

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 1999/2000

September 1999

CAP101/CMP101 - Prinsip-Prinsip Pengaturcaraan

Masa: [3 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **LAPAN** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
 - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. (a) Nyatakan sama ada kenyataan-kenyataan berikut **BENAR** atau **PALSU**.

- (i) Gelung ulangan bagi kenyataan **while** akan diuji pada bahagian akhir gelung tersebut.
- (ii) Keratan atur cara ini akan memberikan hasil j sebagai 5 selepas keratan atur cara dilaksanakan.

```

j = 3;
k = 9;
if (( 5 * j) >= k)
    j +=2;
j +=3;

```

- (iii) Jumlah parameter sebenar yang dihantar oleh panggilan fungsi mestilah sama dengan jumlah parameter formal dalam takrifan fungsi.
- (iv) Struktur kawalan **while** lebih sesuai digunakan apabila bilangan ulangan telah dikenal pasti terlebih dahulu.
- (v) Keratan atur cara C berikut memberikan nilai yang sama kepada pemboleh ubah x dan pemboleh ubah y jika nilai pemboleh ubah x lebih besar dari nilai pemboleh ubah y.

```

if (x > y) {
    y = x;
    x = y;
}

```

- (vi) Tidak semua struktur ulangan **for** boleh di tulis semula sebagai kod atur cara dengan menggunakan struktur ulangan **while**.
- (vii) Atur cara mestilah disimpan di dalam memori komputer sebelum ianya di laksanakan.
- (viii) Apabila sesuatu atur cara mula dilaksanakan, fungsi-fungsi akan dilaksanakan mengikut urutan kewujudan mereka dalam atur cara.
- (ix) Sesuatu ulangan yang memproses data input sehingga satu nilai khas dimasukkan dipanggil gelung kawalan sentinel.
- (x) Nilai awal data hanya boleh dilakukan semasa pengkompilan.

(20/100)

(b) Apakah yang dimaksudkan oleh ungkapan-ungkapan berikut. Jelaskan dengan ringkas.

- (i) carta alir
- (ii) panggilan fungsi
- (iii) pengkompil
- (iv) atur cara bermodul
- (v) pemalar

(20/100)

...3/-

- (c) Tulis satu pseudokod bagi masalah berikut. Galeri Smart Shop menjual barang seperti di bawah. Apabila pelanggan memesan barang, mereka akan memberikan kod item barang yang dipesan dan kuantiti yang dipesan. Kira jumlah harga barang yang dipesan. Seorang pelanggan boleh memesan lebih daripada satu barang dan mesej yang sesuai akan dikeluarkan jika membaca kod item yang tidak wujud.

Kod Item	Harga (RM)
1	30.00
2	35.00
3	40.00
4	90.00
5	120.00

(30/100)

- (d) Kaji keratan atur cara berikut. Semua pemboleh ubah adalah berjenis integer.

```

k = 0;
m = 0;
for (p = 0; p <= 10; p = p + k) {
    k = k + 1;
    m = m + p;
    printf("%4d%4d%4d\n", p, k, m);
}

```

- (i) Apakah hasil cetakan dari keratan atur cara di atas ?
- (ii) Pemboleh ubah manakah merupakan pemboleh ubah kawalan gelung?
- (iii) Tulis semula keratan atur cara di atas menggunakan gelung do-while.

(30/100)

2. (a) Anda diberikan satu fungsi **kira** yang ditakrifkan sebagaimana berikut:

```

int kira(int q, int b, int c)
{
    int i, j, k, p;
    p = 0;
    for (i = 1; i <= q; ++i) {
        for (j = b; j >= 4; --j)
            p = p + j;
        for (k = 3; k <= c; ++k)
            p += k;
    }
    return (p);
}

```

- (i) Tulis satu prototaip bagi **kira**.
- (ii) Surih (trace) fungsi ini, dan kira nilai yang diumpukkan kepada **q** dalam kenyataan

```
q = kira(3, 3, 4);
```

Tunjuk surihan yang dilakukan.

(30/100)

...4/-

- (b) Andaikan satu syarikat bercadang untuk menambah gaji pekerjaanya berpandukan jadual berikut :

status_pekerja	tahun_khidmat	peratus_naik (%)
sepenuh masa	kurang daripada 5 tahun	4.0
sepenuh masa	5 tahun atau lebih	5.0
sambilan	kurang daripada 5 tahun	2.5
sambilan	5 tahun atau lebih	3.0

Tulis kenyataan 'switch' dan 'if' yang akan mengira gaji baru bagi setiap pekerja dengan diberikan data **status_pekerja**, **tahun_khidmat** dan **gaji**.

Jika nilai **status_pekerja** adalah 'F', maka pekerja tersebut adalah sepenuh masa; jika beliau 'P', beliau adalah sambilan.

Anda hendaklah juga menyemak nilai data **status_pekerja** dan **tahun_khidmat**. Pemboleh ubah **status_pekerja** hanya boleh menerima dua nilai sahaja : 'F' dan 'P', manakala **tahun_khidmat** tidak boleh bernilai negatif dan tidak boleh melebihi 20 tahun. Berikan mesej data tidak sah bagi nilai yang tidak diterima.

