input, formula and the output. Write the pseudocode and draw the flowchart for the problem. [*Diberi masalah-masalah seperti berikut. Analisis setiap masalah dan kenalpasti input, formula serta output. Tuliskan kod sudo dan lakarkan carta alir bagi masalah berkenaan.*]

1. Read three numbers. Calculate the sum of those three numbers and find the average. Display all the three numbers, as well as the total of those numbers and the in average.
[*Baca tiga nombor. Kira dan jumlahkan ketiga-tiga nombor dan cari purata. Paparkan ketiga-tiga nombor, dan juga jumlah ketiga-tiga nombor serta puratanya.*]

Output:

Input:

Formula:

Pseudocode design of the problem:

Flowchart design of the problem:

2. Calculate the average weight of four students. Attempt to analyze this problem by stating its input, output and the formula required to determine the average weight. Display each student's weight and the average weight calculated.
[Cari purata berat bagi 4 orang pelajar. Paparkan berat setiap pelajar dan purata yang dikira. Cuba analisa permasalahan ini dengan menyatakan input, output serta formula yang diperlukan untuk mengira purata.]

Output:

Input:

Formula:

Pseudocode design of the problem:

Flowchart design of the problem:

3. Convert the time entered by the user in the form of day, hours and minute to minutes. The conversion unit is as follows:

1 day is equivalent to 24 hours

1 hour is equivalent to 60 minutes

[Tukarkan masa yang dimasukkan oleh pengguna dalam bentuk hari, jam dan minit ke minit. Pertukaran unit adalah seperti berikut:

1 hari bersamaan dengan 24 jam

1 jam bersamaan dengan 60 minit]

Output:

Input:

Formula:

Pseudocode design of the problem:

Flowchart design of the problem:

4. Convert the temperature entered by the user in the unit of Fahrenheit to the unit of Celsius. The conversion formula is as follows:

$$\text{Fahrenheit} = \left[\text{Celsius} \times \frac{9}{5} \right] + 32$$

[Tukarkan suhu yang dimasukkan oleh pengguna ke dalam unit Fahrenheit kepada unit Celsius. Formula penukaran unit adalah seperti berikut:]

Output:

Input:

Formula:

Pseudocode design of the problem:

Flowchart design of the problem:

EXERCISE 6:

[LATIHAN 6]

Given the following problem. Analyze the problem and design its solution using structured chart. Then write the appropriate pseudocode or flowchart.

[Diberi masalah berikut. Analisa masalah tersebut dan reka bentuk penyelesaiannya menggunakan carta berstruktur. Kemudian, tuliskan kod sudo atau carta alir yang bersesuaian.]

1. A student has designed an object as shown in Figure 2.12. You need to calculate both the perimeter and the area of the object, based on the length and the width of the rectangle entered by the user, as well as the height of the triangle.

[Seorang pelajar telah mereka bentuk suatu objek seperti Rajah 2.12. Anda perlu mengira perimeter dan luas objek tersebut berdasarkan kepada nilai panjang dan lebar segiempat yang dimasukkan oleh pengguna, serta nilai tinggi segitiga objek tersebut.]

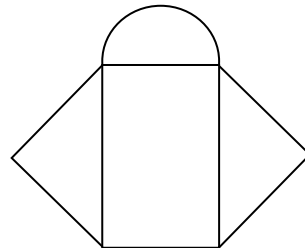
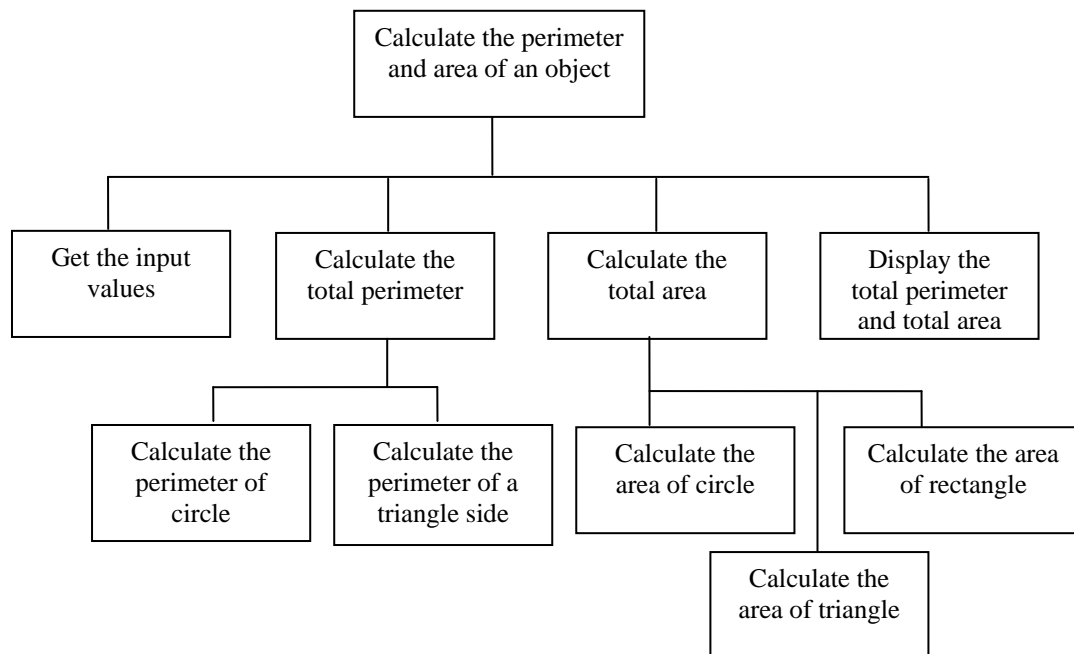


Figure 2.12



LAB 3: DESIGN PSEUDOCODE/FLOWCHART TO SOLVE SELECTION AND LOOP PROBLEM

OBJECTIVES FOR STUDENTS

1. Develop pseudocode to solve selection and loop problem.
[*Membina kod sudo bagi menyelesaikan masalah pilihan dan gelung.*]
2. Develop flowchart to solve selection and loop problem.
[*Membina carta alir bagi menyelesaikan masalah pilihan dan gelung.*]
3. Trace pseudocode and flowchart in order to determine the contents of the variables and the output.
[*Menjejak kod sudo dan carta alir untuk menentukan kandungan pemboleh ubah dan output algoritma.*]

ASSUMPTIONS

1. Students have knowledge on designing simple pseudocode and flowchart.
[*Pelajar mempunyai pengetahuan di dalam merekabentuk kod sudo dan carta alir.*]

LAB EXERCISES

EXERCISE 1:

[LATIHAN 1]

1. Based on the following information regarding input, output and process, write the pseudocode and flowchart to solve the problem.
[*Berdasarkan maklumat input, output dan proses di bawah, tuliskan kod sudo dan carta alir untuk menyelesaikan masalah tersebut.*]

Problem : Determine whether profit, return capital or loss.

[*Masalah : Tentukan sama ada untung, balik modal atau rugi.*]

Input :

- i. Purchase price (in ringgit).
[*Harga beli (dalam ringgit).*]
- ii. Selling price (in ringgit).
[*Harga jualan (dalam ringgit).*]

Output : Message "Profit" or "Return Capital" or "Loss"
 [Mesej "Profit" atau "Return Capital" atau "Loss"]

Process :

Difference = Purchase price - Selling price
 If Difference > 0, Output = "Profit"
 If Difference = 0, Output = "Return Capital"
 If Difference < 0, Output = "Loss"

2. Draw suitable flowchart symbol and write the correct C expression for the following conditions.

[Lukis simbol carta alir yang sesuai dan tuliskan ungkapan C yang betul bagi setiap syarat di bawah.]

i. Number greater than 1 and smaller or equal than 10.
 [Nombor lebih besar daripada 1 dan lebih kecil atau sama dengan 10.]

ii. Number between 0 to 9 but not 6.
 [Nombor adalah di antara 0 hingga 9 tetapi bukan 6.]

iii. Variable ans not character 'y' or 'Y'
 [Pembolehkan ans bukan huruf 'y' atau 'Y'.]

iv. Even number equal to 0.
 [Nombor genap bersamaan 0.]

3. Execute the flowchart in Figure 3.1 using the input values supplied in the following Table 3.1. Record the output statements.

[Laksanakan carta alir di Rajah 3.1 menggunakan data input yang diberikan pada Jadual 3.1. Rekodkan setiap output yang dipaparkan.]

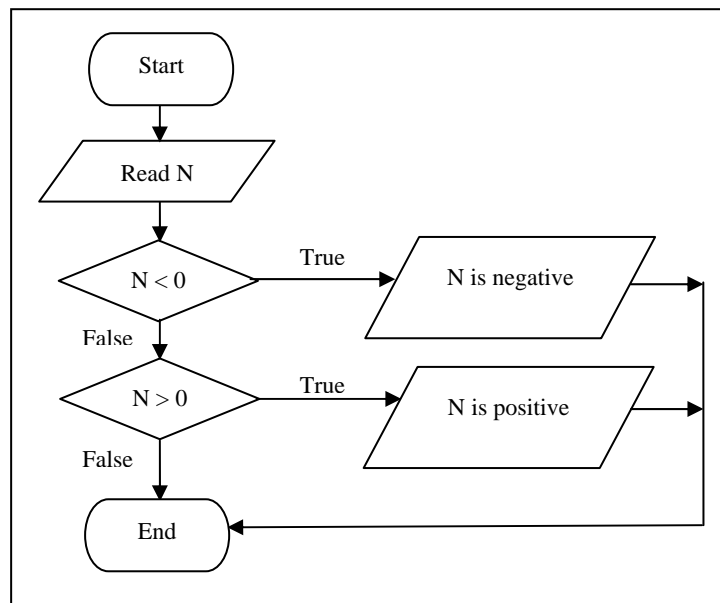


Figure 3.1

Table 3.1

Input Value	Output Statement
7	
-5	
0	
3.5	

- i. Did the output for 0 seem correct?
[Adakah output bagi 0 betul?]
 - ii. Redraw the flowchart and add another selection symbol into Figure 3.1 and an associated output to write "N is zero" when 0 is entered.
[Lukis semula carta alir tersebut dan tambah satu lagi simbol pilihan pada Rajah 3.1 dan juga simbol cetak yang bermesej "N is zero" apabila 0 dimasukkan.]
4. Figure 3.2 is a sample of output after several times of execution of the flowchart in Figure 3.3. Explain how those output were generated. Then, convert the flowchart into pseudo code.
[Rajah 3.2 merupakan contoh output yang dijana dari carta alir pada Rajah 3.3 setelah beberapa kali dilaksanakan. Terangkan bagaimana output tersebut terjana. Kemudian, tukarkan carta alir tersebut ke bentuk kod sudo.]

```

345 is odd number
100 is even number
2 is even number
    
```

Figure 3.2

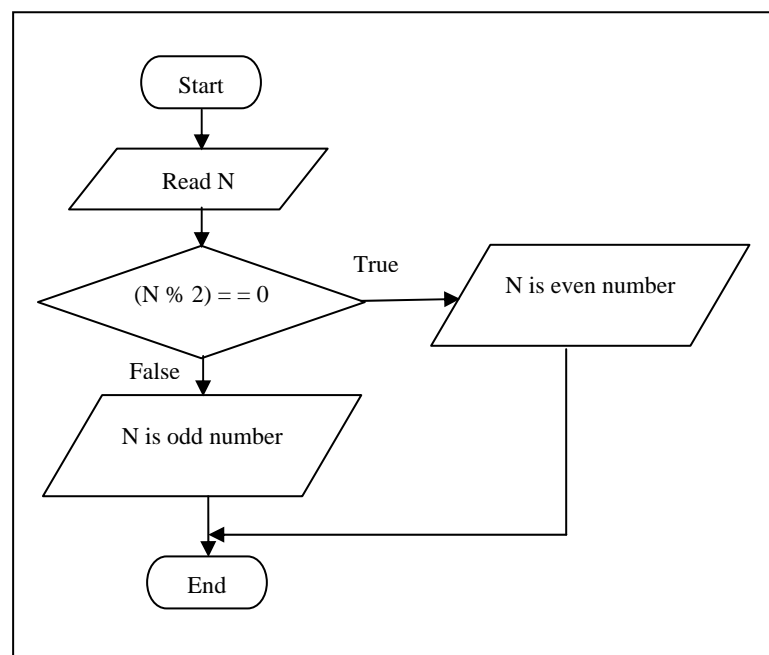


Figure 3.3

5. Trace the following pseudo code using the input values given in Table 3.2. Record the output statements.
 [Jejak kod sudo berikut menggunakan data input yang diberi pada Jadual 3.2. Rekod setiap output yang dipaparkan.]

```

Algorithm 3.1: Trace the output of nested if statement
1. Start
2. Read i
3. If (i>3.0)
    3.1 Start_If
        3.1.1 If (i<12.0)
            3.1.1.1 Start_If
                3.1.1.1.1 j = i / 100 x 5
            3.1.1.2 End_If
        3.1.2 Else
            3.1.2.1 Start_Else
                3.1.2.1.1 j = 10 / i x 2.5
            3.1.2.2 End_Else
    3.2 End_If
4. Else
    4.1 Start_Else
        4.1.1 j = 5.5 x i / 2
    4.2 End_Else
5. Display j
6. End
    
```

Table 3.2

Input Value	Output Statement
5.0	
3.0	
2.5	
15.0	
12.0	
11.5	

6. The electric company gives a discount on electricity based upon usage. The normal rate is RM0.60 per Kilowatt Hour (KWH). If the number of KWH is above 1000, then the rate is RM0.45 per KWH.
 [Syarikat elektrik telah memberikan diskaun ke atas penggunaan tenaga elektrik. Kadar normal ialah RM60 bagi setiap jam Kilowatt (KWH). Jika amaun KWH melebihi 1000, maka kadarnya ialah RM0.45 setiap KWH.]
- i. Design a pseudo code or flowchart that requires the user to enter the number of Kilowatt Hours used and then calculates and displays the total electric bill.
 [Bina kod sudo atau carta alir yang meminta pengguna memasukkan bilangan penggunaan jam Kilowatt dan seterusnya mengira serta memaparkan jumlah bil elektrik.]

- ii. According to the algorithm developed in (i), determine the cost for the following usage:
 [Berdasarkan algoritma yang dibina pada (i), tentukan kos bagi penggunaan berikut:]
 - a. 900 KWH?
 - b. 1,754 KWH?
 - c. 10,000 KWH?

- iii. Then, modify your algorithm in (i) to compute the electric bill based on the rates as stipulated in the conditions supplied by Table 3.1.
 [Seterusnya, ubahsuai algoritma (i), untuk mengira bil elektrik berdasarkan kadar penggunaan elektrik mengikut syarat-syarat pada Jadual 3.3.]

Table 3.3

Usage	Rate
0-999 KWH	\$0.60
1000-1499 KWH	\$0.45
1500-1999 KWH	\$0.40
2000 or more KWH	\$0.35

- iv. Modify your algorithm in (iii) and let's suppose that the electric rates are based upon certain rate classes within each grouping as depicted in Figure 3.4.
 [Ubahsuai algoritma (iii), serta andaikan bahawa kadar elektrik lain adalah berpandukan kepada kelas kadar tertentu dalam kumpulan yang ditunjukkan pada Rajah 3.4.]

0 - 999 KWH	
commercial rate class	RM 0.49
r1 rate class	RM 0.57
r2 rate class	RM 0.60
1000-1499 KWH	
commercial rate class	RM 0.38
r1 rate class	RM 0.43
r2 rate class	RM 0.44
r3 rate class	RM 0.45
1500 - 1999 KWH	
commercial rate class	RM 0.31
r1 rate class	RM 0.40
2000 or more KWH	
commercial rate class	RM 0.29
r1 rate class	RM 0.31
r2 rate class	RM 0.35

Figure 3.4

EXERCISE 2:

[LATIHAN 2]

1. Trace the flowchart in Figure 3.5 and answer the following questions.
[Jejak carta alir pada Rajah 3.5 dan jawab soalan-soalan berikut.]

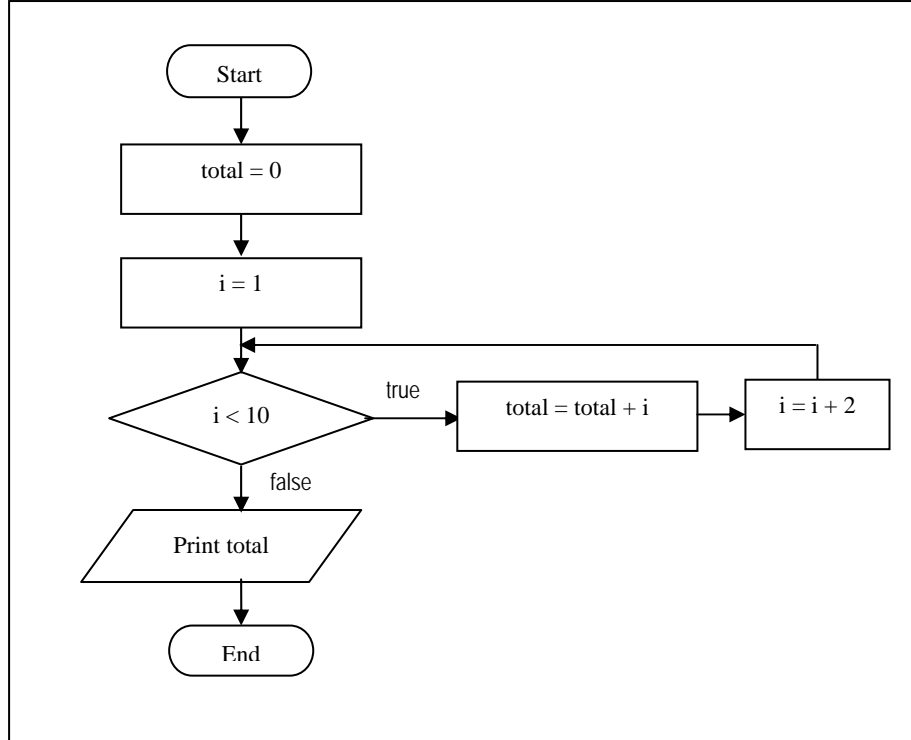


Figure 3.5

- i. Determine the output for `total`.
[Tentukan output bagi `total`.]
 - ii. How many times the loop repeats?
[Berapa kalikah gelung tersebut berulang?]
 - iii. Modify the flowchart in Figure 3.5 by using the decrement counter loop without changing the variables involved and the number of loops involved.
[Ubahsuai carta alir pada Rajah 3.5 dengan menggunakan gelung pembilang menyusut tanpa mengubah pembolehubah serta bilangan gelung yang terlibat.]
2. Trace the flowchart in Figure 3.6 and answer the following questions.
[Jejak carta alir pada Rajah 3.6 dan jawab soalan-soalan berikut.]

