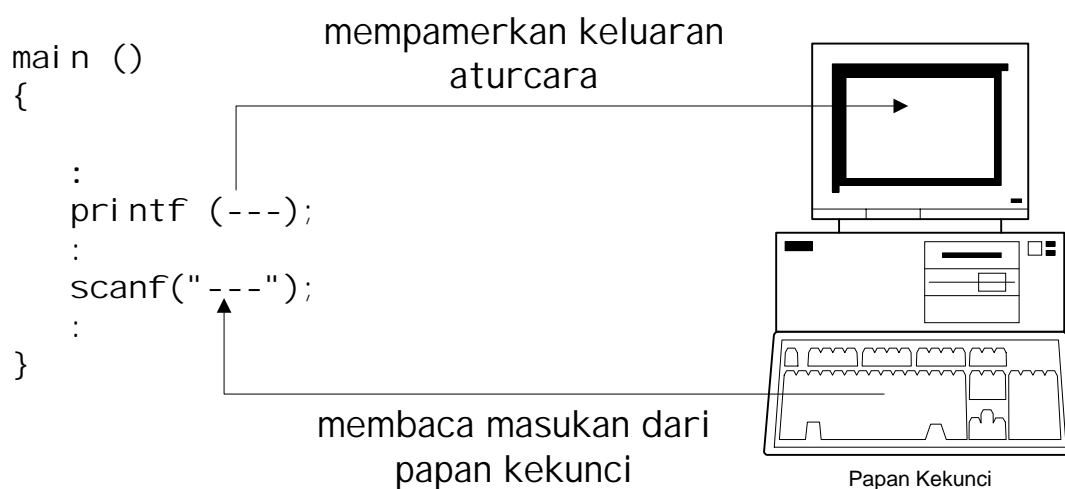


5 FUNGSI INPUT & OUTPUT

5.1 Fungsi Input & Output

- Dlm aturcara yg lepas, kita telah menggunakan fungsi `printf (...)` utk mempamer/mencetak mesej dlm format tertentu.
- Fungsi `scanf (...)` digunakan utk membaca masukan dr papan kekunci dlm format tertentu.



- Fungsi `printf` & `scanf` membolehkan kita berhubung dgn sesuatu aturcara. Fungsi² ini dipanggil fungsi masukan & keluaran (I/O). Fungsi² ini **bukan** sebahagian dari bahasa C. Mereka dipanggil fungsi² perpustakaan yg ditulis/dibekalkan oleh pengeluar pengkompil C.
- Selain fungsi `printf` & `scanf` C juga menyediakan fungsi-fungsi lain untuk operasi input & output. Fungsi² ini boleh dicapai dari perpustakaan C melalui fail kepala piawai `<stdio.h>` & `<conio.h>`.

Contoh:

<code>conio.h</code>	: <code>clrscr</code>	<code>getch</code>	<code>inport</code>
<code>stdio.h</code>	: <code>scanf</code>	<code>getchar</code>	<code>feof</code>

5.2 Fungsi printf & cara Menggunakannya

- Format memanggil fungsi **printf** ialah :
printf(kawalan format, senarai cetakan);

Kawalan format ialah satu rentetan aksara diantara tanda " " yg menerangkan bagaimana pembolehubah² akan dicetak di skrin.

Contohnya :

"luas = %f\n"

"jumlahLembu = %d"

Kawalan format mengandungi 2 bentuk maklumat :

- aksara² yg akan dicetak secara terus
- pencam data @ tentuan penukaran @ tentuan format

Tentuan Format	Keluaran
%d	integer desimal
%c	satu aksara
%s	rentetan aksara
%f	nombor nyata dlm desimal
%e	nombor nyata dlm eksponen
%E	nombor nyata eksponen E dlm huruf besar
%g	guna %f atau %e mana yg lebih pendek
%u	integer desimal tak bertanda
%x	integer hexa tak bertanda
%X	integer hexa tak bertanda (huruf besar)
%o	integer oktal tak bertanda
%%	cetak tanda (%)

Contoh:

```
char a;
printf("%c %d %x %O", a, a, a, a);
```

Keluaran pd skrin:

```
A 65 41 101
```

- Senarai cetakan (jika ada) terdiri dari nama² pembolehubah, pemalar, ungkapan & nama fungsi yg nilainya hendak dicetak. Setiap tentuan format dlm kawalan format mesti mempunyai satu pembolehubah dlm senarai.