(30/100)

- (c) Diberikan masalah berikut:

"Senaraikan nombor genap dan ganjil dari sebilangan (N) nombor-nombor serta dapatkan bilangan setiap kumpulan tersebut"

- (i) Lukis carta berstruktur (perwakilan pepohon reka bentuk) bagi menyelesaikan masalah di atas.
- (ii) Tulis **satu** fungsi bernama **cetak** untuk mencetak senarai nombor genap atau ganjil dan bilangannya pada skrin terminal. Parameter yang dihantar kepada fungsi ini adalah senarai nombor genap atau ganjil (dalam bentuk tatasusunan) dan bilangannya.
- (iii) Beri **dua** kenyataan panggilan kepada fungsi cetak di atas. Satu untuk mencetak senarai nombor genap dan satu lagi untuk senarai nombor ganjil.

(40/100)

3. (a) Apakah output bagi keratan atur cara berikut?

```

(i) void main () {
    int y, x = 1, total = 0;

    while (x <= 5) {
        y = x*x;
        total += y;
        printf("%d\n", total);
        ++x;
    }
}

```

```
(ii) void main () {
        printf("%f \n", kira(35, 120));
        printf("%f \n", kira(50, 100));
    }

float kira (int jam, int jum)
{
    float gaji;

    if (jam <= 40) {
        gaji = 4.5 * jam;
        printf("\n Gaji asas \n");
    }
    else {
        gaji = (jam - 40) * 5.5 + 4.5 * 40;
        printf("\n Gaji lebih masa! \n");
    }
    if (jum > 100)
        gaji += (jum - 100) * 3.0;
    return (gaji);
}

(iii) void main ()
{
    int num, sqnum;
    int *abc;
    int *xyz;
    abc = &num;
    xyz = &num;
    *abc = 5;
    *xyz = 10;
    sqnum = *abc * *xyz;
    printf("num = %d, *abc = %d, *xyz = %d \n", num, *abc, *xyz);
    printf("hasil kiraan = %d \n", sqnum);
    ++(*abc);
    num = sqnum - 10;
    printf("num = %d, *abc = %d, *xyz = %d \n", num, *abc, *xyz);
}

(iv) void main () {
    int x;
    x = 0;
    do {
        printf ("%d\n", x);
        x = x + 2;
    } while (x != 5);
}
```

(50/100)

- (b) Tuliskan satu atur cara yang akan mengira $1 + 2 + 3 + \dots + (n - 1) + n$, **n** adalah nilai **data input** dan **n** mestilah nombor positif. Setelah tamat pengiraan tersebut, banding hasil pengiraan tersebut dengan nilai $(n * (n + 1)) / 2$ dan paparkan mesej yang menyatakan sama ada kedua-dua nilai tersebut sama ataupun tidak. Mesej apakah anda jangka akan dipaparkan ?

(30/100)

- (c) Atur cara berikut memanggil fungsi **cari_maks** yang akan membandingkan dua nombor dan memulangkan nombor yang lebih besar. Nombor terbesar (maks) dipulangkan melalui nama fungsi.

```
int cari_maks(int, int);
main()
{
    int maks, x, y;
    maks = cari_maks(x, y);
}

int cari_maks(int x, int y)
{
    int maks;
    if (x > y)
        maks = x;
    else
        maks = y;
    return maks;
}
```

Tulis semula atur cara di atas supaya nilai **maks** tidak lagi dipulangkan secara nilai, sebaliknya dipulangkan secara rujukan.

(20/100)

4. (a) Takrifkan struktur untuk menyimpan rekod pendaftaran pelajar untuk sesuatu semester yang melibatkan maklumat-maklumat berikut:

- nama pelajar (30 aksara)
- nombor kad pengenalan (12 aksara)
- semester (integer)
- tahun pengajian (5 aksara)
- PNG (titik terapung)
- 5 kursus yang didaftar oleh pelajar berkenaan

Untuk setiap kursus, maklumat berikut akan disimpan:

- kod kursus (6 aksara)
- bilangan unit kursus berkenaan (integer)
- gred (2 aksara)

(20/100)

- (b) Isytiharkan satu pemboleh ubah untuk menyimpan maklumat 20 orang pelajar dengan menggunakan struktur dari (a).

(10/100)

- (c) Tulis satu fungsi bernama **kira_png** untuk mengira PNG untuk data yang tersimpan dalam struktur di atas. Setiap kali fungsi ini dipanggil, fungsi akan mengira PNG untuk seorang pelajar. (Sila rujuk lampiran untuk kaedah pengiraan PNG dan penggunaan gred.)

(30/100)

- (d) Tulis satu fungsi untuk mencetak nama dan PNG setiap pelajar ke dalam satu fail.
(20/100)
- (e) Tulis atur cara utama (fungsi main) yang sesuai untuk memanggil fungsi **input_data**, **kira_png** dan **cetak**. Andaikan sudah wujud satu fungsi untuk menginput data seperti berikut:

```
void input_data (struct pelajar p [ ]) /* kepala fungsi */
```

(Anda tidak perlu menulis fungsi **input_data**.)

(20/100)

Kaedah Pengiraan PNG

Jumlah Nilai Gred (mata penilaian gred * unit kursus) untuk semua kursus dibahagikan dengan jumlah unit bagi semua kursus yang diambil pada satu semester, iaitu:

$$\text{PNG} = \frac{\sum_{i=1}^n G_i U_i}{\sum_{i=1}^n U_i}$$

yang mana

- G_i = Mata penilaian gred kursus ke-i
- U_i = Nilai unit kursus ke-i
- n = Bilangan kursus yang telah diikuti pada semester

Penggunaan Gred

Mata Penilaian Gred	Gred Abjad
4.00	A
3.67	A-
3.33	B+
3.00	B
2.67	B-
2.33	C+
2.00	C
1.67	C-
1.33	D+
1.00	D
0.67	D-
0.00	F

-oooOooo-