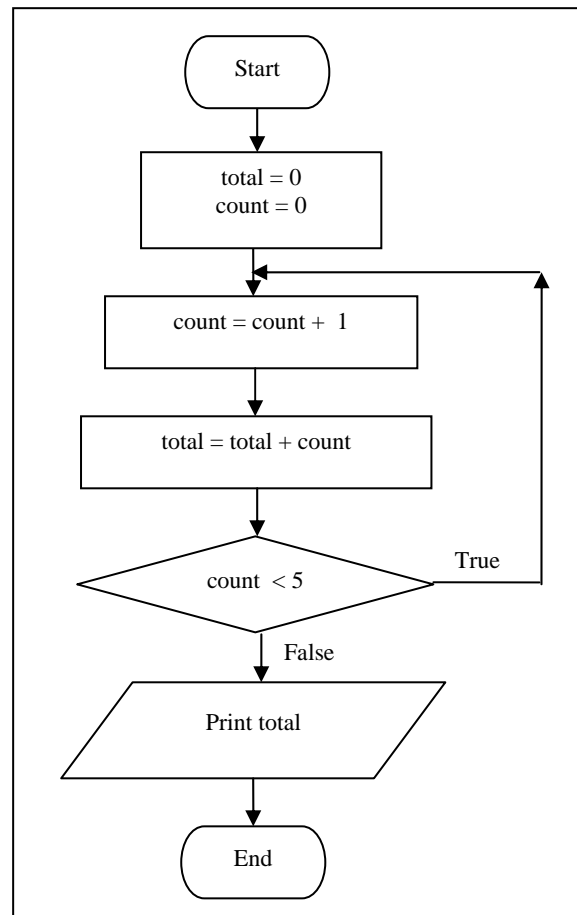


Figure 3.6

- i. Determine the output for `total`.
[*Tentukan output bagi total.*]
 - ii. How many times the loop repeats?
[*Berapa kalikah gelung tersebut berulang?*]
 - iii. Determine the output of `total` if `count = count + 1` is changed to `count = count + 3`.
[*Tentukan output bagi total jika `count = count + 1` ditukarkan kepada `count = count + 3`.*]
 - iv. Modify the flowchart in Figure 3.6 by using while loop without changing the variables and the number of loops involved.
[*Ubahsuai carta alir pada Rajah 3.6 dengan menggunakan gelung while tanpa mengubah pembolehubah serta bilangan gelung yang terlibat.*]
3. Trace the following pseudocode and record the output statements in Table 3.4.
[*Jejak kod sudo berikut dan rekod pernyataan output pada Jadual 3.4.*]

Algorithm 3.2: Loop control by sentinel value

1. Start
2. Read val
3. Set counter = 0
4. while (val > 0)
 - 4.1 start_while
 - 4.1.1 Display "The input value is" val
 - 4.1.2 counter = counter + 1
 - 4.1.3 Read val
 - 4.2 end_while
5. Display counter "values are entered"
6. End

Table 3.4

Input Value	Output Statement
1 2 3 0	
0 1 2 3	
1 2 3 4 5 6	
0	

4. Trace the following pseudocode and answer the following questions that follows.

[Jejak kod sudo berikut dan jawab soalan-soalan berikutnya.]

Algorithm 3.3: Loop control by sentinel value

1. Start
2. Set repeat = 1
3. while (repeat = 1)
 - start_while
 - Read no1
 - Read no2
 - Print no1 + no2
 - Read repeat
 - end_while
4. End

- i. Trace Algorithm 3.3 and fill in the Table 3.5 with appropriate answers for the given input.

[Jejak Algoritma 3.3 dan isikan kotak kosong pada Jadual 3.5 dengan jawapan yang betul bagi data input yang diberi.]

Input :

20	12	1
11	51	2

Table 3.5

repeat	repeat=1	no1	no2	no1+no2
1	true			

- ii. Modify Algorithm 3.3 by changing the sentinel value where the loop will be terminated if **val** is not equal to 'y' or 'Y'.
 [Ubahsuai Algoritma 3.3 dengan menukarkan nilai sentinel yang memberhentikan gelung jika **val** bukan 'y' atau 'Y'.]
5. Write a pseudo code that will generate the output as depicted in Figure 3.7.
 [Tuliskan kod sudo yang akan menjana output seperti di Rajah 3.7.]

```

1,0
2,0    2,1
3,0    3,1    3,2
4,0    4,1    4,2    4,3
    
```

Figure 3.7

EXERCISE 3:

[LATIHAN 3]

1. Given the velocity, v , the distance, d , and the time, t , of a car that accelerates from rest at constant acceleration, a .
 [Diberi halaju, v , jarak, d dan masa, t , bagi sebuah kereta yang bergerak daripada pegun kepada pecutan tetap, a .]

$$v = at$$

$$d = \frac{1}{2} at^2$$

Draw a flowchart diagram to determine v and d as every second for the first 10 seconds for a car with acceleration of $a = 1.55 \text{ m/s}^2$. The flowchart should be able to print the velocity, v and distance, d every 2 seconds.

[Lukis cartalir yang akan mengira nilai v dan d untuk setiap saat bagi tempoh 10 saat dengan pecutan $a = 1.55 \text{ m/s}^2$. Cartalir anda akan mencetak halaju, v dan jarak, d bagi setiap sela 2 saat.]

2. Construct a pseudo code or flowchart which reads an integer, n which is less than 10. Then, using nested loop, display the triangle of * such as depicted in Figure 3.8.
 [Bina kod pseudo atau cartalir yang akan membaca input integer, n di mana nilainya kurang daripada 10. Kemudian dengan menggunakan gelung tersarang, paparkan output segitiga * seperti yang ditunjukkan oleh Rajah 3.8.]

```

*   *   *   *   *
*   *   *   *
*   *   *
*   *
*

```

Figure 3.8

- Construct a pseudo code or flowchart which reads in two integer values. Then display all the numbers that is located between two numbers. If the first number entered is smaller than the second number entered, print all numbers in ascending order. On the other hand, if the first number entered is greater than the second number entered, print all numbers in descending order. Figure 3.9 depicted the example for these cases.

[Bina kod sudo atau carta alir yang akan membaca dua nilai integer. Kemudian dua nombor tadi akan dipamerkan termasuklah nombor-nombor yang terletak di antara kedua-dua nombor tersebut. Sekiranya nombor pertama yang dimasukkan adalah lebih kecil daripada nombor kedua, maka tertib paparan nombor adalah secara menaik. Sebaliknya, jika nombor pertama yang dimasukkan adalah lebih besar daripada nombor kedua, maka susunan paparan nombor adalah dalam keadaan menurun. Rajah 3.9 menggambarkan contoh bagi kes-kes ini.]

Input : 3 7 Output : 3 4 5 6 7 Input : 7 3 Output : 7 6 5 4 3
--

Figure 3.9

LAB 4: CONVERTING ALGORITHM TO C CODE

OBJECTIVES FOR THE STUDENT

1. To convert the given pseudo code into C code.
[*Untuk menukar kod sudo kepada kod C*]
2. To convert the given flow chart into C code successfully.
[*Untuk menukar carta alir kepada kod C*]
3. To compile, link and run the C program and produce the result.
[*Untuk mengkompil, memaut, dan melaksana aturcara C serta menghasilkan keputusan.*]
4. To identify error messages and debug the program.
[*Untuk mengenal pasti mesej ralat dan menyah ralat aturcara*]

ASSUMPTIONS

1. Students have the experience in developing pseudo code and flow chart.
[*Pelajar mempunyai pengalaman dalam membina kod sudo dan carta alir.*]
2. Students have the knowledge on basic structure of C programming language.
[*Pelajar mempunyai pengetahuan asas mengenai struktur asas bahasa pengaturcaraan C*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

Refer to Exercise 3 of Lab 2.

[*Rujuk pada Latihan 3 Makmal 2*]

1. Convert the pseudo code of Algorithm 2.3 into correct C code. Compile and run the program and state the output of the program.
[*Tukarkan kod sudo Algoritma 2.3 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut.*]
2. Convert the pseudo code of Algorithm 2.4 into correct C code. Compile and run the program and state the output of the program.
[*Tukarkan kod sudo Algoritma 2.4 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut.*]

3. Convert the pseudo code of Algorithm 2.5 into correct C code. Compile and run the program and state the output of the program if the `Radius` is input as:
[*Tukarkan kod sudo Algoritma 2.5 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut jika jejari input adalah:*]
 - i. 67
 - ii. 320
 - iii. 19

4. Convert the pseudo code of Algorithm 2.6 into correct C code. Compile and run the program and state the output of the program if the `Collection` is input as:
[*Tukarkan kod sudo Algoritma 2.6 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut jika Simpanan input adalah:*]
 - i. 89123
 - ii. 45560
 - iii. 12

EXERCISE 2:

[LATIHAN 2]

Refer to Exercise 4 of Lab 2

[Rujuk pada Latihan 4 Makmal 2]

1. Convert the flow chart of Figure 2.10 into correct C code. Compile and run the program and state the output of the program.
[*Tukarkan carta alir Rajah 2.10 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut.*]

2. Convert the flow chart of Figure 2.11 into correct C code. Compile and run the program and state the output of the program if the input for the variables are as follows:
[*Tukarkan carta alir Rajah 2.11 ke dalam kod C dengan betul. Kompil dan larikan aturcara dan nyatakan output bagi aturcara tersebut jika input pembolehubah adalah seperti berikut:*]
 - i. ItemName: **Pencil**
PricePerUnit: **0.50**
Quantity: **15**

 - ii. ItemName: **Note book**
PricePerUnit: **2.00**
Quantity: **5**

- iii. ItemName: **Pencil**
PricePerUnit: **3**
Quantity: **1.20**

EXERCISE 3:
[LATIHAN 3]

Given the following C program, compile and run the program. Write down any syntax error and logic error that occurs. Rewrite the program to correct the error. [Diberi aturcara C di bawah, kompil dan larikan aturcara tersebut. Tuliskan semula aturcara untuk membetulkan sintaks.]

1.

```
1 //Program 4.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main(){
6     Printf("I like programming ");
7     Getch();
8     Return 0;
9 }
```

2.

```
1 //Program 4.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main (){
6     int nilai1;
7     nilai1 = 4;
8     printf("%d+%d=%d", nilai1, nilai2,
9                                     nilai1+nilai2);
10    getch();
11    return 0;
12 }
```

3.

```
1 //Program 4.3
2 #include <stdio.h>
3 #include <conio.h>
4 void main ( )
5 {
6     int 125umur
7     125umur = 125;
8     printf "umur= %d tahun", umur);
```

4.

```
1 //Program 4.4
2 #include <stdio.h>
3 #include <conio.h>
4 int main()
5 {
6     printf("This is my simple program");
7     getch();
8     return 0;
9 }
```

5.

```
1 //Program 4.5
2 #include <stdio.h>
3 #include <conio.h>
4 void my_address (void)
5 {
6     printf("Dayang Norhayati Abang Jawawi\n");
7     printf("\tFakulti Sains Komputer");
8     printf("dan Sistem Maklumat\n");
9     printf("\tSkudai, Johor\n");
10 }
11
12 void main ( )
13 {
14     print_my_address();
15     getch();
16 }
```

6.

```
1 //Program 4.6
2 #include <stdio.h>
3 #include <conio.h>
4 int main (){
5     char c1, c2, c3, c4;
6     c1=65; c2='A'; c3=0x41; c4=0101;
7     printf("%c %d %f %s", c1, c2, c3, c4);
8     getch();
9     return 0;
}
```

LAB 5: WORKING WITH DATA

OBJECTIVES FOR STUDENTS

1. Identify the difference between integer and floating-point numbers.
[*Mengenal pasti perbezaan di antara nombor integer dan titik terapung.*]
2. Write arithmetic expressions in C.
[*Menulis ungkapan aritmetik dalam C.*]
3. Define and initialize variables and constants in C.
[*Mentakrifkan dan memberi nilai awal kepada pembolehubah dan pemalar di dalam C.*]
4. Write simple programs that assign values to variables, process the values and display the results.
[*Menulis satu aturcara yang mudah yang dapat mengumpukkan nilai kepada pembolehubah, memproses nilai-nilai tersebut dan memaparkan hasil operasi tersebut.*]

ASSUMPTIONS

1. Students have experience in compiling and running a C program.
[*Pelajar mempunyai pengalaman mengkompil dan melaksanakan aturcara C.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Write assignment statements that perform the following operations with variables a, b and c.
[*Tulis pernyataan umpukan yang menjalankan operasi berikut terhadap pembolehubah a, b dan c.*]
 - i. Add 2 to variable a and stores the result in b.
[*Tambah 2 kepada a dan simpan hasilnya dalam b.*]
 - ii. Multiply b with 4 and stores the result in b.
[*Darab b dengan 4 dan simpan hasilnya dalam b.*]

- iii. Divide a by 3.14 and stores the result in b.
[Bahagi a dengan 3.14 dan simpan hasilnya dalam b.]
- iv. Subtract 8 from b and stores the result in a.
[Tolak 8 daripada b dan simpan hasilnya dalam a.]
- v. Store the value 27 in a.
[Simpan nilai 27 dalam a.]
- vi. Store the character 'L' in c.
[Simpan huruf 'L' dalam c.]

EXERCISE 2:**[LATIHAN 2]**

1. Download Program 5.1. There are a number of syntax errors in the Program 5.1. Locate as many as you can.
[Muat turun Program 5.1. Terdapat beberapa ralat sintaks di dalam Program 5.1. Kenalpasti sebanyak yang mungkin.]

```
1 //Program 5.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int a,b;
8     a=3;
9     b=4;
10    c=a+b;
11    printf ("The value of c is %d, c");
12    getch();
13    return 0;
14 }
```

2. Explain and correct the errors.
[Berikan penerangan tentang ralat sintaks yang dikenal pasti dan betulkan ralat-ralat tersebut.]

EXERCISE 3:
[LATIHAN 3]

1. Type in the following program (Program 5.2):
[*Taip aturcara yang berikut (Program 5.2):*]

```
1 //Program 5.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int length, width, area;
8
9     length = 12;
10    width = 13;
11    area=length*width;
12    printf ("Area: %d", area, "\n");
13    getch();
14    return 0;
15 }
```

2. Compile and run the Program 5.2. What is the output?
[*Kompil dan laksanakan Program 5.2. Apakah hasilnya?*]
3. In your own words, explain the program.
[*Di dalam ayat kamu sendiri, berikan penerangan tentang aturcara tersebut.*]
4. Now, modify the Program 5.2 by changing the value of `length` to 12.12 and `width` to 13.12. Compile and run the modified program. What is the output? Explain the output.
[*Ubah aturcara di dalam Program 5.2 dengan menukarkan nilai `length` kepada 12.12 dan `width` kepada 13.12. Kompil dan laksanakan aturcara yang telah diubahsuai. Apakah hasilnya. Berikan penerangan tentang hasil tersebut.*]
5. Now, change the `length` and `width` variable type to `float`. Compile and run the modified program. What is the output?
[*Ubah pembolehubah `length` dan `width` kepada jenis `float`. Kompil dan laksanakan aturcara yang telah diubahsuai. Apakah hasilnya?*]

EXERCISE 4:**[LATIHAN 4]**

1. What will be printed by the extracted code below. Explain how you get such output.

[Apakah yang akan dicetak oleh cebisan kod di bawah. Terangkan bagaimana kamu boleh mendapat hasil tersebut.]

```
1 //Program 5.3
2
3     int n, m, x, y;
4     m=10;
5     n=m++*2;
6     printf ("n: %d", n);
7     printf ("\nm: %d", m);
8
9     x=10;
10    y=++x*2;
11    printf ("\nx: %d", x);
12    printf ("\ny: %d", y);
```

2. What is the final value of t ? Why such output? Explain the output.

[Apakah hasil akhir nilai t ? Kenapa hasil tersebut dihasilkan? Berikan penerangan tentang hasil tersebut.]

```
1 //Program 5.4
2
3     int j=2, k=3;
4     double r=24.5, s=3.0, t1, t2;
5
6     t1*=r+s;
7     t2=r-s*3%(2+j)/k;
```


EXERCISE 5:

[LATIHAN 5]

1. Convert the following pseudocode (Algorithm 5.1) to C code.
[*Tukarkan kod sudo berikut (Algorithm 5.1) ke kod C.*]

Algorithm 5.1: Calculating total sales value for three items

1. Start
2. Store 12.95 in the item1 variable.
3. Store 24.95 in the item2 variable.
4. Store 6.95 in the item3 variable.
5. Sum up the three variables and store the result in the subtotal variable.
6. Multiply the subtotal variable with 0.06 and store the result in the sales tax variable.
7. Subtract the sales tax amount from the subtotal and store the result in the total variable.
8. Display each item's value, the subtotal value, the amount of sales tax and the total.
9. End

2. Convert either Algorithm 2.1 or Algorithm 2.2 (Lab 2, Exercise 1, Question 2) to C code.
[*Tukarkan samada Algorithm 2.1 atau Algorithm 2.2 (Lab 2, Exercise 1, Question 2) kepada kod C.*]
3. Convert the designed solution for "Get the input values" (Lab 2, Exercise 6, Question 1) to C code.
[*Tukarkan penyelesaian yang telah direkabentuk bagi "Get the input values" (Lab 2, Exercise 6, Question 1) kepada kod C.*]
4. Convert the designed solution for "Calculate the total perimeter" (Lab 2, Exercise 6, Question 1) to C code.
[*Tukarkan penyelesaian yang telah direkabentuk bagi "Calculate the total perimeter" (Lab 2, Exercise 6, Question 1) kepada kod C.*]

EXERCISE 6:**[LATIHAN 6]**

1. Write a program that computes the tax and tip of a restaurant bill for a patron with a cost of RM50.60. The tax should be 10% of the meal cost. The tip should be 5% of the total after adding the tax. Display the cost, tax amount, tip amount and total bill on the screen.
[Tuliskan satu aturcara yang akan mengira nilai cukai dan tip untuk bil restoran yang mempunyai nilai kos RM50.60. Cukai ialah 10% daripada kos tersebut. Tip pula ialah 5% daripada jumlah kos setelah dicampur dengan cukai. Paparkan nilai kos, nilai cukai, nilai tip dan jumlah bil di skrin.]
2. An electronics company sells circuit boards at a 40 percent profit. Write a program that will calculate the selling price of a circuit board that costs RM12.67. Display the result on the screen.
[Sebuah syarikat elektronik menjual papan litar pada keuntungan 40%. Tuliskan satu aturcara yang akan mengira harga jualan papan litar yang berkos RM12.67. Paparkan hasil di skrin.]