5.2 Fungsi printf & cara Menggunakannya (samb....)

○ Contoh 1:

```
bil_buku = 50; harga_buku = 2599.15;
printf("%d buku dengan harga RM%f", bil_buku, harga_buku);
```

Diagram explaining the printf function call:

- tentuan format**: Points to the format string "%d buku dengan harga RM%f".
- dicetak secara terus**: Points to the two arguments, `bil_buku` and `harga_buku`.
- senarai pembolehubah**: Points to the variable names `bil_buku` and `harga_buku` in the arguments.

Keluaran pd skrin:

50 buku dengan harga RM2599.15

○ Contoh 2:

```
/* Contoh aturcara mencetak luas segi empat & bulatan */
/* Penulis : Dayang Norhayati Tarikh: 28.06.2020 */
#include <stdio.h>
#include <conio.h>

main ()
{
    const float pi = 3.1415;
    int panjang, lebar;
    float jejari;

    panjang = 12; lebar = 3;
    jejari = 3.5;

    clrscr();
    printf("Luas bulatan = %f Luas segi 4 = %d\n",
           pi * jejari * jejari, panjang * lebar );
}
```

Keluaran pd skrin:

Luas bulatan = 38.483375 Luas segi 4 = 36

5.2 Fungsi printf & cara Menggunakannya (samb....)

- Format pembolehkan yg dicetak boleh diubahsuai dgn cara memasukkan pengubahsuai di antara % & aksara tentuan format. Contohnya :

%-6d, %5d, %6.2f

Pengubahsuai	Makna
digit	Lebar medan minimum bagi mencetak pembolehkan. Medan yg lebih lebar akan digunakan jika nombor yg dicetak tidak muat dlm medan yg diberi.
.digit	Ketepatan bagi nombor nyata. Mewakili bil digit yg dicetak selepas titik perpuluhan.
-	Pembolehkan akan dicetak bermula dari kiri lebar medannya.
l	Pembolehkan yg akan dicetak adalah jeis integer panjang (long).

Contoh :

Suruhan	Keluaran Skrin
<code>printf("%d\n", 987);</code>	987
<code>printf("%2d\n", 987);</code>	987
<code>printf("%8d\n", 987);</code>	987
<code>printf("%-8d\n", 987);</code>	987
<code>printf("%0.2f\n", 9876.54);</code>	9876.54
<code>printf("%4.2f\n", 9876.54);</code>	9876.54
<code>printf("%3.1f\n", 9876.54);</code>	9876.5
<code>printf("%10.3f\n", 9876.54);</code>	9876.540
<code>printf("%10.3e\n", 9876.54);</code>	9.876e+03

5.3 Fungsi scanf & cara Menggunakannya

- Format memanggil fungsi `scanf` ialah :

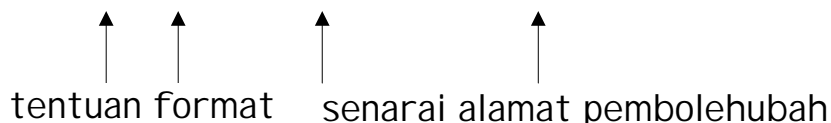
```
scanf(kawalan format, senarai alamat pembolehubah);
```

Kawalan format menyatakan format & jenis data yg akan dibaca dari papan kekunci. Bentuk dan maknanya sama spt format printf.

- Senarai alamat pembolehubah terdiri dari nama² pembolehubah yg dimulai dgn simbol (&). Nilai yg dibaca dari papan kekunci akan diisikan ke dlm pembolehubah tersebut. Setiap tentuan format dlm kawalan format mesti mempunyai satu pembolehubah dlm senarai.

