

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'
 [Pembolehubah 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.]

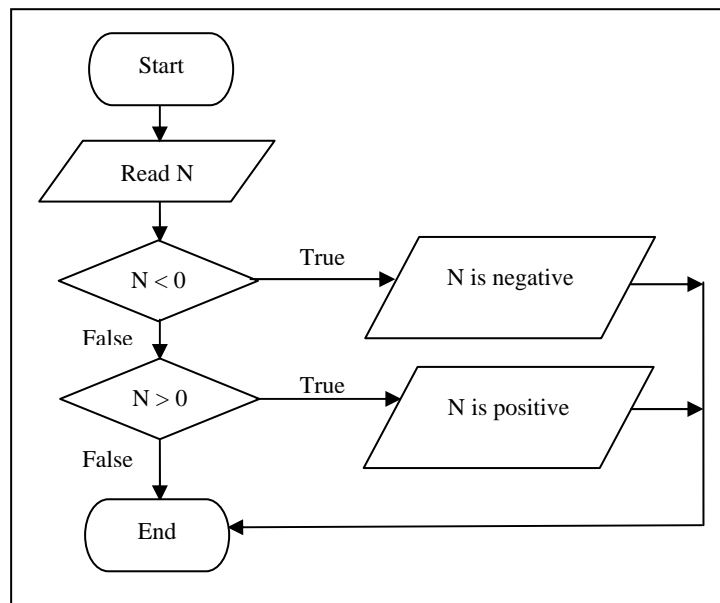


Figure 3.1

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

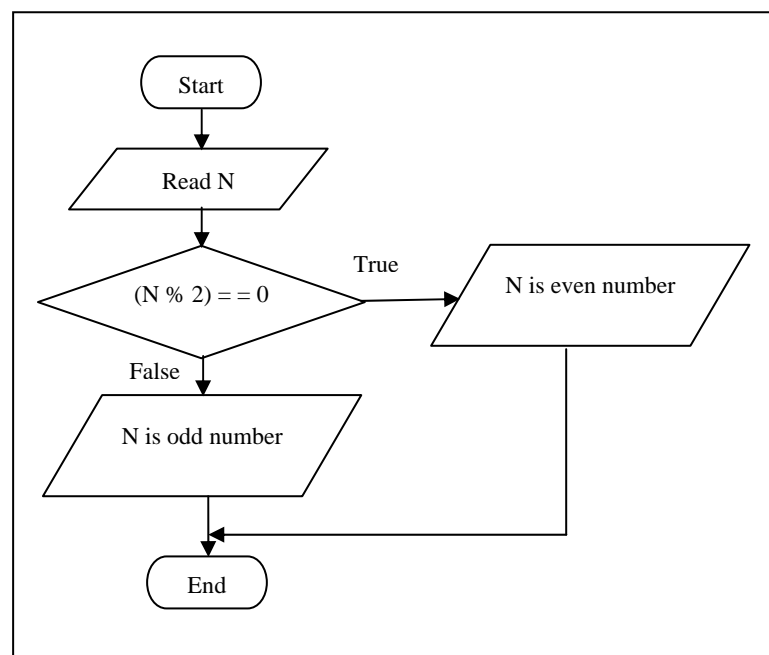


Figure 3.3

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.]

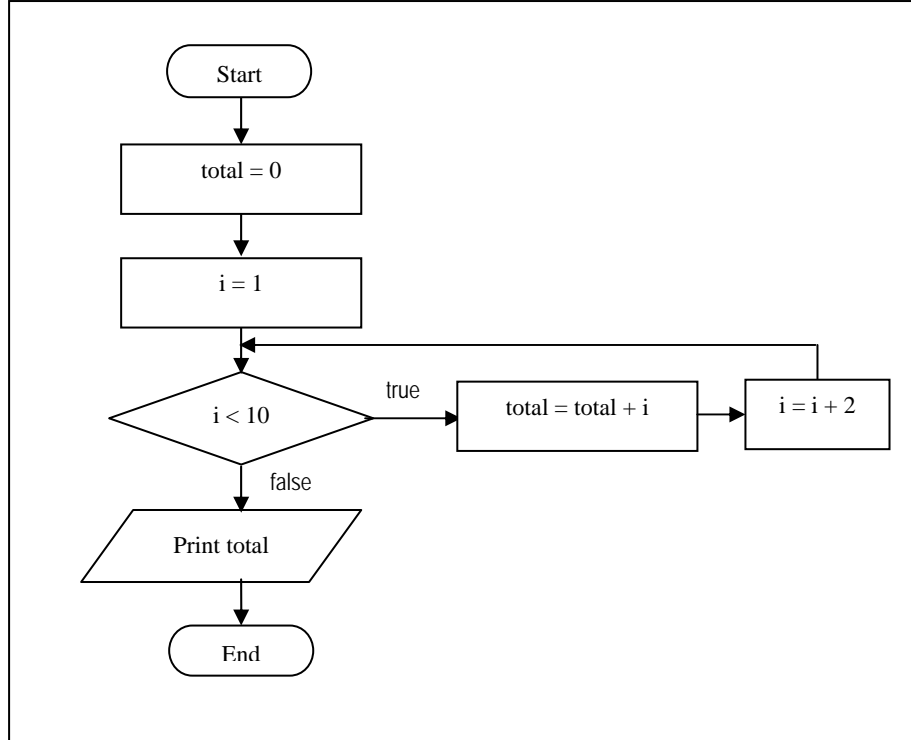


Figure 3.5

- 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.]

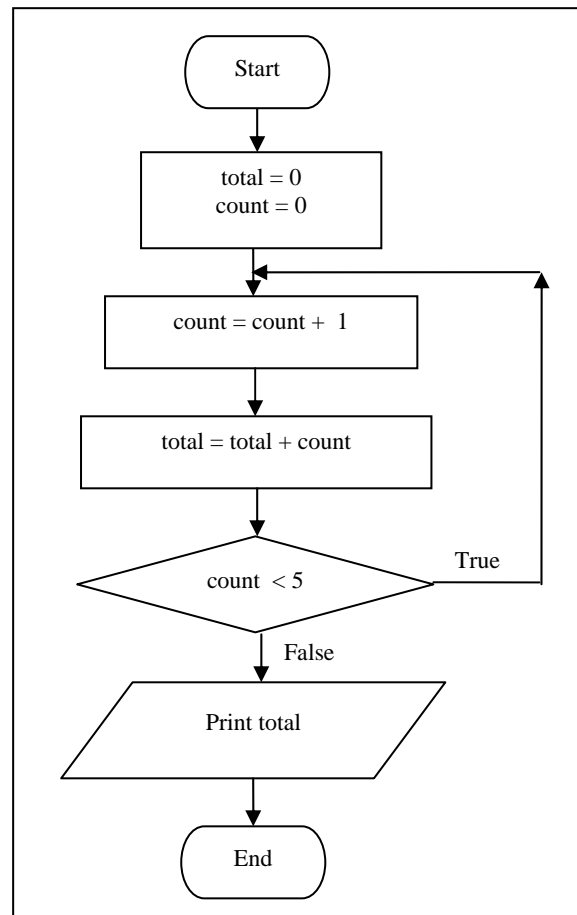


Figure 3.6

- 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