LAB 6: INPUT & OUTPUT

OBJECTIVES FOR THE STUDENT

1. To read user input from keyboard and display program output.
[*Untuk membaca masukan pengguna daripada papan kekunci dan memaparkan hasil aturcara.*]
2. To write simple programs that read numbers and text, process the input and display the results.
[*Untuk menulis sebuah aturcara mudah yang dapat membaca nombor-nombor dan teks, memproses data masukan dan memaparkan hasil-hasil proses tersebut.*]
3. To format output.
[*Untuk memformat output.*]

ASSUMPTIONS

1. Students have experience in compiling and running a C program.
[*Pelajar mempunyai pengalaman mengkompil dan melaksanakan aturcara C.*]
2. Students have a basic knowledge of previous lessons (Data Types and Arithmetic).
[*Pelajar mempunyai pengetahuan asas tentang pelajaran yang terdahulu (Jenis Data dan Aritmetik).*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Write a C statement that performs the following operations:
[*Tulis pernyataan C yang menjalankan operasi-operasi berikut*]
 - i. Declare variables; `weight` and `height` of float type.
[*Istiharkan pembolehubah; `weight` dan `height` berjenis float.*]
 - ii. Declare a `counter` of type integer with an initial value of 0.
[*Istiharkan pembolehubah `counter` berjenis integer dengan nilai awalan 0.*]
 - iii. Stores the value 180.5 to the variable.

[Simpan nilai 180.5 kepada pembolehubah *weight*.]

- iv. Add a value of 1 to the `counter` variable and store the result to the `counter` variable.
[Tambah nilai 1 dalam pembolehubah `counter` dan simpan hasil dalam pembolehubah `counter`.]
- v. Declare variables; `weight_reduced` and `new_weight` of floating type.
[Istiharkan pembolehubah; `weight_reduced` dan `new_weight` berjenis `float`.]
- vi. Read the `weight_reduced` variable from a keyboard.
[Baca pembolehubah `weight_reduced` daripada papan kekunci.]
- vii. Calculate the value of `new_weight` as a difference between the `weight` and the `weight_reduced` values.
[Kira nilai `new_weight` sebagai perbezaan di antara nilai `weight` dan `weight_reduced` .]
- viii. Display the value of `new_weight` on the screen.
[Paparkan nilai `new_weight` pada skrin.]
- ix. Display the statement; “Congratulations! You have reduced your weight” on the screen.
[Paparkan pernyataan, “Congratulations! You have reduced your weight” pada skrin.]

EXERCISE 2:

[LATIHAN 2]

1. Type in the following 2 programs (Program 6.1 and Program 6.2):
[Taip 2 aturcara berikut (Program 6.1 dan Program 6.2):]

1	//Program 6.1
2	#include <stdio.h>
3	#include <conio.h>
4	
5	int main ()
6	{
7	char char1;
8	char char2;
9	char char3;
10	char char4;
11	printf("Input four characters > ");
12	scanf ("%c%c%c%c", &char1, &char2, &char3,
13	&char4);
14	printf("%c %c %c %c", char1, char2, char3,
15	char4);

```

16     getch();
17     return 0;
18 }

```

```

1 //Program 6.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main ()
6 {
7     char  char1;
8     char  char2;
9     char  char3;
10    char  char4;
11    printf("Input four characters > ");
12    char1=getchar();
13    char2=getchar();
14    char3=getchar();
15    char4=getchar();
16    putchar(char1);
17    putchar(char2);
18    putchar(char3);
19    putchar(char4);
20    getch();
21    return 0;
22 }

```

2. Compile and run them with the same set of the input data. e.g. (a b c d). Examine the results carefully.
[*Kompil dan laksanakan aturcara-aturcara tersebut dengan set data input yang sama (contoh a b c d). Semak hasil tersebut secara teliti.]*
3. Understand what each line of the statement in the program does. In your own words, explain what the program does and the difference between the two programs.
[*Fahamkan apa yang dilakukan oleh setiap baris di pernyataan dalam aturcara tersebut. Di dalam ayat kamu sendiri, terangkan apa yang dilakukan oleh setiap aturcara tersebut lakukan dan perbezaan di antara kedua-dua aturcara tersebut.*]

EXERCISE 3:

[LATIHAN 3]

1. Write a C program that asks the user to enter an integer, a floating point number, and a character, and write the results back out. All output must be in the format shown in the sample output.
[*Tulis satu aturcara C yang meminta pengguna memasukkan satu nilai integer, satu nilai titik terapung dan satu aksara, dan paparkan semula nilai-nilai*

yang telah dimasukkan. Semua hasil mestilah dicetak menggunakan format seperti contoh output di bawah.]

Sample input:

Enter an integer: 7
Enter a decimal number: 2.25
Enter a single character: r

Sample output:

You entered 7
You entered 2.25
You entered r

2. Write a C program that will read your yearly gross wage from a keyboard, calculate and display your monthly net wage. Assuming your yearly gross wage is between RM50,000 and RM80,000. Below are the deductions that need to be considered:

- i. Yearly income tax: 10% from the gross wage.
- ii. CPF: 8% from the first RM45000 of the gross wage and 4% from the rest.

[Tulis satu aturcara C yang akan membaca gaji kasar setahun daripada papan kekunci, mengira dan memaparkan pendapatan bersih sebulan. Anggapkan yang gaji kasar anda setahun ialah di antara RM50,000 dan RM80,000. Berikut ialah potongan yang perlu dibuat:

- i. Cukai pendapatan setahun: 10% daripada gaji kasar.
- ii. CPF: 8% daripada RM45000 yang pertama daripada gaji kasar dan 4% selebihnya.]

EXERCISE 4:

[LATIHAN 4]

1. Type in Program 6.3 as follows:
[Taip Program 6.3 seperti berikut:]

```

1 //Program 6.3
2 #include <stdio.h>
3
4 int main()
5 {
6
7     float val1, val2, val3, val4;
8     FILE *inData;
9     FILE *outData;
10
11     inData = fopen("DataIn.txt", "r");
12     outData = fopen("DataOut.txt", "w");
13

```

```

14     fscanf(inData,"%f%f%f%f", &val1, &val2,
15         &val3, &val4);
16     fprintf(outData,"%f %f %f %f", val1, val2,
17         val3, val4);
18
19     return 0;
20 }

```

2. Create the input data file DataIn.txt. The file consists of the following data:
[Cipta satu fail input DataIn.txt. Fail tersebut mengandungi data-data berikut:]

```

5.5
6.6
7.7
8.8

```

3. Compile and run Program 6.3. Examine the results carefully. Understand what each line of the statement in the program does. In your own words, explain what the program does.
[Kompil dan laksanakan Program 6.3. Kaji hasilnya dengan teliti. Fahamkan apa yang dilakukan oleh setiap baris pernyataan di dalam aturcara tersebut. Di dalam ayat kamu sendiri, terangkan apa yang aturcara tersebut laksanakan.]
4. Modify the program so it also outputs the average of the four input numbers.
[Ubahsuai Program 6.3 aturcara tersebut supaya ia menghasilkan purata keempat-empat nombor tersebut.]

EXERCISE 5:

[LATIHAN 5]

1. Type in Program 6.4 as follows:
[Taip Program 6.4 seperti berikut:]

```

1 //Program 6.4
2 #include <stdio.h>
3 #include <conio.h>
4 int main()
5 {
6     int n;
7     float f;
8     char s[100];
9
10    printf("Input one integer: ");
11    scanf("%d", &n);
12
13    printf("%d\n%6d\n%-6dend\n", n, n, n, n);
14

```

```

15     printf("Input one string: ");
16     scanf("%s", s);
17     printf("%s\n%20s\n%-20send\n", s,s,s);
18
19
20     printf("Input one floating number: ");
21     scanf("%f", &f);
22     printf("%f\n%5.2f\n%-10.2eend\n", f, f, f);
23
24     getch();
25     return 0;
26 }

```

2. Compile and run Program 6.4. Examine the results carefully. Understand what each line of the statement in the program does. In your own words, explain what the program does.

[Kompil dan laksanakan Program 6.4. Periksa hasilnya dengan teliti. Fahamkan apa yang dilakukan oleh setiap baris arahan di dalam aturcara tersebut lakukan. Di dalam ayat kamu sendiri, terangkan apa yang dilakukan oleh aturcara tersebut.]

EXERCISE 6:

[LATIHAN 6]

1. Write a program that will convert Malaysian Ringgit (RM) amounts to Japanese Yen and to Euros. The conversion factors to use are:

1 RM = 0.21734 Euros

1 RM = 36.0665 Yen

Format your currency amounts in fixed-point notation, with two decimal places of precision, and be sure the decimal point is always displayed.

[Tulis satu aturcara yang akan menukarkan nilai Ringgit Malaysia (RM) ke Yen Jepun dan Euros. Nilai penukaran adalah seperti berikut:

1 RM = 0.21734 Euros

1 RM = 36.0665 Yen

Nilai matawang tersebut mestilah berjenis titik terapung, dengan dua tempat perpuluhan. Pastikan titik perpuluhan tersebut dipaparkan.]

2. The monthly payment on a loan may be calculated by the following formula:
[Bayaran bulanan untuk sebuah pinjaman boleh dikira dengan menggunakan rumus berikut:]

$$\text{Payment} = \frac{\text{Rate} * (1+\text{Rate})^N}{((1+\text{Rate})^N - 1)} * L$$

Rate is the monthly interest rate, which is the annual interest rate divided by 12 (e.g. 12% annual interest would be 1 percent monthly interest). N is the number of payments and L is the amount of the loan. Write a program that asks for these values and displays a report as follows:

[Rate merujuk kepada kadar interest bulanan, iaitu kadar interest tahunan dibahagi dengan 12 (contohnya, kadar interest bulanan untuk 12% kadar interest tahunan ialah 1%). N ialah bilangan pembayaran dan L ialah jumlah pinjaman. Tulis satu aturcara yang akan membaca nilai-nilai tersebut dan memaparkan report seperti format di bawah:]

Loan Amount:	RM10000.00
Monthly Interest Rate:	1%
Number of Payments:	36
Monthly Payment:	RM 332.14
Amount Paid Back:	RM11957.15
Interest Paid:	RM 1957.15

LAB 7: SELECTION OR DECISION

OBJECTIVES FOR STUDENTS

1. Use selection control structure.
[*Menggunakan struktur kawalan pilihan.*]
2. Use the `if` selection control structure.
[*Menggunakan struktur kawalan pilihan `if`.*]
3. Use the `if...else` selection control structure.
[*Menggunakan struktur kawalan pilihan `if...else`.*]
4. Use the `switch` selection control structure.
[*Menggunakan struktur kawalan pilihan `switch`.*]

ASSUMPTIONS

1. Students have knowledge to assess relational and logical Boolean operations.
[*Pelajar mempunyai pengetahuan untuk menilai operasi Boolean berbentuk hubungan dan logik.*]
2. Students have knowledge to form input/output statements.
[*Pelajar mempunyai pengetahuan untuk membina kenyataan input/output.*]
3. Students have knowledge to form increment/decrement operators.
[*Pelajar mempunyai pengetahuan untuk membina operasi `++`/`--`.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Given the following declarations, evaluate the expression below and assign `true` or `false` value for each expression.
[*Diberi pengisytiharan berikut, selesaikan ungkapan di bawah serta berikan nilai `true` atau `false` untuk setiap ungkapan.*]

```
bool found = true;
bool flag = false;
int number = 5;
int xint = 12;
int yint = 100;
double xdouble = 50.3;
```

```
double ydouble = 1.3;
char ch = 'D';
```

- i. !found
- ii. !number
- iii. (number >= xint) && (number<=yint)
- iv. (found || flag)
- v. (xint != yint)
- vi. (ydouble * 10) <= xdouble
- vii. 'A'<= ch && ch<='Z'
- viii. number-5 == xint) && !flag

2. Write a C expression for each the following conditions.

[Tuliskan ungkapan C bagi setiap syarat berikut:]

- i. Number greater than 1 and smaller than 9.
[Nombor lebih besar daripada 1 tetapi lebih kecil daripada 9.]
- ii. Variable ch not character 'q' or 'k'.
[Pembolehkan ch bukan huruf 'q' atau 'k'.]
- iii. Number between 1 to 9 but not number 5.
[Nombor adalah di antara 1 hingga 9 tetapi bukan nombor 5.]
- iv. Not the number between 1 to 9.
[Nombor bukan di antara 1 hingga 9.]

EXERCISE 2:

[LATIHAN 2]

1. Download Program 7.1. Study the code and run the program.
[Muat turun Program 7.1. Fahamkan aturcara tersebut dan laksanakan.]

```
1 //Program 7.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main(){
6     double mark;
7
8     printf ("Enter your mark: ");
9     scanf ("%lf", &mark);
10
11     if (mark >= 30)
12         printf ("TEST 1 -> Pass\n");
13     if (mark < 30)
14         printf ("TEST 1 -> Fail\n");
15
16     getch();
17     return 0;
18 }
```

- i. Explain the function or objective of the program.
[*Terangkan fungsi atau objektif aturcara.*]
 - ii. Change the usage of the 2 if statements (Line 11 and Line 13) to an if..else statement.
[*Tukarkan penggunaan 2 kenyataan if (Baris 11 dan Baris 13) kepada kenyataan if..else.*]
 - iii. Show the difference between the usage of the 2 if statements and the if..else statement using a flowchart.
[*Tunjukkan perbezaan penggunaan 2 kenyataan if dan if..else dengan menggunakan carta alir.*]
2. Download Program 7.2. The program requests the user to input Test 1 mark. Contribution to the final mark will be calculated if the Test 1 mark is greater than or equal to 30%, and if the mark is less than 30% the student is required to re-sit the test. The contribution is 20% of Test 1 mark to the final mark.
[*Muat turun Program 7.2. Program tersebut meminta pengguna memasukkan markah Test 1. Sumbangan kepada markah akhir akan dikira jika markah Test 1 yang dimasukkan adalah lebih besar atau sama dengan 30%, dan jika kurang daripada 30% pelajar perlu menduduki Test 1 sekali lagi. Test 1 memberi sumbangan 20% kepada markah akhir.*]

```

1 //Program 7.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main(){
6 double mark, final20p=0;
7
8     printf ("Enter your mark: ");
9     scanf ("%lf", &mark);
10
11     if (mark >= 30)
12         printf ("TEST 1 -> Pass\n");
13         final20p = ((20.0/100.0) * mark);
14         printf ("Contribution to final mark %d",
15 final20p);
16     if (mark < 30)
17         printf ("TEST 1 -> Fail\n");
18         printf ("Please re-sit TEST 1\n");
19     getch();
20     return 0;
21 }
```

- i. Identify the logic errors from the program.
[*Kenal pasti ralat logik dari program tersebut.*]
- ii. Correct the logic errors.
[*Betulkan ralat logik tersebut.*]

EXERCISE 3:**[LATIHAN 3]**

1. Download Program 7.3. Study the code and run the program.
[Muat turun Program 7.3. Fahamkan aturcara tersebut dan laksanakanannya.]

```
1 //Program 7.3
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main(){
6     double mark;
7
8     print ("Enter your mark");
9     scanf ("%lf", &mark);
10
11     if (mark >=75)
12         printf ("Your score: A\n");
13     if ((mark <75)&& (mark>=60))
14         printf ("Your score: B\n");
15     if ((mark <60)&& (mark>=45))
16         printf ("Your score: C\n");
17     if ((mark <45)&& (mark>=30))
18         printf ("Your score: D\n");
19     if (mark <30)
20         printf ("Your score: E\n");
21     getch();
22     return 0;
23 }
```

- i. Explain the function or objective of the program.
[Terangkan fungsi atau objektif aturcara tersebut.]
 - ii. Draw a flowchart for the program.
[Lukiskan carta alir untuk program ini.]
 - iii. Given the input `mark = 90.6`, identify which `if` statement will be evaluated and what are the results of the evaluated expression.
[Diberi input `mark = 90.6`, kenal pasti garis kenyataan `if` yang akan dinilai dan keputusan penilaian ungkapan tersebut.]
2. Download Program 7.4. Study the code and run the program.
[Muat turun Program 7.4. Fahamkan aturcara tersebut dan laksanakanannya.]

```
1 //Program 7.4
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main(){
6     double mark;
7
8     print ("Enter your mark");
9     scanf ("%lf", &mark);
```

```

10
11  if (mark >=75)
12      printf ("Your score: A\n");
13  else if ((mark <75)&& (mark>=60))
14      printf ("Your score: B\n");
15  else if ((mark <60)&& (mark>=45))
16      printf ("Your score: C\n");
17  else if ((mark <45)&& (mark>=30))
18      printf ("Your score: D\n");
19  else if (mark <30)
20      printf ("Your score: E\n");
21  getch();
22      return 0;
23  }

```

- i. Draw a flowchart for the program.
[Lukiskan carta alir untuk aturcara ini.]
- ii. Given the input mark = 90.6, identify which if statement will be evaluated and what are the results of the evaluated expression.
[Diberi input mark = 90.6, kenal pasti garis kenyataan if yang akan dinilai dan keputusan penilaian ungkapan tersebut.]
- iii. Explain the difference between the if statement in Program 7.3 with the if..else..if statement in Program 7.4.
[Jelaskan perbezaan kenyataan if dalam Program 7.3 dan kenyataan if..else..if dalam Program 7.4.]