- Contoh :

```
scanf("%d %f", &bil_buku, &harga_buku);
```



- Contoh : Rekakan satu aturcara untuk mencetak luas satu bulatan, jejari bulatan dibaca dari pengguna aturcara.

```

/* Contoh aturcara kira luas bulatan */
#include <stdio.h>

main ()
{
    const float pi = 3.1415;
    float jejari, luas_bulatan;

    printf ("Masukkan jejari bagi bulatan : ");
    scanf("%f", &jejari);
    luas_bulatan = pi * jejari * jejari;

    printf("Luas bulatan = %.2f\n", luas_bulatan );
}

```


5.4 Fungsi² Input & Output Mudah

i) Fungsi `getchar()` & `putchar()`

- dicapai dari perpustakaan piawai `<stdio.h>`
- Fungsi `getchar` digunakan utk membaca satu aksara dari papan kekunci dan menunggu aksara `<ENTER>` dimasukkan. Aksara yang dimasukkan akan dipamir.

Contoh memanggil `getchar`:


```
ch = getchar ();
```

 nilai yg dibaca dari papan kunci akan disimpan dalam `ch`

- Fungsi `putchar` digunakan utk mencetak satu aksara ke skrin.

Contoh memanggil `putchar`:

```
putchar (ch);
```

 parameter pembolehubah @ pemalar aksara yg hendak dicetak

- Contoh 1: baca satu aksara dari papan kekunci & mencetak aksara yg dibaca & aksara selepasnya :

```
char huruf;
```

```
huruf = getchar();
```

```
putchar (huruf);    putchar (huruf+1);
```

Keluaran skrin jika 'E' ditaip :

EF

5.4 Fungsi² Input & Output Mudah (samb....)

○ Contoh 2:

```
char hurufA = 0x41;
putchar (hurufA);   putchar (' b');
putchar (' \n');    putchar (' C');
```

Keluaran skrin :

```
Ab
C
```

ii) Fungsi getch() & putch()

○ dicapai dari perpustakaan piawai <conio.h>

- Fungsi **getch** digunakan utk membaca satu aksara dari papan kekunci tanpa menunggu <ENTER> dimasukkan. Aksara yang dimasukkan tidak akan dipamir.

Contoh memanggil getch:

```
ch = getch ();
```

nilai yg dibaca dari papan kunci akan disimpan dalam ch

- Fungsi **putch** digunakan utk mencetak satu aksara ke skrin.

Contoh memanggil putch:

```
putch (ch);
```

parameter pembolehubah @ pemalar aksara yg hendak dicetak

- Jika contoh dari `getchar` & `putchar` ditukarkan kpd `getch` & `putch` ia akan memberi keluaran yg sama, hanya cara masukan berbeza.

5.4 Fungsi² Input & Output Mudah (samb....)

iii) Fungsi `getc()` & `putc()`

- dicapai dari perpustakaan piawai `<stdio.h>`
- Fungsi `getc` digunakan utk membaca satu aksara dari peranti input piawai. Contoh input piawai fail dan papan kekunci. Operasi bacaan aksara sama spt `getchar()`.

Format memanggil fungsi `getc`:
`getc(jeni s_peranti)`

- Contoh memanggil `getc` & aksara dibaca dari papan kekunci:

```
ch = get (stdin);
```

←
 nilai yg dibaca dari papan kunci akan disimpan dalam ch

- Fungsi `putc` digunakan utk mencetak satu aksara ke skrin.
 Format memanggil fungsi `putc`:

```
putc (operan, jeni s_peranti);
```

- Contoh fungsi `putc` digunakan utk mencetak satu aksara ke skrin.

```
putc(ch, stdout);
```

←
 parameter pembolehkan @ pemalar aksara yg hendak dicetak

- Jika contoh dari `getchar` & `putchar` ditukarkan kpd `getc` & `putc` ia akan dilaksanakan dgn cara yg sama.

5.4 Fungsi² Input & Output Mudah (samb....)

iv) Fungsi `gets()` & `puts()`

- dicapai dari perpustakaan piawai `<stdio.h>`
- Fungsi `gets` digunakan utk membaca rentetan(string) dari keyboard mengakhiri bacaan rentetan jika aksara baris baru (`\n`) @ `<ENTER>` dimasukan. Rentetan akan disimpan & aksara baris baru akan ditukar ke aksara nul (`\0`).
- Fungsi `puts` digunakan utk mencetak rentetan di skrin memasukan aksara baris baru (`\n`).
- Contoh fungsi `gets` & `puts` digunakan dlm aturcara membaca dr papan kekunci & mencetak rentetan di skrin.

```
#include <stdio.h>

main ()
{
    char nama[30];

    printf("\nEnter Your Name please >>");
    gets(nama);    /*baca rentetan*/

    printf("Good day ");
    puts(nama);    /*cetak rentetan*/
    puts("It's your lucky day !!");

    return 0;
}
```