EXERCISE 4:**[LATIHAN 4]**

1. Given a selection control structure with switch in Program 7.5.
[Diberikan struktur kawalan pilihan switch dalam Program 7.5.]

```

1 //Program 7.5
2 #include <studio.h>
3 #include <conio.h>
4
5 int main(){
6     int choice;
7
8     printf("What flavor ice cream do you want?\n");
9     printf("Enter 1 for chocolate\n");
10    printf("Enter 2 for vanilla\n");
11    printf("Enter 3 for strawberry\n");
12    printf("Enter 4 for Yam\n");
13    printf("Enter your choice: \n");
14    scanf("%d", &choice);
15
16    switch (choice) {
17        case 1:

```

```
18     printf("Chocolate, Muhammad's favorite.\n");
19     break;
20     case 2:
21         printf("Vanilla, Ismael's favorite\n");
22         break;
23     case 3:
24         printf("Strawberry, Adibah's favorite\n");
25         break;
26     case 4:
27         printf("Yam, Munirah's favorite\n");
28         break;
29     default:
30         printf("We don't have any\n");
31         printf("Make another selection\n");}
32
33     getch();
34     return 0;
35 }
```

- i. Draw a flowchart for the program.
[Lukiskan carta alir untuk aturcara ini.]
- ii. Based on the flowchart rewrite Program 7.5 using selection control structure with `if..else..if`.
[Berdasarkan carta alir yang dilukis, tulis semula Program 7.5 dengan menggunakan struktur kawalan pilihan `if..else..if`.]
- iii. Compare the original Program 7.5 and the program from answer ii. Which selection control structure is easy to read?
[Bandingkan Program 7.5 asal dan aturcara dari jawapan ii. Struktur kawalan pilihan yang manakah lebih mudah untuk dibaca?]

EXERCISE 5:

[LATIHAN 5]

1. Given flowchart in Figure 7.1, write a C program.
[Diberikan carta alir di Rajah 7.1, tuliskan aturcara C.]

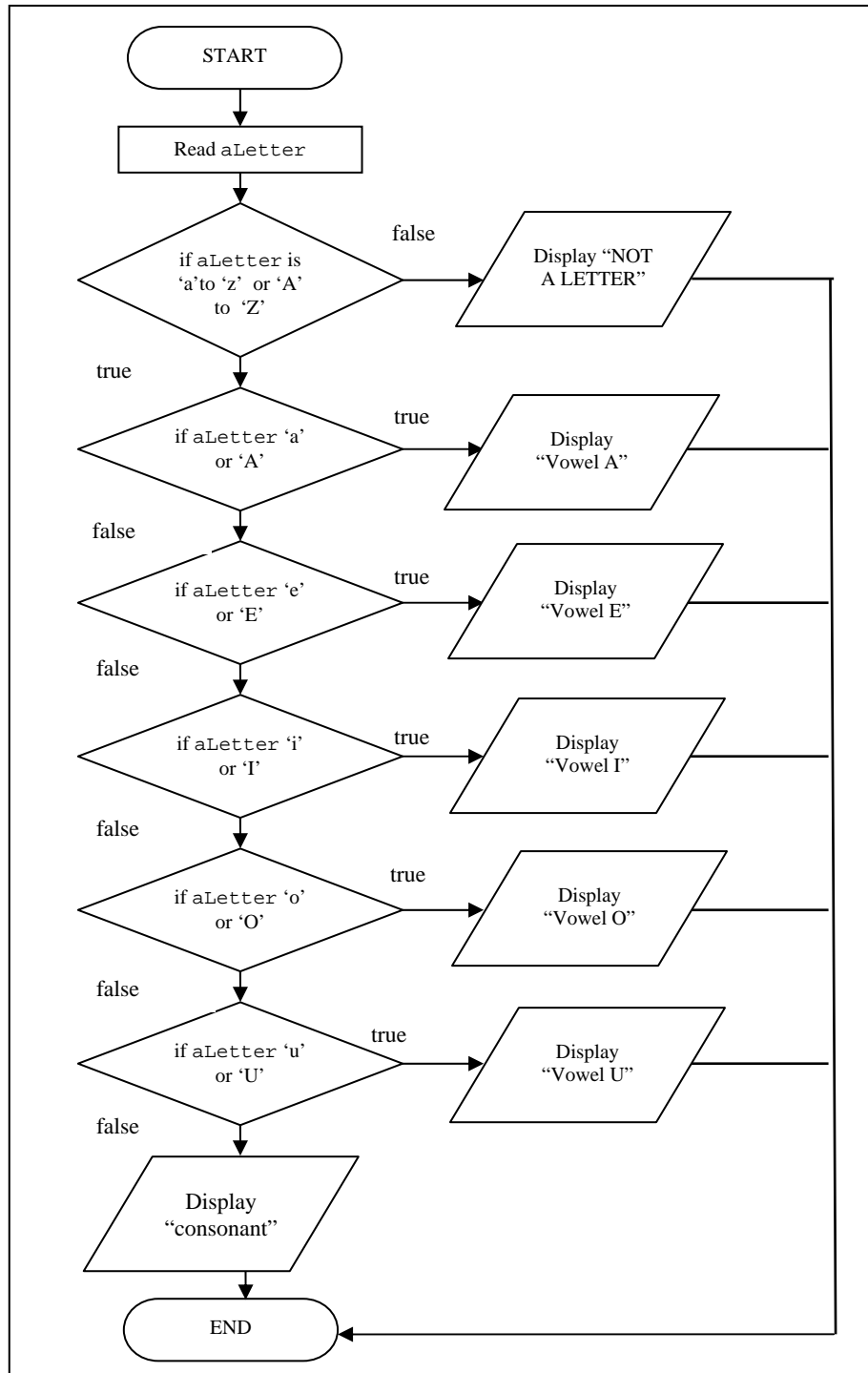


Figure 7.1

EXERCISE 6:**[LATIHAN 6]**

1. Write a complete C program to identify the grade for the subject C Programming Techniques (SCP1103) based on the assessments as listed in Table 7.1. The input for this program is the 5 assessment marks. The full mark for each assessment is 100%. The final mark for this subject will be calculated based on the 5 assessments input mark. Based on this final mark the program will assign a grade. The grade scales are as tabulated in Table 7.2.
[Tuliskan aturcara C lengkap untuk mengenal pasti gred bagi subjek Teknik Pengaturcaraan C (SCP1103) berdasarkan penilaian yang disenaraikan di Jadual 7.1. Input untuk aturcara ini ialah markah bagi 5 penilaian. Markah penuh untuk setiap penilaian ialah 100%. Markah akhir untuk subjek ini dikira berdasarkan 5 masukan markah penilaian tersebut. Markah ujian akhir ini akan digunakan untuk menentukan gred. Skala gred dijadualkan di Jadual 7.2.]

Table 7.1: Assessment Items

Assessments	Percentage (%)
Test 1	20
Assignments	30
Quiz	10
Lab works	10
Final Exam	30
Total	100

Table 7.2: Grade Scale

Assessments	Percentage (%)
A+	90-100
A	80-89
A-	75-79
B+	70-74
B	65-69
B-	60-64
C+	55-59
C	50-54
C-	45-49
D+	40-44
D	35-39
D-	30-34
E	00-29

EXERCISE 7:**[LATIHAN 7]**

1. Write a C program to compute bill payment of electrical consumption for different users. In order to promote saving, Tenaga National Berhad (TNB) charge a lower rate to users with lower consumption as shown in Table 7.3. Figure 7.2(a) and Figure 7.2(b) show a design of flowchart to solve this problem.

[Tuliskan satu aturcara C yang akan mengira jumlah bayaran bil penggunaan elektrik bagi setiap pengguna. Untuk menggalakkan penjimatan, pihak Tenaga National Berhad (TNB) mengenakan kadar bayaran lebih rendah kepada pengguna bagi penggunaan yang minima seperti di Jadual 7.3. Rajah 7.2(a) and Rajah 7.2(b) menunjukkan carta alir untuk menyelesaikan masalah ini.]

Table 7.3: New Rate

User Code [Kod Pengguna]	User Type [Jenis Pengguna]	First Rate [Kadar Pertama]	Second Rate [Kadar Kedua]
R	Household [Rumah Tangga]	0.10 for first 500 unit	0.25 for the following units
P	Office [Pejabat]	0.25 for first 1000 unit	0.50 for the following units
K	Factory [Kilang]	0.50 for first 1500 unit	0.75 for the following units

The following choice is used to identify the type of user.

[Tentukan jenis pengguna melalui pilihan berikut.]

User Code:

1. Household
2. Office
3. Factory
4. Program End

Choice: _

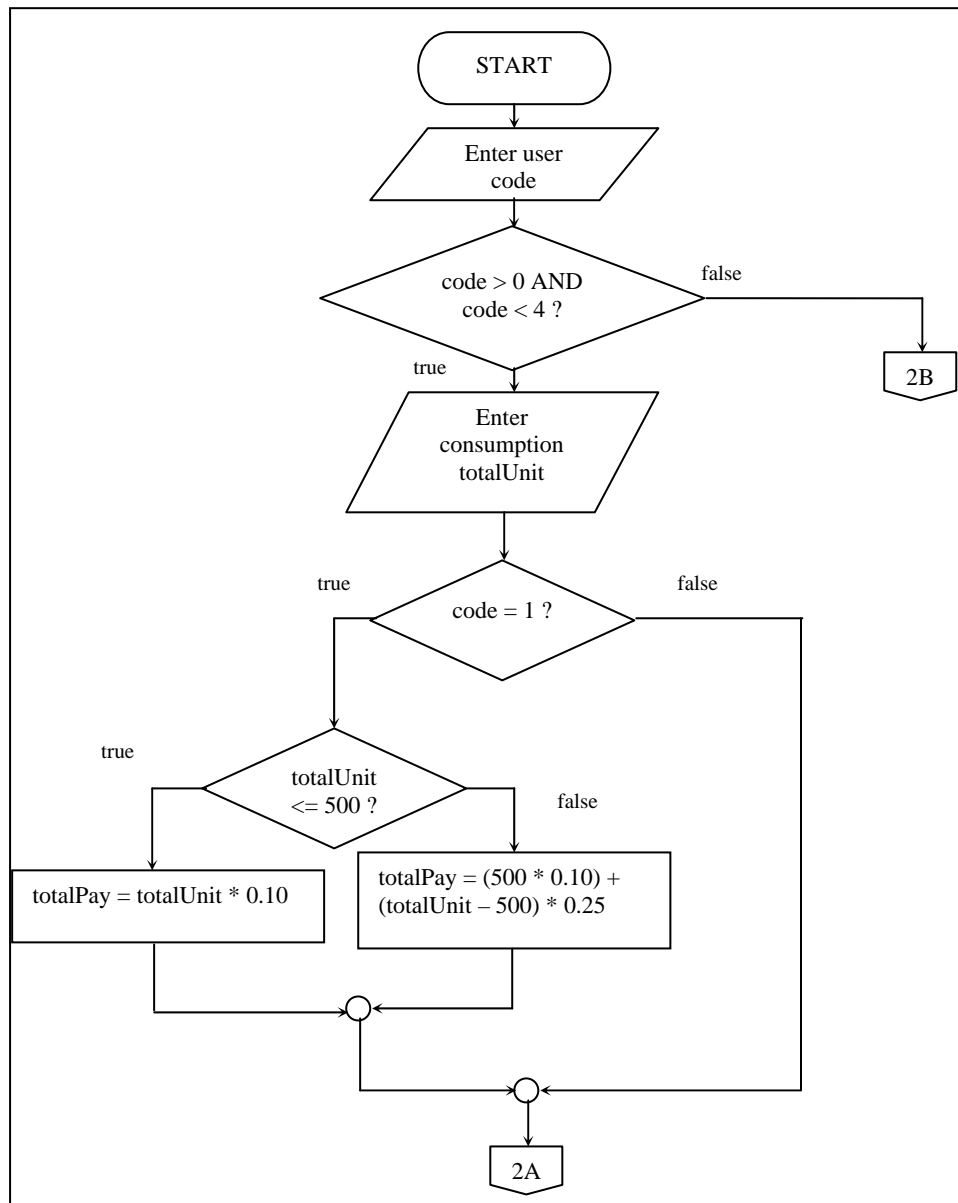


Figure 7.2(a): Flow Chart Part I

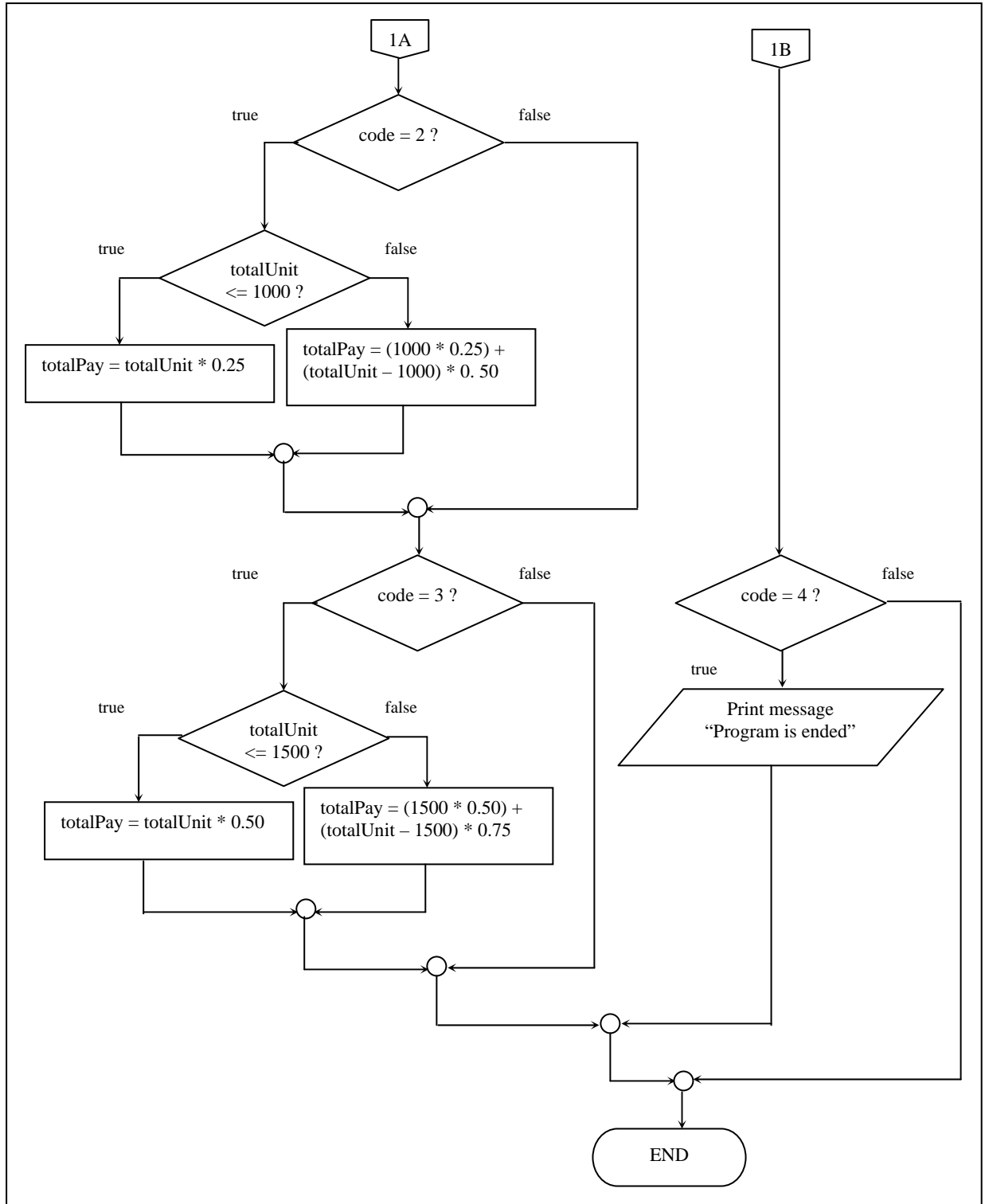


Figure 7.2(b): Flow Chart Part II

LAB 8: LOOP STATEMENTS

OBJECTIVES FOR THE STUDENT

1. To identify `counter` and event controlled loops.
[*Untuk mengenal pasti counter dan peristiwa gelung kawalan.*]
2. To work with the `for` loop.
[*Untuk menggunakan gelung for.*]
3. To work with the `while` loop.
[*Untuk menggunakan gelung while.*]
4. To work with the `do-while` loop.
[*Untuk menggunakan gelung do-while.*]

ASSUMPTIONS

1. Students have the knowledge on basic C programming language.
[*Pelajar mempunyai pengetahuan mengenai asas bahasa pengaturcaraan C.*]
2. Students have the knowledge on input and output statements.
[*Pelajar mempunyai pengetahuan dalam pernyataan input dan output.*]
3. Students have the knowledge on increment and decrement operators.
[*Pelajar mempunyai pengetahuan mengenai operator penambahan dan pengurangan.*]
4. Students have the knowledge on Boolean expressions.
[*Pelajar mempunyai pengetahuan mengenai ungkapan Boolean.*]

LAB EXERCISES

EXERCISE 1:

[LATIHAN 1]

1. Type in the following program (Program 8.1):
[Taip aturcara yang berikut (Program 8.1)]:

```
1 //Program 8.1
2 #include <stdio.h>
3 #include <conio.h>
4 int main()
5 {
6     double price1, price2, price3, price4, price5,
7     price6, price7, totalPrice=0;
8     printf("Program To Calculate The Average and Total
9     Price Of The 7 Items \n\n ");
10    printf("Enter price of item:  ");
11    scanf("%lf", &price1);
12    totalPrice += price1;
13    printf("Enter price of item:  ");
14    scanf("%lf", &price2);
15    totalPrice += price2;
16    printf("Enter price of item:  ");
17    scanf("%lf", &price3);
18    totalPrice += price3;
19    printf("Enter price of item:  ");
20    scanf("%lf", &price4);
21    totalPrice += price4;
22    printf("Enter price of item:  ");
23    scanf("%lf", &price5);
24    totalPrice += price5;
25    printf("Enter price of item:  ");
26    scanf("%lf", &price6);
27    totalPrice += price6;
28    printf("Enter price of item:  ");
29    scanf("%lf", &price7);
30    totalPrice += price7;
31
32    printf("\n\nThe Average Price Of The Items Is  %f",
33    totalPrice/7);
34    printf("\nThe Total Price Of The Items Is      %f",
35    totalPrice);
36    printf("\n\nEnd Of Program");
37    getch();
38    return 0;
39 }
```

2. Study Program 8.1 carefully, then compile and execute it.
[Kaji Program 8.1 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What does the program do?
[Apakah yang dilakukan oleh aturcara tersebut?]
 - ii. Why do you need 7 variables to hold the prices of the items?
[Mengapa 7 pembolehubah diperlukan untuk menyimpan harga barangan?]
 - iii. Why is the variable `totalPrice` initialized to 0?
[Mengapa pembolehubah `totalPrice` diberi nilai awal 0?]
 - iv. Identify the set of statements that are repeated many times in the program.
[Kenalpasti set pernyataan yang diulang beberapa kali dalam aturcara ini.]
 - v. How many times were the set of statements repeated?
[Berapa kalikah set pernyataan tersebut diulang?]