Keluaran pd skrin jika masukan "Dayang Norhayati" dimasukkan:

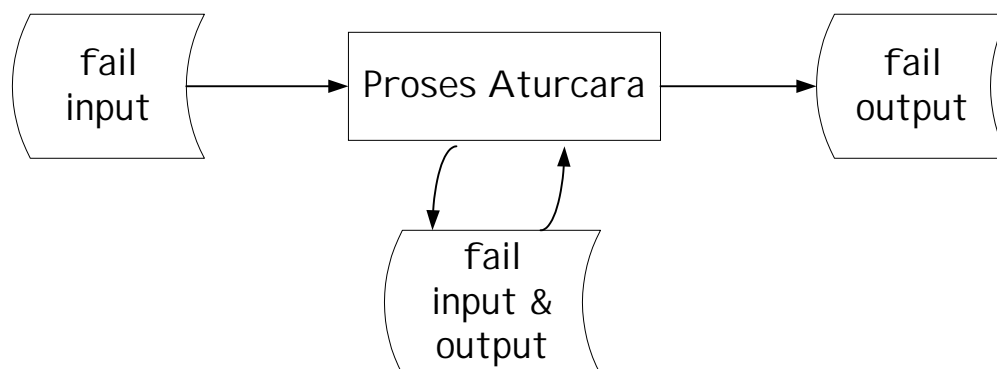
```
Good day Dayang Norhayati
It's your lucky day !!
```

5.5 Pemrosesan Fail

- 2 ragam operasi komputer adalah:
 1. berkelompok (batch)
 2. interaktif (interactive).

Aturcara C yg kita tulis setakat ini adalah ragam interaktif.
Untuk menulis aturcara yang beroperasi secara berkelompok fail perlu digunakan.

- 3 operasi yg dilakukan utk memproses fail:
 1. membina fail
 2. membaca data dr fail
 3. menulis data ke fail
 - wajib
 } salah satu @ kedua2



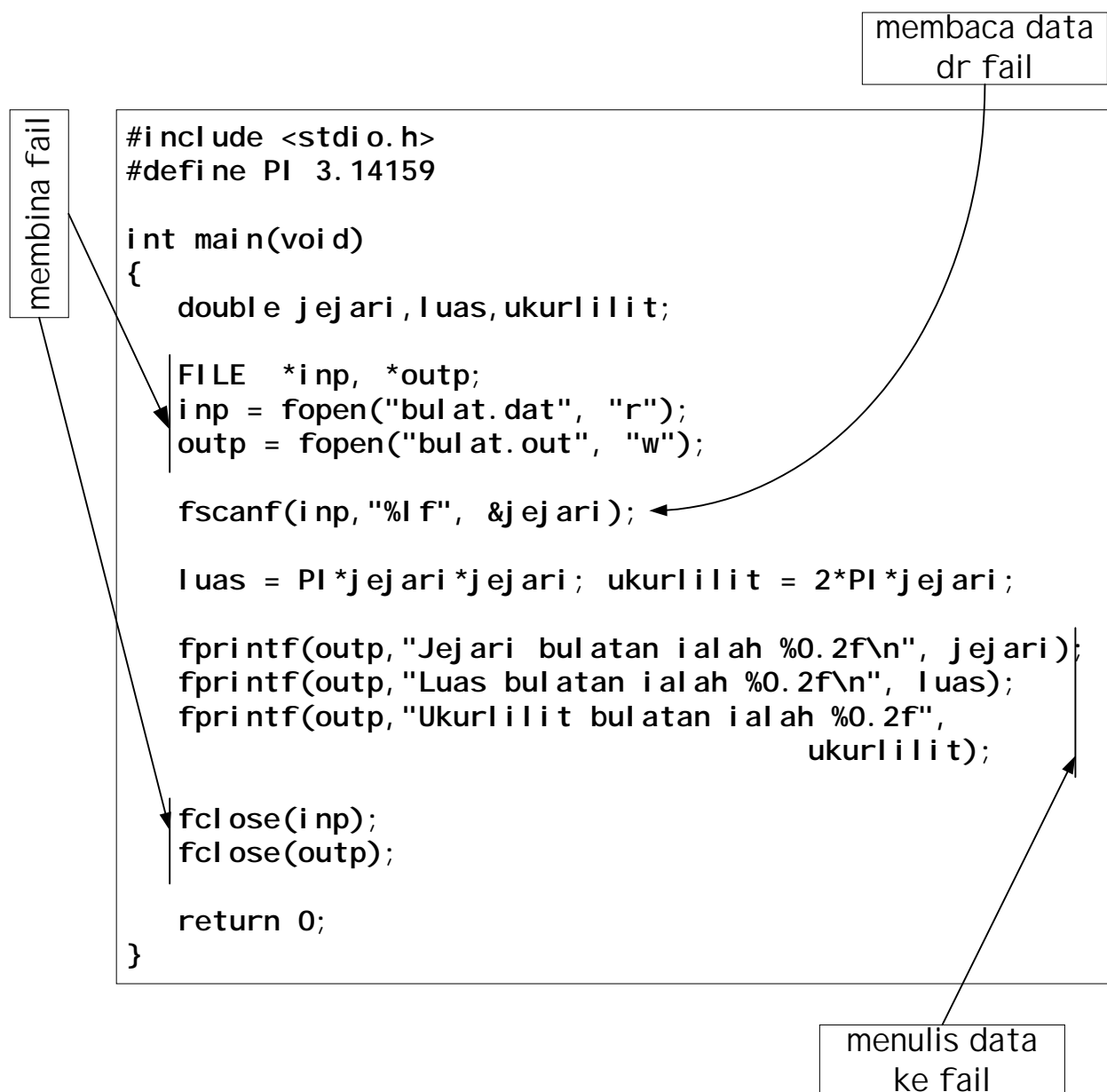
- Merujuk kpd masalah bulat.c, jika kita tukarkan masalah tersebut menggunakan fail:
 - ◆ membaca dr papan kekunci kpd membaca fail bul at. dat
 - ◆ mencetak output ke skrin kpd menulis di fail bul at. out

Katakan isi bul at. dat spt berikut:

5.0

5.5 Pemrosesan Fail (samb....)

- Contoh aturcara menggunakan fail, membaca fail bul at. dat & menulis output di fail bul at. out.



Output dr aturcara akan dikeluarkan ke fail bul at. out & isinya adalah spt berikut:

```

Jejari bulatan ialah 5.00
Luas bulatan ialah 78.54
Ukurlilit bulatan ialah 31.42

```

5.5 Pemprosesan Fail (samb....)

i) Membina fail

- Pengisytiharan fail menggunakan pembolehkan penuding fail, format pengisytiharan:

```
FILE *pembpenudi ngfail ;
```

- Untuk membolehkan fail digunakan, fail tersebut mesti dibuka terlebih dahulu.

Format membuka fail:

```
pembpenudi ngfail = fopen("namafail", "jenis_capaian");
```

↑
pembolehkan penuding fail yg disytiharkan sebelum

↑
nama fail yg sah

↑
jenis mod capaian menentukan samada fail tersebut adalah fail input, output atau keduanya

- jenis mod capaian fail

Mod	Jika fail wujud	Jika fail tidak wujud
"r"	buka fail utk dibaca	Ralat
"w"	buka fail utk ditulis	Bina fail baru
"a"	buka fail utk ditambah di hujung fail	Bina fail baru
"r+"	buka fail utk dibaca & ditulis	Ralat
"w+"	buka fail utk dibaca & ditulis	Bina fail baru
"a+"	buka fail utk dibaca & ditambah di hujung fail	Bina fail baru

- Jika fail tidak digunakan lagi fail perlu ditutup. Format utk menutup fail:

```
fclose(pembpenudi ngfail);
```

5.5 Pemprosesan Fail (samb....)

ii) Membaca dan menulis fail

- Banyak cara boleh dilakukan untuk membaca dan menulis data ke @ dari fail, Ini bergantung kpd cara data hendak diproses:
 - ◆ sama ada aksara demi aksara pada satu masa
 - ◆ satu baris data (satu rentetan aksara) pada satu masa