EXERCISE 2:

[LATIHAN 2]

1. Type the following program (Program 8.2):
[Taipkan aturcara yang berikut (Program 8.2) :]

```
1 //Program 8.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     double price, totalPrice=0;
8     int Item;
9     printf("Program To Calculate The Average and Total
10 Price Of The 7 Items \n\n ");
11     for (Item=0; Item<7; Item++)
12     {
13         printf("Enter price of item:  ");
14         scanf("%lf", &price);
15         totalPrice += price;
16
17     }
18     printf("\n\nThe Average Price Of The Items
19 Is %f", totalPrice/7);
20     printf("\n\nThe Total Price Of The Items
21 Is %f", totalPrice);
22     printf("\n\nEnd Of Program");
23     getch();
```

21	<pre>return 0; }</pre>
----	------------------------

2. Study Program 8.2 carefully, then compile and execute it.
[Kaji Program 8.2 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What does the program do? In what way is this program different from Program 8.1.
[Apakah yang dilakukan oleh aturcara tersebut? Apakah perbezaan antara Program 8.1. dan Program 8.2.]
 - ii. Identify the body of the loop, and state how many times it is repeated.
[Kenalpasti badan gelung, dan nyatakan berapa kalikah ia diulang.]
 - iii. Why do we now need only one variable to hold the prices of the item.
[Mengapa kali ini kita hanya perlu satu pembolehubah untuk menyimpan harga barangan.]
 - iv. Change the statement `for (Item=0; Item<7; Item++)` in line 8 to the following; and state whether the output of the program remains the same, or if the output changes, what are the changes:
[Tukar pernyataan `for (Item=0; Item<7; Item++)` pada baris 8 kepada pernyataan berikut; dan nyatakan sama ada hasil aturcara tetap sama atau berubah. Jika berubah, nyatakan apakah perubahan tersebut:]
 - a. `for (Item=0; Item <=7: Item ++)`
 - b. `for (Item =0; Item <35: Item +=5)`
 - c. `for (Item =10; Item I<17: Item ++)`
 - d. `for (Item =7; Item >0; Item --)`
3. Modify the program such that it will calculate and print the average and the total prices of 20 items.
[Ubahsuai Program 8.2 supaya ia mengira dan mencetak purata dan jumlah harga untuk 20 barangan.]

EXERCISE 3:**[LATIHAN 3]**

1. Download Program 8.3:
[*Muat turun aturcara Program 8.3*]

```
1 //Program 8.3
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     double price, totalPrice=0;
8     int Item, itemCnt;
9     printf("Program To Calculate The Average and
10            Total Price Of Items \n\n ");
11     printf("Enter the number of item :  ");
12     scanf("%d", &itemCnt);
13
14     for (Item=0; Item<itemCnt; Item++)
15     {
16         printf("Enter price of item:  ");
17         scanf("%lf", &price);
18         totalPrice += price;
19
20     }
21
22     printf("\n\nThe Average Price Of The Items
23            Is  %f", totalPrice/ 7);
24     printf("\nThe Total Price Of The Items
25            Is  %f", totalPrice);
26     printf("\n\nEnd Of Program");
27     getch();
28     return 0;
29 }
```

2. Study Program 8.3 carefully, then compile and execute it.
[*Kaji Program 8.3 dengan teliti, kemudian kompil dan laksana aturcara tersebut.*]
 - i. What does the program do? In what way is this program different from Program 8.2
[*Apakah yang dilakukan oleh aturcara tersebut? Apakah perbezaan antara Program 8.2. dan Program 8.3*]
 - ii. State how many times the loop is repeated.
[*Nyatakan berapa kalikah gelung diulang.*]
 - iii. What is the role of variable itemCnt in this program?
[*Apakah peranan pembolehubah itemCnt dalam aturcara ini?*]
 - iv. Program 8.3 contains 1 logic error that makes the program unreliable. Identify and fix the error.

[Program 8.3 mengandung 1 kesalahan logik yang menjadikan aturcara tersebut tidak boleh diharap. Kenalpasti dan betulkan kesalahan tersebut.]

EXERCISE 4:

[LATIHAN 4]

1. Download Program 8.4:
[Muat turun aturcara Program 8.4 :

```

1 //Program 8.4
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     double price, totalPrice=0;
8     int itemCnt=0;
9     // read price of 1 item before entering loop
10    printf("Enter price of item:  ");
11    scanf("%lf", &price);
12
13    while (price>0)
14    {
15        totalPrice += price;
16        itemCnt++;
17        printf("Enter price of item:  ");
18        scanf("%lf", &price);
19    }
20
21    printf("\n\nThe Average Price Of The Items Is  %f",
22           totalPrice/itemCnt);
23    printf("\nThe Total Price Of The Items
24           Is          %f", totalPrice);
25    printf("\n\nEnd Of Program");
26    getch();
27    return 0;
28 }

```

2. Study Program 8.4 carefully, then compile and execute it.
[Kaji Program 8.4 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What type of control is used by the `while` loop in Program 8.4?
[Apakah jenis kawalan yang diguna oleh gelung `while` dalam Program 8.4?]
 - ii. Does the user know how to tell the program that he has no more item to be read? Modify the program to make it more user-friendly.

[Apakah pengguna tahu bagaimana untuk memberitahu aturcara bahawa dia tiada lagi barangan untuk dibaca? Ubahsuai aturcara supaya ia lebih ramah-pengguna.]

- iii. Why do we need to read the price of 1 item (line 11) before the while loop?
[Mengapa kita perlu membaca harga satu barangan (baris 11) sebelum gelung while?]
- iv. Why is variable itemCnt initialized to 0?
[Mengapa pembolehubah itemCnt perlu diberi nilai awal 0?]

EXERCISE 5:

[LATIHAN 5]

1. Download the following program (Program 8.5):
[Muat turun aturcara berikut (Program 8.5):]

```
1 //Program 8.5
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     double price, totalPrice=0, averagePrice;
8     int proceed=1,itemCnt=0;
9     printf("Program To Calculate The Average and Total
10           Price Of Items \n\n");
11
12     while (proceed==1)
13     {
14         printf("Enter price of item:  ");
15         scanf("%lf", &price);
16         totalPrice += price;
17         itemCnt++;
18         printf("Anymore item? Enter 1 for yes OR 0
19               for no:  ");
20         scanf("%d", &proceed);
21     }
22
23     printf("\n\nYou have %d Items ", itemCnt);
24     printf("\n\nThe Average Price Of The Items Is %f",
25           totalPrice/itemCnt);
26     printf("\n\nThe Total Price Of The Items Is %f",
27           totalPrice);
28     printf("\n\nEnd Of Program");
29     getch();
30     return 0;
31 }
```

2. Study Program 8.5 carefully, then compile and execute it.
[Kaji Program 8.5 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What type of control is used by the WHILE loop in Program 8.5?
[Apakah jenis kawalan yang diguna oleh gelung WHILE dalam Program 8.5?]
 - ii. What is the purpose of variable `proceed` in this program, and why it is initialized to 1?
[Apakah peranan pembolehubah `proceed` dalam aturcara ini, dan mengapakah ia diberi nilai awal 1?]
 - iii. What value of variable `proceed` will exit the loop?
[Apakah nilai pembolehubah `proceed` yang akan menamatkan pelaksanaan gelung ?]
 - iv. Is this program user friendly? Give reasons to your answer.
[Adakah aturcara ini ramah pengguna? Beri sebab kepada jawapan anda.]
 - v. Modify Program 8.5 to implement `proceed` as a Boolean flag.
[Ubahsuai Program 8.5 supaya `proceed` dilaksana sebagai Boolean flag.]

EXERCISE 6:**[LATIHAN 6]**

1. Download the following program (Program 8.6):
[Muat turun aturcara berikut (Program 8.6):]

```

1 //Program 8.6
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     double price, totalPrice=0, averagePrice;
8     int N,itemCnt=0;
9     printf("Program To Calculate The Average and
10            Total Price Of N Items \n\n");
11     printf("How many items do you have?");
12     scanf("%d", &N);
13
14     do
15     {
16         printf("Enter price of item:  ");
17         scanf("%lf", &price);
18         totalPrice += price;
19         itemCnt++;
20     }while (itemCnt<N);

```

```
20
21     printf("\n\nYou have %d Items ", itemCnt);
22     printf("\n\nThe Average Price Of The Items Is  %f",
23           totalPrice/itemCnt);
24     printf("\nThe Total Price Of The Items
25           Is  %f", totalPrice);
26     printf("\n\nEnd Of Program");
27     getch();
28     return 0;
29 }
```

2. Study Program 8.6 carefully, then compile and execute it.
[Kaji Program 8.6 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
- i. What type of loop is used in Program 8.6, and how does it different from Program 8.5?
[Apakah jenis gelung yang diguna dalam Program 8.6? Bagaimana ia berbeza dengan Program 8.5?]
 - ii. What is the purpose of variable N in this program?
[Apakah peranan pembolehubah N dalam aturcara ini?]
 - iii. Why do you need variable `itemCnt` in this program, and why it is initialized to 0?
[Mengapa pembolehubah `itemCnt` diperlukan dalam aturcara ini, dan mengapa ia perlu diberi nilai awal 0?]
 - iv. Replace `while (itemCnt<N)` with the following, and state it's effect to the program.
[Tukar pernyataan `while (itemCnt<N)` kepada pernyataan berikut, dan nyatakan kesannya ke atas aturcara.]
 - a. `while (--N)`
 - b. `while (N--N)`

EXERCISE 7:**[LATIHAN 7]**

1. Download the following program (Program 8.7):
[Muat turun aturcara berikut (Program 8.7):.]

```
1 //Program 8.7
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int mark;
8     do
9     {
10         printf("\nEnter marks for test 1:");
11         scanf("%d", &mark);
12     } while (mark<0 || mark>100);
13     printf("\nYour mark for test 1 is %d", mark);
14     printf("\n\nEnd Of Program");
15     getch();
16     return 0;
}
```

2. Study Program 8.7 carefully, then compile and execute it
[Kaji Program 8.7 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What does the program do?
[Apakah yang dilakukan oleh aturcara tersebut?]
 - ii. We need ONE value for mark in this program. Why do we need the DO-WHILE loop?
[Kita memerlukan SATU nilai untuk mark dalam aturcara ini.
Mengapa kita perlu gelung DO-WHILE?]
 - iii. What is a valid value for mark?
[Apakah nilai yang sah untuk mark?]
3. Modify Program 8.7 such that the data validation is implemented using a DO-WHILE loop,
[Ubahsuai Program 8.7 supaya pengesahan data dilakukan menggunakan gelung DO-WHILE.]

EXERCISE 8:**[LATIHAN 8]**

1. Download the following program (Program 8.7):
[Muat turun aturcara berikut (Program 8.7).]

```
1 //Program 8.8
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int choice=0;
8     printf("\nWELCOME TO CYBER BREAKFAST\n");
9     printf("\n1:   Nasi Lemak with Kopi O  ");
10    printf("\n2:   Roti Canai with Teh Tarik  ");
11    printf("\n3:   Scramble Egg with Iced Milo  ");
12    printf("\n4:   Mixed Cereals with Strawberry
13           Shake  ");
14    printf("\n5:   No thank you, I'm overweight..!
15           \n\n");
16
17    while (choice<1 || choice>5)
18    {
19        printf("Enter your choice:  ");
20        scanf("%d", &choice);
21    }
22
23    switch (choice )
24    {
25        case 1:    printf("\nThat will be RM 6:50 ");
26                  break;
27        case 2:    printf("\nThat will be RM 4:50 ");
28                  break;
29        case 3:    printf("\nThat will be RM 5:00 ");
30                  break;
31        case 4:    printf("\nThat will be RM 8:50 ");
32                  break;
33        default:   printf("\nYou look just great..! ");
34    }
35    printf("\nPlease take a seat and make yourself
36           comfortable.  ");
37    printf("\n\nEnd Of Program");
38    getch();
39    return 0;
40 }
```

2. Study Program 8.8 carefully, then compile and execute it.
[Kaji Program 8.8 dengan teliti, kemudian kompil dan laksana aturcara tersebut.]
 - i. What does the program do?
[Apakah yang dilakukan oleh aturcara tersebut?]

- ii. How do you exit the loop?
[*Bagaimana untuk keluar dari gelung?*]
 - iii. Why is the variable `choice` initialized to 0 ? What other possible values can be used instead?
[*Mengapa pembolehubah `choice` diberi nilai awal 0? Apakah nilai lain yang boleh digunakan menggantikan 0?*]
 - iv. Experiment with entering (invalid) value for `choice`. How does this effect the execution of the program?
[*Uji dengan memberikan nilai tidak sah untuk `choice`. Apakah kesannya terhadap pelaksanaan aturcara?*]
3. Modify Program 8.8 such that the data validation is implemented using a `do-while` loop.
[*Ubahsuai Program 8.7 supaya pengesahan data dilakukan menggunakan gelung `do-while`.*]

EXERCISE 9:

[LATIHAN 9]

1. Translate the flowchart given in Figure 2 to a complete C program.
[*Tukar carta alir yang diberi dalam Gambar Rajah 2 kepada aturcara C yang lengkap.*]

EXERCISE 10:

[LATIHAN 10]

Analyze the given problems, design their solutions using pseudocode or flowchart and/or structure charts; and implement them using a C program.

[*Analisa masalah yang diberi. Buat rekabentuk penyelesaian menggunakan kod `sudo` atau carta alir dan/atau carta struktur.]*

1. The cost of renting a car is RM100 for the first day, RM 70 per day for the next 3 days and RM 50 per day thereafter. If the input is the number of days the car was rented, compute the cost of renting the car. Allow the user to repeat the calculations as many times as needed and enter a zero to stop.
[*Bayaran sewa kereta untuk hari pertama ialah RM100, RM70 sehari untuk 3 hari berikutnya, dan RM50 untuk setiap hari yang seterusnya. Aturcara hendaklah membaca jumlah hari kereta telah disewa, dan kira berapakah jumlah sewa yang perlu dibayar. Aturcara hendaklah membenarkan pengguna untuk mengulangi proses tersebut sebanyak yang ia kehendaki. Pengguna hendaklah memasukkan nilai 0 untuk memberhentikan pengulangan.]*

2. A customer named PetrolUpHere wants you to write a program for customer fueling. It must display a nice greeting, ask the user if they would like unleaded or diesel fuel and how many whole litres they wish to purchase; it then calculates the cost and displays it to the screen on the pump. There is a small keypad on the pump for the customer to input their choices. The price of unleaded is RM1.98 per gallon; diesel is RM1.29. The program must use 4 modules: the greeting, input, calculation, and output. You do not need to perform error checking.

[Pelanggan bernama PetrolUpHere mahu anda menulis satu aturcara untuk pengguna mengisi petrol. Aturcara hendaklah mempamer mesej selamat datang yang menarik, menanya pengguna jika mereka mahu petrol tanpa plumbum atau diesel, dan berapa jumlah yang diperlu dibeli. Kemudian aturcara akan mengira berapa litres yang akan diberikan dan mempamernya di skrin. Pengguna boleh menggunakan papan kekunci yang disediakan di setiap pam untuk memasukkan pilihan mereka. Harga petrol tanpa plumbum ialah RM1.98 setiap liter, dan harga diesel ialah RM1.29 setiap liter. Aturcara hendaklah menggunakan 4 modul: pamer selamat datang, baca input dari pengguna, pengiraan jumlah liter, dan output. Pengesahan data tidak perlu dilakukan.]