- Antara fungsi-fungsi yang digunakan untuk membaca & menulis data:
 - ◆ `fscanf()` & `fprintf()` - pelbagai jenis data pada satu masa.
 - ◆ `fgetc()` & `fputc()` - satu aksara demi satu aksara pada satu masa (data berjenis char)
 - ◆ `fgets()` & `fputs()` - satu baris data (rentetan aksara) pada satu masa (data berjenis rentetan)

- Contoh aturcara menggunakan fail, membaca fail `stor12.dat` & menulis output di fail yg mana nama fail dimasukan oleh pengguna melalui papan kekunci.

Isi fail `stor12.dat` adalah spt berikut:

```
MAKMAL KAWALAN
ABC
10 45.20
23 17.39
4 100.14
```

Hasil dari aturcara di ms sebelah, fail output yg terhasil.

```
Laporan Tahunan Stor MAKMAL KAWALAN

Kod Part      Bil      Harga
A             10      45.20
B             23      17.39
C             4       100.14

Nilai stok 1252.53 dan purata harga satu part 54.24
```

5.5 Pemrosesan Fail (samb....)

```

#include <stdio.h>
int main(void)
{
    char    namastor[30], failLaporan[12];
    char    partA, partB, partC;

    int     bilpartA, bilpartB, bilpartC;
    float   harga1partA, harga1partB, harga1partC, nilai stok,
           purataharga;

    FILE    *stor,
           *laporan;

    stor = fopen("stor12.dat", "r");
    printf("Masukkan nama fail laporan > ");
    gets(failLaporan);
    laporan = fopen(failLaporan, "w");

    fgets(namastor, 30, stor);
    partA = fgetc(stor); partB = fgetc(stor); partC =
                                           fgetc(stor);

    fscanf(stor, "%d %f", &bilpartA, &harga1partA);
    fscanf(stor, "%d %f", &bilpartB, &harga1partB);
    fscanf(stor, "%d %f", &bilpartC, &harga1partC);

    nilai stok = (bilpartA*harga1partA) + (bilpartB*harga1partB)
                + (bilpartC*harga1partC);
    purataharga= (harga1partA + harga1partB +harga1partC)/3;

    fprintf(laporan, "Laporan Tahunan Stor ");
    fputs(namastor, laporan);
    fprintf(laporan, "\nKod Part\tBi l\t\tHarga\n");
    fprintf(laporan, "%c\t\t%d\t\t%0.2f\n", partA, bilpartA,
                harga1partA);
    fprintf(laporan, "%c\t\t%d\t\t%0.2f\n", partB, bilpartB,
                harga1partB);
    fprintf(laporan, "%c\t\t%d\t\t%0.2f\n", partC, bilpartC,
                harga1partC);
    fprintf(laporan, "\n\nNilai stok %0.2f dan purata
                harga satu part %0.2f", nilai stok, purataharga);

    fclose(stor);
    fclose(laporan);

    return 0;
}

```

Latihan

1. Apakah keluaran bagi setiap suruhan berikut :

- a) `printf("Panjang seekor semut ialah : %2.2f sm", 2.445e-2);`
- b) `printf("%c %d %u", 66, 0x50, 'C');`
- c) `printf("%-8d\n%6.3f %-6.3f", 4356, 1.52, 1.52);`
- d) `putchar('\ '); putchar('\ '); putchar(66);`

2. Cari kesilapan² dlm aturcara berikut:

```
#include <stdio.h>
main()
{
    int nom1, nom2;
    unsigned int jum1;
    char warna;
    jum1=-1;
    scanf("%d %d", nom1, num2);
    printf("%u", jum1);
    &warna=getchar();
    printf("%d %c %u", warna, warna, warna);
}
```

3. Tulis aturcara untuk melakukan perkara berikut :

- a) Baca 6 nombor dari papan kekunci dan cari purata nombor² tersebut.
- b) Cari isipadu satu bungkah dimana dimensi² diberikan oleh pengguna.
- c) Tukar jarak dlm batu kpd km
- d) Tukar nombor heksadesimal 3 digit kpd nombor desimal (2 cara)