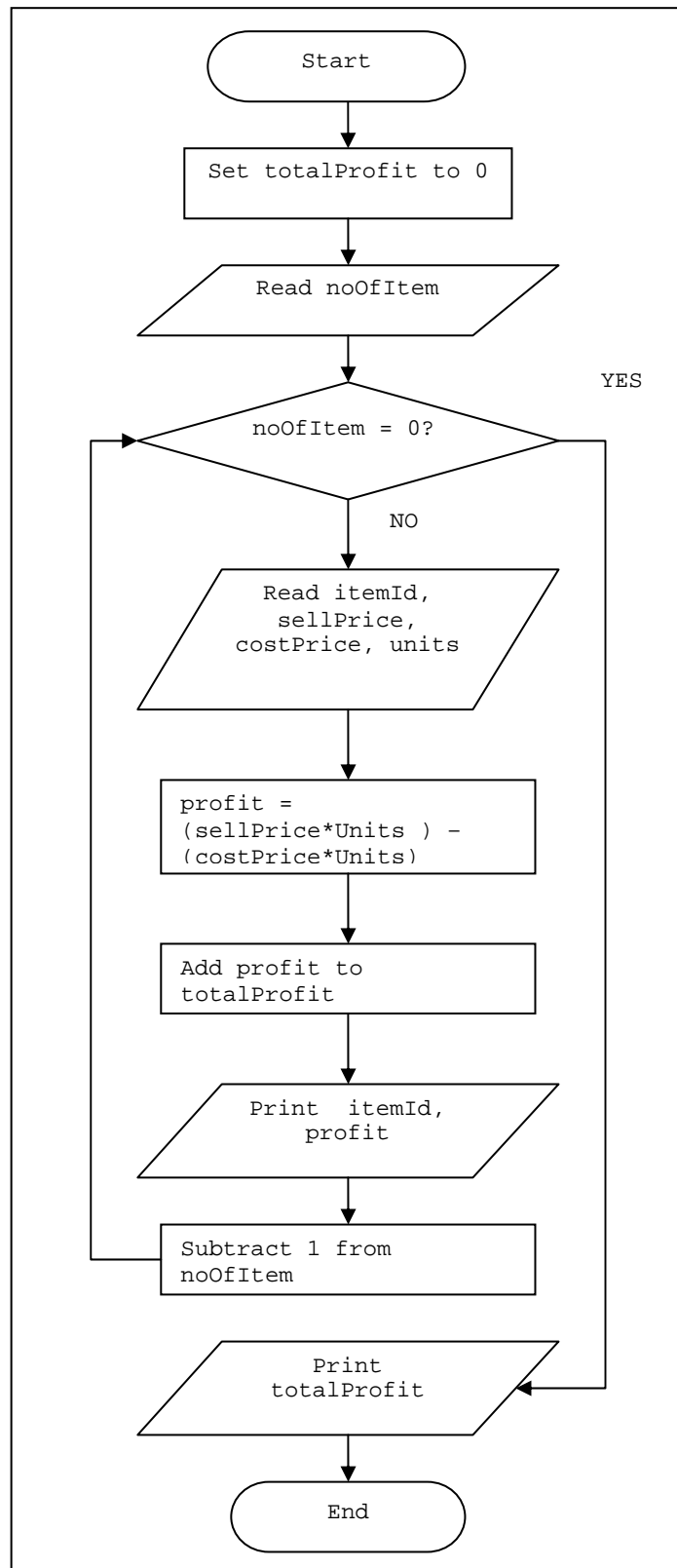


Figure 8.1

LAB 9: NESTED LOOPS

OBJECTIVES FOR STUDENTS

1. Use nested loops.
[*Menggunakan gelung tersarang.*]
2. Independently develop small programs that implement loops.
[*Membina sendiri aturcara ringkas dengan mengaplikasi gelung.*]

ASSUMPTIONS

1. Students must understand basic loops.
[*Pelajar perlu memahami asas gelung.*]
2. Students should have a good understanding of counters.
[*Pelajar perlu memahami konsep counters dengan baik.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Study Program 9.1 carefully, then answer the following questions.
[*Kaji Program 9.1 dengan teliti, kemudian jawab soalan berikut.*]

```
1 // Program 9.1
2 #include <stdio.h>
3 #include <conio.h>
4 int main()
5 {
6     printf ("\n\t Multiplication Tables");
7     printf
8     ("\n_____ \n\n");
9     // print multiplication table for 1
10    for (int j=1; j<=10; j++)
11    {
12        printf ("%4d", j*1);
13    }
14    printf ("\n");
15
16    // print multiplication table for 2
17    for (int j=1; j<=10; j++)
18    {
```

19	printf ("%4d", j*2);
20	}
21	printf ("\n");
22	getch();
23	return 0;
24	}

- i. Identify the first loop in Program 9.1, and explain what does it do.
[Kenalpasti gelung pertama dalam Program 9.1, dan terangkan apakah yang dilakukan oleh gelung tersebut.]
 - ii. Identify the second loop in Program 9.1, and explain what does it do.
[Kenalpasti gelung kedua dalam Program 9.1, dan terangkan apakah yang dilakukan oleh gelung tersebut.]
 - iii. What is the role of variable *j* in both loops?
[Apakah peranan pembolehubah *j* dalam kedua-dua gelung tersebut?]
 - iv. What are the similarities and differences between both loops?
[Apakah persamaan dan perbezaan antara kedua-dua gelung tersebut?]
2. Type, compile and execute Program 9.1
[Taip, kompil dan laksana Program 9.1.]
 - i. What is the output of the program?
[Apakah output aturcara tersebut?]
 - ii. Explain what does the program do?
[Terangkan apakah yang dibuat oleh aturcara tersebut?]
 3. Modify the program such that it will produce the following output using individual loops (3 loops):
[Ubahsuai Program 9.1 supaya ia mengeluarkan output berikut menggunakan gelung berasingan (3 gelung):]
 - i.

4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
 - ii. Is this a good way of writing the program? Give your comments.
[Adakah ini cara yang baik untuk menulis aturcara tersebut? Beri komen anda.]

EXERCISE 2:**[LATIHAN 2]**

1. Type in the following program (Program 9.2):
[*Taip aturcara yang berikut (Program 9.2):*]

```
1 //Program 9.2
2 #include <stdio.h>
3 #include <conio.h>
4 int main()
5 {
6     printf ("\n\t Multiplication Tables");
7     printf
8     ("\n_____ \n\n");
9     // print multiplication table for 1 to 2
10    for (int n=1; n<=2; n++)
11    {
12        for (int j=1; j<=10; j++)
13        {
14            printf ("%4d", j*1);
15        }
16        printf ("\n");
17    }
18
19    printf ("\nEnd of Program");
20    getch();
21    return 0;
22 }
```

2. Study Program 9.2 carefully, then type, compile and execute it.
[*Kaji Program 9.2 dengan teliti, kemudian taip, kompil dan laksana aturcara tersebut.*]
 - i. What is the output of the program?
[*Apakah output aturcara tersebut?*]
 - ii. Program 9.2 is supposed to produce the same output as Program 9.1, but the output is different. Identify the problem and fix it.
[*Program 9.2 sepatutnya mengeluarkan output sama seperti Program 9.1, tetapi output adalah berbeza. Kenalpasti punca perbezaan tersebut dan betulkan.*]
 - iii. Identify the outer loop. How many times will the body of the outer loop execute? What does each outer loop represent? Which exact code determine the number of times the outer loop will execute?
[*Kenalpasti gelung sebelah luar. Berapa kalikah badan gelung luar dilaksanakan? Apakah yang diwakili oleh gelung ini? Kod manakah yang menentukan berapa kali gelung ini akan dilaksana?*]
 - iv. Identify the inner loop. How many times will the body of the inner loop execute? What does each inner loop represent? Which exact code determines the number of times the inner loop will execute?

[Kenalpasti gelung sebelah dalam. Berapa kalikah badan gelung dalam dilaksanakan? Apakah yang diwakili oleh gelung ini? Kod manakah yang menentukan berapa kali gelung ini akan dilaksana?]

3. Modify Program 9.2 so that it will print multiplication tables for 4 to 9.
[Ubahsuai Program 9.2 supaya ia mencetak sifir 4 hingga 9.]

EXERCISE 3:

[LATIHAN 3]

1. Study Program 9.3 carefully, then type, compile and execute it.
[Kaji Program 9.3 dengan teliti, kemudian taip, kompil dan laksana aturcara tersebut.]

```

1 // Program 9.3
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int j;
8     printf ("\n\t Multiplication Tables");
9     printf ("\n _____\n\n");
10    // print multiplication table for 1 to 10
11    for (int n=1; n<=10; n++)
12    {
13        j=1;
14        while (j<=10)
15        {
16            printf ("%4d", j*n);
17            j++;
18        }
19        printf ("\n");
20    }
21
22    printf ("\n\t End of Program");
23    getch();
24    return 0;
25 }
```

- i. Look at variable `j` at line 13. What role does variable `j` play in the `for` loop, and why is it set to 1?
[Lihat pembolehubah `j` pada baris 13. Apakah peranan pembolehubah `j` dalam gelung `for`, dan mengapa ia diberi nilai 1?]
- ii. Look at variable `j` at line 17. What role does variable `j` play in the `while` loop, and why is it incremented with 1?
[Lihat pembolehubah `j` pada baris 17. Apakah peranan pembolehubah `j` dalam gelung `while`, dan mengapa ia ditambah dengan nilai 1?]

2. Modify Program 9.3 to implement the following:
[Ubahsuai Program 9.3 untuk melaksanakan perkara berikut:]
 - i. Use a for loop for the outer loop; and a do-while loop for the inner loop.
[Guna gelung for untuk gelung luar; dan gelung do-while untuk gelung dalam.]
 - ii. Use a while loop for the outer loop; and a for loop for the inner loop.
[Guna gelung while untuk gelung luar; dan gelung for untuk gelung dalam.]

EXERCISE 4:**[LATIHAN 4]**

Analyze the given problems, design their solutions using pseudocode or flowchart and/or structure charts; and implement them using a C program.

[Analisa masalah yang diberi. Buat rekabentuk penyelesaian menggunakan kod sudo atau carta alir dan/atau carta struktur, dan bangukan menggunakan program C.]

1. Write a program to process monthly paycheck for all contract workers of an organization. Each worker will have the following information: Identification number (a 4-digit integer), the worker's name, hourly pay rate, and the number of hours worked during a given month. The normal monthly working hours is 0 to 200. If the gross pay is greater than 2000, a tax of 5% will be deducted. The program output should show the worker's identification number, worker's name, gross pay, tax deduction and net pay for all the workers. Then the output should also show the total gross pay and total tax deductions for the organization. Use an appropriate sentinel value to end the program.
[Tulis satu aturcara untuk mengira gaji bulanan semua pekerja kontrak sebuah organisasi. Setiap pekerja mempunyai maklumat berikut: Nombor pekerja (nombor bulat 4 digit), nama pekerja, kadar bayaran setiap jam, dan jumlah jam bekerja. Jumlah jam bekerja untuk sebulan ialah antara 0 dan 200. Jika jumlah gaji kasar melebihi 2000, potongan cukai sebanyak 5% akan dikenakan. Aturcara hendaklah mengeluarkan output yang terdiri dari nombor pekerja, nama pekerja, gaji kasar, potongan cukai dan gaji bersih. Kemudian jumlah gaji kasar dan jumlah potongan cukai untuk organisasi tersebut hendaklah juga dicetak. Gunakan nilai sentinel yang sesuai untuk menamatkan aturcara.]
2. Write a program that will print neatly the calendar for a year using loops. The required year will be given by the user with values in the range 2000 through 2020, and the program must validate this value. Watch out for leap year. You can use the following format for printing the calendar:
[Tulis satu aturcara untuk mencetak dengan kemas kalendar tahunan menggunakan gelung. Tahun kalendar akan diberi oleh pengguna dengan

nilai antara 2000 hingga 2020. Pastikan aturcara anda mengesahkan nilai adalah dalam julat yang betul. Lihat juga sama ada tahun yang diberi oleh pengguna adalah tahun lompat. Anda boleh gunakan format berikut untuk mencetak kalendar tersebut.]

YEAR 2007

January

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	27
29	30					

Use the following algorithm to assign day of the week for the given date, assumed valid and in the Gregorian calendar. The algorithm assigned value is: 1 for Monday, 2 for Tuesday, ..., 7 for Sunday.

[Guna algoritma berikut untuk mengumpuk satu hari dalam satu minggu jika diberikan tarikh yang sah dan dalam kalender Gregorian. Nilai yang diumpuk ialah: 1 untuk Isnin, 2 untuk Selasa,, 7 untuk Ahad.]

Algorithm 9.1 :

1. Start
2. If (month < 3)
 - Start_If
 - month=month+12
 - decrement year
 - End_If
3. result= day + (13 * month - 27)/5 + year + year /4 - year /100 + year /400
4. result =result+6
5. dayofWeek= (result % 7)+1
6. End

LAB 10: PREDEFINED FUNCTIONS

OBJECTIVES FOR STUDENTS

1. Use predefined functions.
[*Menggunakan fungsi takrifan piawai.*]
2. Convert mathematical formula into C statement using predefined functions.
[*Menukarkan formula matematik kepada bentuk pernyataan C menggunakan fungsi takrifan piawai.*]

ASSUMPTIONS

1. Students have knowledge on arithmetic calculation using C.
[*Pelajar mempunyai pengetahuan di dalam pengiraan aritmetik menggunakan C.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Describe the difference between predefined function and programmer-defined function?
[*Jelaskan perbezaan di antara fungsi takrifan piawai dan fungsi takrifan pengguna?*]
2. Write statement in C to calculate equation bellow using function from math library.
[*Tulis dalam pernyataan C bagi setiap persamaan matematik berikut dengan menggunakan fungsi daripada pustaka math.*]

i. $(x)^{\frac{1}{2}}$

ii. $\cos(x)$

iii. $\sin(x)$

iv. $(x)^7$

v. $\frac{(x^3 + x^7)}{x}$

vi. $\sqrt{|a - b|}$

EXERCISE 2:**[LATIHAN 2]**

1. What is the output of the following C statements? What is the difference between the result in line 5 and 6?

[Apakah output bagi pernyataan C berikut? Apakah perbezaan keputusan pada baris 5 dan 6?]

```

1 double x = 781.2856789;
2 printf ("%5lf\n", x);
3 x =(floor(x * 100 + 0.5)/100);
4 printf ("%5lf\n", x);
5 x = (int(x * 100 + 0.5)/100);
6 printf ("%5lf\n", x);

```

2. Complete Program 10.1 below. Trace the output of each math functions and brief what it does.

[Lengkapkan Program 10.1 di bawah. Tentukan output dan huraikan peranan bagi setiap fungsi math yang digunakan.]

```

1 //Program 10.1
2 #include <stdio.h>
3 #include <conio.h>
4 // 1. Write down an appropriate directive here
5
6 int main()
7 {
8     int a = sqrt(9);
9     printf("\n 1. sqrt(9) is %d ", a);
10
11     int i = pow(5,3);
12     printf("\n 2. pow(5,3) is %d ", i);
13
14     int j = ceil(2.5);
15     printf("\n 3. ceil(2.5) is %d ", j);
16
17     int k = floor(2.5);
18     printf("\n 4. floor(2.5) is %d ", k);
19
20     int x = ceil(2.1);
21     printf("\n 5. ceil(2.1) is %d ", x);
22
23     int y = floor(2.1);
24     printf("\n 6. floor(2.1) is %d ", y);
25     getch();
26     return 0;
27 }

```

EXERCISE 3:**[LATIHAN 3]**

1. Program 10.2 uses several predefined functions from `stdlib` library. Determine what the program tries to show.

[Program 10.2 menggunakan beberapa fungsi yang terdapat di dalam pustaka `stdlib`. Tentukan paparan Program 10.2 tersebut.]

```
1 //Program 10.2
2 #include <stdio.h>
3 #include <conio.h>
4 #include <stdlib.h>
5
6 int main()
7 {
8     int a,b,c,d;
9     printf ("Enter the starting number\n");
10    scanf ("%d", &a);
11    srand(a);
12    b=rand()%RAND_MAX;
13    c=rand()%200;
14    d=rand()%10+200;
15    printf ("The number between 0 and %d is : %d\n",
16    RAND_MAX, b);
17    printf ("The number between 0 and 199
18    is : %d\n", c);
19    printf ("The number between 200 and 210
20    is : %d\n", d);
21    getch();
22    return 0;
23 }
```

EXERCISE 4:**[LATIHAN 4]**

1. Write a program that find the round value of the following number:
[Tulis aturcara bagi mendapatkan nilai bulat seperti nombor berikut:]
 - i. To the nearest tenth.
[Kepada sepuluh terhampir.]
 - ii. To the nearest 1/tenth.
[Kepada 1/sepuluh terhampir.]
 - iii. To the nearest 1/hundredth.
[Kepada 1/seratus terhampir.]

LAB 11: USER-DEFINED FUNCTION

OBJECTIVES FOR STUDENTS

1. Modularize a program system and pass information between systems modules or functions.
[*Memodulkan sistem aturcara dan menghantar maklumat antara modul sistem atau fungsi.*]
2. Differentiate between passing data by value and passing data by reference.
[*Membezakan antara penghantaran data secara nilai dan penghantaran data secara rujukan.*]
3. Return function results through a function's arguments.
[*Memulangkan hasil fungsi menerusi argumen fungsi.*]

ASSUMPTIONS

1. Students should have the knowledge of the predefined functions.
[*Pelajar sepatutnya mempunyai pengetahuan dalam fungsi takrifan piawai.*]
2. Students should have the knowledge of the previous lesson.
[*Pelajar sepatutnya mempunyai pengetahuan daripada pelajaran sebelum ini.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Given the following program in Program 11.1.
[*Diberikan aturcara berikut dalam Program 11.1.*]

```
1 //Program 11.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 void func1(); void func2();
6
7 int main(){
8     int input;
9
10     printf ("Enter a number >");
11     scanf ("%d", &input);
```

```

12
13     if (input < 10) {
14         printf("Faculty of Computer Science & Info.
Systems\n");
15         printf("  Universiti Teknologi Malaysia\n");
16     } else {
17         printf("  Universiti Teknologi Malaysia\n");
18         printf("Faculty of Computer Science & Info.
Systems\n");
19     }
20     getch(); return 0;
21 }

```

- i. Understand the code in program 11.1 and run the program.
[Fahamkan aturcara tersebut dan laksanakan.]
- ii. Change the code so that it follows the design of the structured chart as in Figure 11.1.
[Tukarkan aturcara tersebut supaya ia mematuhi reka bentuk carta berstruktur seperti Rajah 11.1]

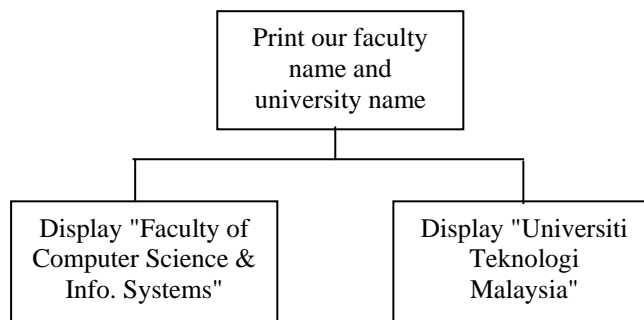


Figure 11.1: Structure chart for a simple print problem

2. Refer to problem in Lab 8, Exercise 8. The program must use 4 functions: the greeting, input, calculation, and output.
[Rujuk kepada masalah pada Lab 8 Latihan 8. Aturcara hendaklah menggunakan 4 fungsi: pamer selamat datang, baca input dari pengguna, pengiraan jumlah liter, dan output.]

EXERCISE 2:

[LATIHAN 2]

1. Given Program 11.2 which consists of a void function. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.
[Diberi Program 11.2 yang mengandungi fungsi tanpa parameter. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```

1 //Program 11.2 - void function example
2 #include <stdio.h>
3 #include <conio.h>
4
5 void printmessage ();
6
7 int main ()
8 {
9     printmessage ();
10    getch();return 0;
11 }
12
13 void printmessage ()
14 {
15     printf ("I'm a function!");
16 }

```

2. Given Program 11.3 which consists of a function that is passing data by value. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.

[Diberi Program 11.3 yang menghantar data secara nilai. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```

1 //Program 11.3
2 #include <stdio.h>
3 #include <conio.h>
4
5 int subtraction (int, int);
6
7 int main ()
8 {
9     int x=5, y=3, z;
10    z = subtraction (7,2);
11    printf ("The first result is %d\n", z);
12    printf ("The second result is %d\n",
13    subtraction(7,2));
14    printf ("The third result is %d\n",
15    subtraction(x,y));
16    z= 4 + subtraction (x,y);
17    printf ("The fourth result is %d\n", z);
18    getch();return 0;
19 }
20
21 int subtraction (int a, int b)
22 {
23     int r;
24     r=a-b;
25     return (r);
26 }

```

3. Given Program 11.4 which consists of a function that is passing data by reference. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.

[Diberi Program 11.4 yang menghantar data secara rujukan. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```
1 //Program 11.4
2 #include <stdio.h>
3 #include <conio.h>
4
5 void duplicate (int &a, int &b, int &c)
6 {
7     a*=2;
8     b*=2;
9     c*=2;
10 }
11
12 int main ()
13 {
14     int x=1, y=3, z=7;
15     duplicate (x, y, z);
16     printf ("x=%d, y=%d, z=%d", x, y, z);
17     getch();return 0;
18 }
```

4. Given Program 11.5 which consists of a function that is returning more than one value. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.

[Diberi Program 11.5 yang mengembalikan lebih dari satu nilai. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```
1 //Program 11.5
2 #include <stdio.h>
3 #include <conio.h>
4
5 void prevnext (int x, int *prev, int *next);
6
7 int main ()
8 {
9     int x=100, y, z;
10    prevnext (x, &y, &z);
11    printf ("Previous=%d, Next=%d", y, z);
12    getch();return 0;
13 }
14 void prevnext (int x, int *prev, int *next)
15 {
16     *prev = x-1;
17     *next = x+1;
18 }
```

5. Given Program 11.6 which consist of fixed value in a function. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.

[Diberi Program 11.6 yang mempunyai nilai tetap dalam fungsi. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```

1 //Program 11.6
2 #include <stdio.h>
3 #include <conio.h>
4
5 int divide (int a, int b=2);
6
7 int main ()
8 {
9     printf ("20/default-value is: %d", divide (12));
10    printf ("\n");
11    printf ("20/4 is: %d", divide (20,4));
12    getch();return 0;
13 }
14
15 int divide (int a, int b)
16 {
17     int r;
18     r=a/b;
19     return (r);
20 }
```

6. Given Program 11.7 which consists of inline function. Compile and identify the output of the program. State the name of function call, function header and function declaration in the program.

[Diberi Program 11.7 yang terdiri daripada fungsi dalam baris. Kompil dan kenal pasti output bagi aturcara tersebut. Nyatakan nama fungsi, kepala fungsi dan pengisytiharan fungsi bagi aturcara tersebut.]

```

1 //Program 11.7
2 inline float cube (const float a) {return a*a*a;}
3
4 int main ( ){
5     float side;
6     printf ("Enter the side length of your cube:  ");
7
8     scanf ("%f", &side);
9     printf ("Volume of cube with side %f is %f\n",
10    side, cube(side));
11    return 0;
12 }
```


EXERCISE 3:

[LATIHAN 3]

1. Download Program 11.9. This program is demonstrates scoping issues with global variables and local variables.

[*Muat turun Program 11.9. Aturcara ini menunjukkan isu skop pembolehubah global dan pembolehubah setempat.*]

```
1 //Program 11.9
2 #include <stdio.h>
3 #include <conio.h>
4
5 void a(void);
6 void b(void);
7
8 int x = 1;
9 int linenum = 0;
10
11 int main( ){
12     int x = 5;
13
14     linenum++;
15     printf ("%dx = %d\n", linenum, x);
16     {
17         int x = 7;
18
19         linenum++;
20         printf ("%dx = %d\n", linenum, x);
21     }
22     linenum++;
23     printf ("%dx = %d\n", linenum, x);
24
25     a(); b();
26
27     linenum++;
28     printf ("%dx = %d\n", linenum, x);
29     getch(); return 0;
30 }
31
32 void a(void){
33     int x = 25;
34
35     linenum++;
36     printf ("%dx = %d\n", linenum, x);
37 }
38
39 void b(void){
40     linenum++;
41     printf ("%dx = %d\n", linenum, x);
42 }
```

- i. Compile and identify the output of the program.
[*Kompil dan kenalpasti output aturcara tersebut.*]
 - ii. Identify which variable x that is being printed for each line of the output.
[*Kenalpasti pembolehubah x manakah yang dicetak untuk setiap baris output.*]
2. Download Program 11.10. This program is demonstrates scoping issues with static variables and local variables.
[*Muat turun Program 11.10. Aturcara ini menunjukkan isu skop pembolehubah statik, pembolehubah setempat dan pembolehubah global.*]

```

1 //Program 11.10
2 #include <stdio.h>
3 #include <conio.h>
4
5 void shownum(); // Function prototype
6
7 int main()
8 {
9     for (int count = 0; count < 5; count++)
10         shownum();
11     getch(); return 0;
12 }
13
14 void shownum()
15 {
16     static int num=5;
17     int num2=10;
18
19     printf ("num is %d\n", num);
20     printf ("num2 is %d\n", num2);
21     num++; num2++;
22 }

```

- i. Compile and identify the output of the program.
[*Kompil dan dapatkan output aturcara tersebut.*]
- ii. What are the difference between static variable and local variable?
[*Apakah perbezaan antara skop pembolehubah statik dan pembolehubah setempat?*]
- iii. What are the differences between static variable and global variable?
[*Apakah perbezaan antara skop pembolehubah statik dan pembolehubah global?*]

EXERCISE 4:**[LATIHAN 4]**

1. Given Program 11.11 that reads three integer numbers and find the maximum number among the three. Complete the program by writing a function named `maximum`, that finds the largest of the three numbers.

[Diberikan Program 11.11 yang membaca tiga nombor integer dan mencari nombor maksimum daripada ketiga-tiga nombor tersebut. Lengkapi aturcara tersebut dengan menulis fungsi untuk mencari nombor terbesar di antara ketiga-tiga nombor.]

```
1 //Program 11.11
2 #include <stdio.h>
3 #include <conio.h>
4
5 int maximum(int,int,int);
6
7 int main()
8 {
9     int a, b, c ;
10
11     printf ("Enter Three Integers: ");
12     scanf ("%d %d %d", &a, &b, &c);
13     printf ("Maximum is: %d\n", maximum(a,b,c));
14
15     getch(); return 0;
16 }
```

2. Given Program 11.12 that reads an integer number and determines whether the number is even or odd. Complete the program by writing a function named `is_even`, that prints "yes" if the number is even, or prints "no", otherwise.

[Diberikan Program 11.12 yang membaca satu nombor integer dan menentukan sama ada nombor itu adalah genap atau ganjil. Lengkapi aturcara tersebut dengan menulis fungsi bernama `is_even`, yang mencetak "yes" sekiranya nombor adalah genap atau mencetak "no" jika sebaliknya.]

```
1 //Program 11.12
2 #include <stdio.h>
3 #include <conio.h>
4
5 void is_even(int a); // Function Prototype
6
7 int main(void)
8 {
9     int a;
10    printf ("Enter an integer:\n");
11    scanf ("%d", &a);
12    printf ("Is %d even?", a);
13    is_even(a);
14    printf ("\n");
15    getch();
16    return 0;
17 }
```

3. Given Program 11.13 that reads the distance in miles and hours, calculates and prints miles per hour. Complete the program by writing the function prototype named `GetData` (at line 5), writing the code that invoke the function `GetData` (at line 13), and writing the body of the function `GetData` that reads miles and hours (at line 23).

[Diberikan Program 11.13 yang membaca jarak dalam unit batu serta jam, mengira dan mencetak batu sejam. Lengkapi aturcara tersebut dengan menulis prototaip fungsi bernama `GetData` (pada baris 5), menulis kod yang memanggil fungsi `GetData` (pada baris 13), dan menulis badan fungsi `GetData` yang membaca batu dan jam (pada baris 23).]

```
1 //Program 11.13
2 #include <stdio.h>
3 #include <conio.h>
4
5 /* Write the function(s) prototype for GetData */
6
7 int main ()
8 {
9     float miles;
10    float hours;
11    float milesPerHour;
12
13 /* Write the code(s) to invoke function GetData */
14
15    milesPerHour = miles / hours;
16    printf ("%10f %10f %10f\n", miles, hours,
17 milesPerHour);
18
19    getch();
20    return 0;
21 }
22
23 /* Write the body of the function(s) GetData that
24 read miles and hours */
```

4. Given Program 11.14 that has a function named `average`, which finds the average of given numbers. Complete the program by writing the function prototype named `average` (at line 5 and 6) that sends 3 parameters, writing the code that invoke the function `average` (at line 15 and 16), and writing the body of the overloaded function `average` that finds the average of three numbers (at line 22).

[Diberikan Program 11.14 yang mempunyai fungsi sarat bernama `average`, yang mendapatkan purata bagi nombor yang diberi. Lengkapi aturcara tersebut dengan menulis prototaip fungsi bernama `average` (pada baris 5 dan 6), menulis kod yang memanggil fungsi `average` (pada baris 15 dan 16) dengan menghantar 3 parameter, dan menulis badan fungsi sarat `average` yang mendapatkan purata bagi tiga nombor (pada baris 22).]

```
1 //Program 11.14
2 #include<stdio.h>
3 #include<conio.h>
4
5 /* Write the function prototype for the function
6 average that sends 3 parameters */
7
8 int main()
9 {
10     int number_A = 5, number_B = 3, number_C = 10;
11
12     printf ("The integer average of %d and %d and %d
13 is\n", number_A, number_B, number_C);
14
15 /* Write the code that invoke the function average
16 with 3 parameters */
17
18     getch();
19     return 0;
20 }
21
22 // write the body of the function average
```

EXERCISE 5:**[LATIHAN 5]**

1. Write a complete program that consists of 4 user-defined functions as listed below, to solve the problem as designed in the flow chart in Figure 11.2. You need to draw a structured chart design before writing the program.

[Tuliskan satu aturcara lengkap yang mengandungi 4 fungsi seperti yang disenaraikan di bawah, untuk menyelesaikan masalah seperti yang direkabentuk pada carta alir di Rajah 11.2. Anda perlu melukis satu reka bentuk carta berstruktur sebelum menulis aturcara tersebut.]

- i. `int getCoin()`
To read the value of a coin input from the user.
[Untuk membaca satu nilai duit siling yang diinput oleh pengguna.]
- ii. `void calculateCoin(int & totalCoin, int coin)`
To calculate the total coins entered by the user and display the total amount.
[Untuk mengira jumlah siling yang dimasukkan oleh pengguna dan mencetak jumlah keseluruhannya.]
- iii. `int calculateBalance(int totalcoin)`
To compute the balance of the coin and display the balance amount.
[Untuk mengira baki duit siling dan mencetak jumlah bakinya.]
- iv. `int main()`
To solve the problem by calling the appropriate function based on the following designed flow chart at Figure 11.2.
[Untuk menyelesaikan masalah dengan memanggil fungsi-fungsi yang bersesuaian berdasarkan reka bentuk carta alir pada Rajah 11.2.]

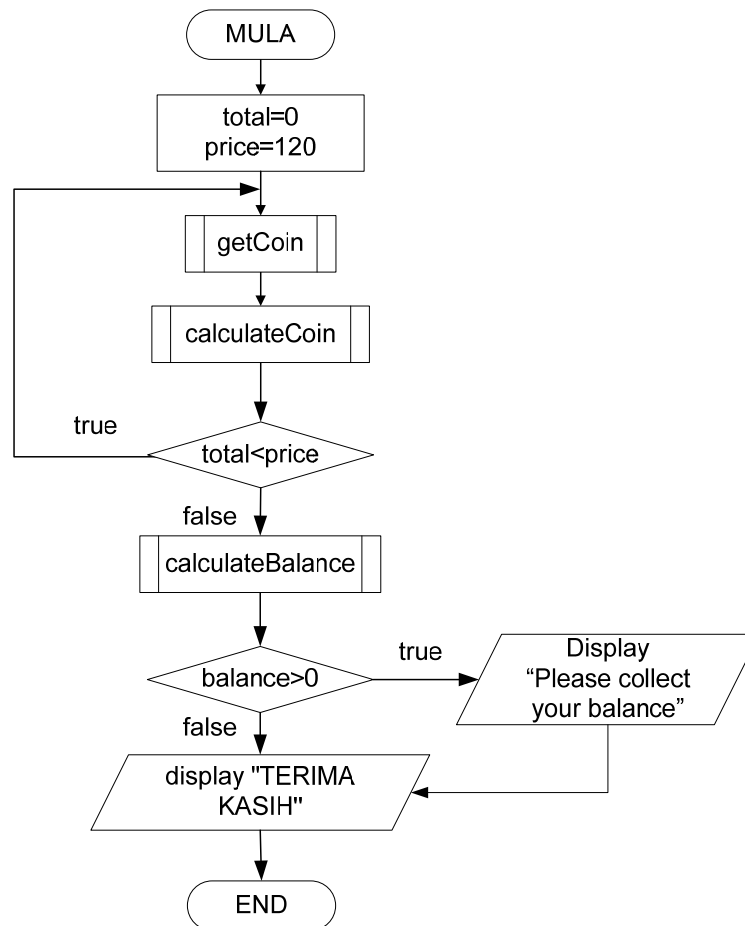


Figure 11.2

EXERCISE 6:
[LATIHAN 6]

1. Write a program that reads a student's name together with his or her assessment marks. The program should then compute the total mark for each student and assign the appropriate grade. In order to compute the total mark, the distribution of marks is tabulated in Table 11.1, while the grade scale is tabulated in Table 11.2. Your program must use the following functions:
[Tuliskan satu aturcara yang membaca nama pelajar dan markah-markah penilaian pelajar tersebut. Aturcara ini perlu mengira jumlah markah bagi setiap pelajar dan memberi gred yang bersesuaian kepadanya. Untuk mengira jumlah markah Jadual 11.1 menunjukkan pembahagian markah, manakala skala gred dijadualkan di Jadual 11.2. Aturcara anda perlu menggunakan fungsi-fungsi berikut.]
 - i. A value-returning function named, `calculateTotal`, that determine the total mark for each student based on the mark distribution in Table 11.1. The function should not display the output of the total mark. That task must be done in the function main.
[Fungsi yang memulangkan satu nilai bernama, `calculateTotal`, yang menentukan jumlah markah untuk setiap pelajar berdasarkan pembahagian markah di Jadual 11.1. Fungsi ini tidak perlu mencetak

output bagi jumlah markah yang dihasilkan. Tugas ini perlu dilakukan dalam fungsi utama.]

- ii. A void function named, `assignGrade`, that determine and return each student's grade. The function should not display the output of the total mark. That task must be done in the function main.

[Fungsi void bernama, assignGrade, yang menentukan dan memulangkan gred untuk setiap pelajar. Fungsi ini tidak perlu mencetak output bagi jumlah markah yang dihasilkan. Tugas ini perlu dilakukan dalam fungsi utama.]

Table 11.1: Assessment Items

Assessments	Percentage (%)
Test 1	20
Assignments	30
Quiz	10
Lab works	10
Final Exam	30
Total	100

Table 11.2: Grade Scale

Assessments	Percentage (%)
A+	90-100
A	80-89
A-	75-79
B+	70-74
B	65-69
B-	60-64
C+	55-59
C	50-54
C-	45-49
D+	40-44
D	35-39
D-	30-34
E	00-29

- iii. Test your program on the following data. Read the data from a text file and send the output to another text file. Do not use global variables. Use the appropriate parameters to pass values in and out of functions.

[Uji aturcara anda dengan data berikut. Baca data dari fail teks dan hantar output ke fail teks yang lain. Jangan menggunakan pembolehubah global. Sila gunakan parameter yang bersesuaian untuk menghantar nilai ke dalam atau keluar dari fungsi.]

Hasyimah	85	90	88	79	90
Rahim 77	80	65	76	60	
Syadhila	79	88	68	79	85
Nurfarhana	81	84	84	82	85
Nurliyana	55	68	69	75	80
Ooi 88	82	88	81	86	
Ping 60	85	45	39	45	
Quah 85	67	50	75	67	
Sham 80	84	89	93	95	
Sien 32	45	86	78	65	
Selan 40	57	60	57	67	
Susan 89	79	88	81	80	
Tan 46	58	69	79	82	

Sample Output:

[*Contoh Output:*]

STUDENT	Test 1	Assignments	Quiz	Lab works	Final Exam	Total Mark	Grade
Hasyimah	85	90	88	79	90	87.7	A
Rahim	77	80	65	76	60	71.5	B+
Syadhila	79	88	68	79	85	82.4	A
Nurfarhana	81	84	84	82	85	83.5	A
Nurliyana	55	68	69	75	80	69.8	B
Ooi	88	82	88	81	86	84.9	A
Ping	60	85	45	39	45	59.4	C+
Quah	85	67	50	75	67	69.7	B
Sham	80	84	89	93	95	87.9	A
Sien	32	45	86	78	65	55.8	C+
Selan	40	57	60	57	67	56.9	C+
Susan	89	79	88	81	80	82.4	A
Tan	46	58	69	79	82	66.0	B

LAB 12: ONE-DIMENSIONAL ARRAY

OBJECTIVES FOR STUDENTS

1. Write and create one-dimensional array.
[Menulis dan membina tatasusunan 1 dimensi.]
2. Write and use one-dimensional array with functions.
[Menulis tatasusunan 1 dimensi bersama fungsi.]

ASSUMPTIONS

1. Students should have knowledge in using loop statement.
[Pelajar sepatutnya mempunyai pengetahuan dalam menggunakan pernyataan gelung.]
2. Students should have knowledge in passing arguments to functions.
[Pelajar sepatutnya mempunyai pengetahuan dalam menghantar parameter kepada fungsi.]

LAB EXERCISES

EXERCISE 1:

[LATIHAN 1]

1. Write Program 12.1 and Program 12.2. Differentiate both programs.
[Tulis Program 12.1 dan Program 12.2. Huraikan perbezaan kedua-dua program.]

```
1 //Program 12.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     float score1, score2, score3, score4;
8     printf ("Enter four floats: \n");
9     scanf ("%f", &score1);
10    scanf ("%f", &score2);
11    scanf ("%f", &score3);
12    scanf ("%f", &score4);
13    printf ("The scores in reverse order are: \n");
14    printf ("%f %f %f %f", score4, score3, score2,
15    score1;
```

```

15   getch();
16   return 0;
17 }

```

```

1 //Program 12.2
2 #include <stdio.h>
3 #include <conio.h>
4 #define SIZE 4
5
6 int main()
7 {
8     float score[SIZE];
9     int i;
10    printf ("Enter %f floats: ", SIZE);
11    for(i = 0; i <= (SIZE - 1); i = i + 1)
12        scanf ("%f", &score[i]);
13    printf ("The scores in reverse order are: \n");
14    for(i = SIZE - 1; i >= 0; i = i - 1){
15        printf ("%f", score[i]);
16        printf ("\n");}
17    getch();
18    return 0;
19 }

```

2. Write and run the following Program 12.3, then answer the following questions.

[Tulis dan laksanakan Program 12.3, kemudian jawab soalan berikutnya.]

```

1 //Program 12.3
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     char a[11] = "Boring";
8     printf ("Index 0 has %c\n", a[0]);
9     printf ("Index 1 has %c\n", a[1]);
10    printf ("Index 2 has %c\n", a[2]);
11    printf ("Index 3 has %c\n", a[3]);
12    printf ("Index 4 has %c\n", a[4]);
13    printf ("Index 5 has %c\n", a[5]);
14    printf ("Index 6 has %c\n", a[6]);
15    printf ("Index 7 has %c\n", a[7]);
16    printf ("Numerically, the a[6] is %c ",a[6]);
17    getch();
18    return 0;
19 }

```

- i. How many indexes do array named, **a**, have?
[Berapakah bilangan indeks yang terdapat pada tatasusunan bernama a?]

- ii. How many characters (including the null character at the end of the array), do this array hold?
[Berapakah bilangan bilangan aksara (termasuk aksara null yang terdapat di akhir tatasusunan) yang dipegang oleh tatasusunan?]
 - iii. The number of an index is also called an index. What is the lowest index?
[Bilangan indeks turut dikenali sebagai indeks. Apakah nilai bagi indeks yang terendah?]
 - iv. What is the highest index for this array?
[Apakah indeks yang tertinggi bagi tatasusunan ini?]
 - v. What is the character value that is stored in the index number 6?
[Apakah nilai aksara yang terdapat pada indeks ke-6?]
3. Run the following Program 12.4 with the first **printf** commented out. Then answer the following questions.
[Laksanakan Program 12.4 berikut dengan pernyataan pertama **printf** dikomenkan. Seterusnya jawab soalan-soalan berikut.]

```

1 //Program 12.4
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     char a[16] = "Programming C";
8     int i;
9     for(i = 0; i <= 9; i = i + 1)
10    {
11        a[i] = a[9 - i];
12        // printf ("i = %d%s\n", i, a);
13    }
14    printf ("%s", a);
15    getch();
16    return 0;
17 }
```

- i. When **i** was 0, which index of **a** was changed? To which value?
[Apabila **i** ialah 0, indeks tatasusunan **a** manakah yang berubah? Perubahan kepada nilai apa?]
- ii. When **i** was 2, which index of **a** was changed? To which value?
[Apabila **i** ialah 2, indeks tatasusunan **a** manakah yang berubah? Perubahan kepada nilai apa?]
- iii. When **i** was 9, which index of **a** was changed? To which value?

[Apabila *i* ialah 9, indeks tatasusunan *a* manakah yang berubah? Perubahan kepada nilai apa?]

- iv. After removing the comment at line 12, show the output of the program.
[Selepas menghapuskan symbol komen pada baris 12, tunjukkan output bagi aturcara.]

4. Run the following Program 12.5 and answer the following questions.
[Laksanakan Program 12.5 berikut dan jawab soalan-soalan berikut.]

```

1 //Program 12.5
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int i, x[6], y[6] = {3, 8, 2, 9, 4, 1};
8     for(i = 0; i <= 5; i = i + 1)
9         x[i] = y[i];
10    for(i = 0; i <= 5; i = i + 1)
11        printf ("%d", x[i]);
12    getch();
13    return 0;
14 }
```

- i. What are the names of the arrays?
[Apakah nama-nama tatasusunan tersebut?]
- ii. Which array is initialized during its declaration?
[Tatasusunan manakah yang dinilaiawalkan semasa penakrifan?]
- iii. Which array is assigned values in a loop?
[Tatasusunan manakah yang mengumpukkan nilai di dalam gelung?]
- iv. Determine the output if statement in line 9 is changed with **`x[i] = y[i] + 1.`**
[Tentukan output jika pernyataan pada baris 9 ditukarkan kepada **`x[i] = y[i] + 1.`**]
- v. Determine the output if statement in line 9 is changed with **`x[i] = y[5 - i].`**
[Tentukan output jika pernyataan pada baris 9 ditukarkan kepada **`x[i] = y[5 - i].`**]

5. Run Program 12.6 below with the statement at line 9 being replaced by suggested statement below. Show the output of this program.
 [Laksanakan Program 12.6 dengan pernyataan pada baris 9 digantikan dengan pernyataan-pernyataan yang dicadangkan berikut. Tunjukkan output bagi aturcara ini.]

```

1 //Program 12.6
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int i, a[] = {40, 20, 70, 10, 80, 30, 90};
8     for(i = 1; i <= 5; i = i + 1)
9         //_____
10        printf ("\n ");
11    getch();
12    return 0;
13 }
```

- i. `printf ("%d\t", a[i]);`
- ii. `printf ("%d\t", a[i+1]);`
- iii. `printf ("%d\t", a[i-1]);`
- iv. `printf ("%d\t", a[i] + a[i+1]);`
- v. `printf ("%d\t", a[i] + i);`

EXERCISE 2: [LATIHAN 2]

1. The following program segment in Program 12.7 is an implementation of a sorting algorithm that sorts a list of numbers in an array. Understand the program and assume that a call to the function `sort` is as follows.
 [Segmen aturcara pada Program 12.7 berikut merupakan implementasi bagi algoritma isihan yang mengisih satu senarai nombor dalam tatasusunan. Fahamkan aturcara tersebut dan andaikan panggilan kepada fungsi `sort` adalah seperti di bawah.]

```

int a[7]={8,6,5,3,9,2,7};
sort(a,6);
```

```

1 //Program 12.7
2 void swap_values(int &v1, int &v2)
3 {
4     int temp;
5     temp = v1;
6     v1 = v2;
7     v2 = temp;
8 }
9
10 int index_of_smallest(const int a[], int begin_index,
    int end_index)
```

```

11 {
12     int min = a[end_index],
13         index_of_min = end_index;
14
15     for (int index=begin_index; index < end_index;
16         index++)
17         if (a[index] < min )
18         {
19             min = a[index];
20             index_of_min = index;
21         }
22     return index_of_min;
23 }
24
25 void sort(int a[], int last_index)
26 {
27     int index_of_next_smallest;
28
29     for (int index=last_index; index>=0; index--)
30     {
31         index_of_next_smallest = index_of_smallest(a, 0,
32         index);
33         swap_values(a[index], a[index_of_next_smallest]);
34     }

```

- i. Write the contents of the array **a** for each loop when the function **sort** is being executed. (Write your answers in a table as shown below).
 [Tuliskan kandungan tatasusunan **a** bagi setiap ulangan semasa fungsi **sort** dilaksanakan. (Tuliskan jawapan anda menggunakan jadual di bawah).]

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]
8	6	5	3	9	2	7

2. Write C statements to perform each of the following:
 [Tulis pernyataan C untuk melaksanakan setiap satu yang berikut:]
- i. Declare an array of variable **number** to allocate 10 elements.
 [Isytihar satu tatasusun bagi pembolehubah **number** untuk menyediakan 10 elemen.]

- ii. Read 10 data then assign the values into array **number**.
[Membaca 10 data untuk dimasukkan ke dalam tatasusun bagi **number**.]
- iii. Assign value of **number[3]** into **number[4]** and **number[4]** takes value of **number[5]**.
[Mengumpulkan nilai **number[3]** kepada **number[4]** dan **number[4]** mengambil nilai **number[5]**.]

EXERCISE 3:

[LATIHAN 3]

1. Given the formula for converting Fahrenheit (F) to Celcius (C):
[Diberi rumus penukaran dari unit Farenheit (F) kepada unit Selsius (C):]

$$C = 5/9 \times (F - 32)$$

Where **C** is the unit of temperature in Celcius and **F** is the unit of temperature in Farenheit

[Dimana **C** ialah unit suhu di dalam Celcius dan **F** ialah uni suhu di dalam Farenheit.]

Write a complete C program that reads in a list of data **F** from a text file, then calculates the values of **C** using the formula given. The program should use an array to store the values of **F** as example shown in Figure 12.1

[Tuliskan satu aturcara C lengkap yang membaca satu senarai data **F** daripada sebuah fail teks, kemudian mengira nilai-nilai **C** menggunakan formula yang diberi. Aturcara tersebut perlu menggunakan tatasusunan untuk menyimpan nilai-nilai **F** seperti contoh pada Rajah 12.1.]


```

13.29
29.76
14.81
23.78
29.37
..
..
..

```

Figure 12.1: input file of data **F**
[Rajah 12.1: fail input bagi data **F**]

```

Average of the temperature : 32.3
Number of high temperature : 2
Number of medium temperature : 20
Number of low temperature : 8

```

Figure 12.2: Example of output skrin
[Rajah 12.2: example of screen output]

The program then prints a **summary output** onto the screen and the **detail output** into a text file as shown in Figures 12.2 and 12.3. Grades 'H' mean high temperature; 'M' is medium temperature and 'L' is low temperature [Seterusnya, aturcara tadi akan mencetak **output rumusan** ke skrin dan **output terperinci** ke dalam sebuah fail teks seperti yang ditunjukkan masing-masing pada Rajah 12.2 dan 12.3. Gred 'H' mewakili suhu tinggi; 'M' pula mewakili suhu sederhana dan 'L' mewakili suhu rendah.]

C(Celcius)	F (Fahrenheit)	Description
=====	=====	=====
54.94	130.89	H
19.86	67.75	L
93.70	200.67	H
13.77	56.78	L
:	:	:
:	:	:
:	:	:

Figure 12.3: example of output file
[Rajah 12.3: contoh fail output]

Your program must define several functions at least as listed in the Table 12.1. You are also required to apply the concept of parameter passing to these functions.

[Aturcara anda perlu mendefinisikan beberapa fungsi sekurang-kurangnya seperti yang disenaraikan dalam jadual berikut. Anda juga dikehendaki melaksanakan konsep penghantaran parameter terhadap fungsi-fungsi tersebut.]

Table 12.1

Fungsi [<i>Function</i>]	Keterangan [<i>Description</i>]
readFile	<p>This function reads in a list of numbers from a text file and stores them into a one-dimensional array. It receives the following parameters:</p> <p>[Fungsi ini digunakan untuk membaca senarai nombor daripada sebuah fail teks dan menyimpan nombor-nombor tersebut ke dalam sebuah tatasusunan. Ia perlu menerima parameter-parameter berikut:]</p> <ul style="list-style-type: none"> • The name of the text file to be read from [Nama bagi fail teks yang hendak dibaca] • An array to store the list of numbers read [Sebuah tatasusunan untuk menyimpan senarai nombor tersebut] • A variable to store the number of data read [Pembolehubah untuk menyimpan bilangan data yang telah dibaca]
computeC	<p>This function computes the values of C. It receives the following parameters:</p> <p>[Fungsi ini digunakan untuk mengira nilai-nilai C. Ia perlu menerima parameter-parameter berikut:]</p> <ul style="list-style-type: none"> • An array that contains data F [Tatasusunan yang mengandungi senarai data F] • An array to store the calculated values of C [Tatasusunan yang akan menyimpan senarai nilai C yang telah dikira] • The number of data [Bilangan data]
average	<p>This function computes the average of a list of numbers stored in an array.</p> <p>[Fungsi ini digunakan untuk mengira purata bagi senarai nombor dalam sebuah tatasusunan.]</p>
grade	<p>This function determines either temperature (C) is high or</p>

	<p>medium or low. This function will return 'H' if $C \geq 35$, 'M' if $C < 35$ and $C \geq 20$ and 'L' if $C < 20$.</p> <p>[Fungsi ini digunakan untuk menentukan sama ada suhu itu tinggi, sederhana atau rendah dalam pengujian kategori suhu tersebut. Fungsi ini memulangkan nilai 'H' sekiranya $C \geq 35$, 'M' jika $C < 35$ atau $C \geq 20$ dan 'L' jika $C < 20$]</p>
writeFile	<p>This function prints the output file as in Figure 12.3. It receives the following parameters:</p> <p>[Fungsi ini digunakan untuk mencetak fail output seperti dalam Rajah 12.3. Ia perlu menerima parameter-parameter seperti di bawah:]</p> <ul style="list-style-type: none"> • An array that contains data F [Tatasusunan yang mengandungi senarai data F] • An array that contains data C [Tatasusunan yang mengandungi senarai data C] • The number of data [Bilangan data]

For printing summary output onto the screen, you may define another function or you may just put the code into the function **main**.

[Untuk cetakan **output rumusan** ke skrin, anda juga boleh mendefinisikan satu fungsi yang lain atau anda boleh menulis kod-kod yang berkenaan terus ke dalam fungsi **main**.]

LAB 13: TWO-DIMENSIONAL ARRAY

OBJECTIVES FOR STUDENTS

1. Write and create two-dimensional array.
[*Menulis dan membina tatasusunan dua dimensi.*]
2. Write and use two-dimensional array with functions.
[*Menulis tatasusunan dua dimensi bersama fungsi.*]

ASSUMPTIONS

1. Students should have knowledge in using loop statement.
[*Pelajar sepatutnya mempunyai pengetahuan dalam menggunakan pernyataan gelung.*]
2. Students should have knowledge in passing arguments to functions.
[*Pelajar sepatutnya mempunyai pengetahuan dalam menghantar argument kepada fungsi.*]
3. Students should have knowledge in manipulating basic array.
[*Pelajar sepatutnya mempunyai pengetahuan memanipulasi tatasusunan asas.*]

LAB EXERCISES

EXERCISE 1:

[*LATIHAN 1*]

1. Show the output of Program 13.1 and answer the following questions.
[*Paparkan output bagi Program 13.1 dan jawab soalan-soalan berikut.*]

```
1 //Program 13.1
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int i, j;
8     int a[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};
9
10    for(i = 0; i <= 2; i = i + 1)
11    {
12        for(j = 0; j <= 3; j = j + 1)
13            printf ("a[%d][%d] = %d\t", i, j, a[i][j]);
```

```

14     printf ("\n");
15     }
16     getch();
17     return 0;
18     }

```

- i. When **i** is 0 and **j** is 1, the element in which row and which column is printed?
[Apabila **i** ialah 0 dan **j** ialah 1, elemen pada baris dan lajur manakah yang akan dipaparkan?]
 - ii. When **i** is 0 and **j** is 3, the element in which row and which column is printed?
[Apabila **i** ialah 0 dan **j** ialah 3, elemen pada baris dan lajur manakah yang akan dipaparkan?]
 - iii. When **i** is 2 and **j** is 1, the element in which row and which column is printed?
[Apabila **i** ialah 2 dan **j** ialah 1, elemen pada baris dan lajur manakah yang akan dipaparkan?]
2. Write and run the following Program 13.2, then try answering questions that follow.
[Tulis dan laksanakan Program 13.2 serta jawab soalan-soalan berikutnya.]

```

1 //Program 13.2
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int i, j;
8     int a[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};
9
10    for(i = 2; i >= 0; i = i - 1)
11    {
12        for(j = 3; j >= 0; j = j - 1)
13            printf ("a[%d][%d] = %d\t", i, j, a[i][j]);
14        printf ("\n");
15    }
16    getch();
17    return 0;
18 }

```

- i. When **i** is 2 and **j** is 3, the element in which row and which column is printed?
[Apabila **i** ialah 2 dan **j** ialah 3, elemen pada baris dan lajur manakah yang akan dipaparkan?]
- ii. When **i** is 2 and **j** is 2, the element in which row and which column is printed?

[Apabila *i* ialah 2 dan *j* ialah 2, elemen pada baris dan lajur manakah yang akan dipaparkan?]

- iii. When *i* is 2 and *j* is 0, the element in which row and which column is printed?
[Apabila *i* ialah 2 dan *j* ialah 0, elemen pada baris dan lajur manakah yang akan dipaparkan?]
- iv. When *i* is 1 and *j* is 3, the element in which row and which column is printed?
[Apabila *i* ialah 1 dan *j* ialah 3, elemen pada baris dan lajur manakah yang akan dipaparkan?]
- v. When the last element is printed, what is *i* and what is *j*?
[Apabila elemen terakhir dipaparkan, apakah nilai *i* dan *j*?]

3. What is the output of the following code fragment of Program 13.3?
[Apakah output bagi keratan Program 13.3 berikut?]

```
1 //Program 13.3
2 int array[3][3], index1, index2;
3 for(index1=0;index1<3;index1++)
4     for(index2=0;index2<3;index2++)
5         array[index1][index2]=index1 + index2;
6 for(index1=0;index1<3;index1++)
7     {
8         for(index2=0;index2<3;index2++)
9             printf ("%d", array[index1][index2]);
10            printf ("\n");
11    }
```

EXERCISE 2:**[LATIHAN 2]**

1. Run the following program and enter this data: **a, b, c, d, e, f, g, h, i, j, k, l**. Show the output and answer the questions.

[Laksanakan aturcara berikut dan masukkan data seperti berikut: **a, b, c, d, e, f, g, h, i, j, k, l**. Paparkan output dan jawab soalan-soalan berikut.]

```

1 //Program 13.4
2 #include <stdio.h>
3 #include <conio.h>
4
5 int main()
6 {
7     int i, j;
8     char a[3][4];
9     for(i = 0; i <= 2; i = i + 1)
10        for(j = 0; j <= 3; j = j + 1)
11            scanf ("%d", &a[i][j]);
12
13        for(i = 0; i <= 2; i = i + 1)
14        {
15            for(j = 0; j <= 3; j = j + 1)
16                printf ("a[%d][%d] = %d\t", i, j, a[i][j]);
17            printf ("\n");    }
18    getch();
19    return 0;
20 }
21

```

- i. Fill in the box given in Figure 13.1 with all the characters are read properly into this array.

[Lengkapkan kotak kosong yang terdapat pada Rajah 13.1 dengan aksara dibaca mengikut aturan yang betul ke dalam tatasusunan.]

Figure 13.1