



UNIVERSITI TUN HUSSEIN ONN MALAYSIA

**PEPERIKSAAN AKHIR
SEMESTER II
SESI 2010/2011**

NAMA KURSUS : PENGATURCARAAN KOMPUTER
KOD KURSUS : BTI 10202
PROGRAM : SARJANA MUDA KEJURUTERAAN
MEKANIKAL DAN PEMBUATAN
TARIKH PEPERIKSAAN : APRIL / MEI 2011
JANGKA MASA : 2 JAM
ARAHAN : JAWAB **SEMUA** SOALAN DI
BAHAGIAN A DAN **SATU (1)**
SOALAN BAHAGIAN B

*ANSWER ALL QUESTIONS IN PART
A AND ANSWER **ONE (1)** QUESTION
IN PART B*

KERTAS SOALAN INI MENGANDUNGI SEBELAS (11) MUKA SURAT

BAHAGIAN A: JAWAB SEMUA SOALAN (70 MARKAH)

S1 Apakah keluaran yang dihasilkan oleh setiap aturcara di bahagian (a) dan (b) berikut:

(a)

```
#include<stdio.h>
void fn (int, int);
int main (void) {
    int x = 3, y = 8;
    fn (x, y);
    printf("\nThe total value is : %d", x + y);
    return 0;
}
void fn (int k, int l){
    k = 6; l = 9;
    printf("\nThe total value is : %d", k + l);
}
```

(5 Markah)

(b)

```
#include<stdio.h>
int func (int );

int main (void) {
    int n, count;
    for (count = 1; count <= 4; count++){
        n = func (count);
        printf("%dt",n);
    }
    return 0;
}
int func (int z){
    int y;
    y = z * z;
    return (y);
}
```

(5 Markah)

- S2 Bayaran saman yang dikenakan oleh pihak MPBP adalah RM30.00 bagi setiap kesalahan letak kereta. Sempena Hari Diskaun, pihak MPBP memberikan diskaun bayaran saman mengikut jumlah tiket saman, seperti dalam Jadual S2:

Jadual S2: Bilangan tiket saman dan diskaun yang dikenakan oleh pihak MPBP

Bilangan tiket saman	Diskaun [%]
1 – 15	20
16 – 20	50
Lebih daripada 20	60

Berdasarkan jadual, tulis aturcara C menggunakan pilihan kawalan if-else untuk mengira jumlah bayaran saman. Bilangan tiket saman mestilah **diterima oleh pengguna**.

(20 Markah)

- S3 Berikut adalah dua fungsi untuk mengira lilitan dan luas bulatan dengan jejari sebagai kemasukan.

```
#include<stdio.h>
```

```
double circumference (double rad){
    const double PI = 3.1415;
    return (2 * PI * rad);
}
```

```
double area (double rad){
    const double PI = 3.1415;
    return (PI * pow (rad, 2));
}
```

Tulis fungsi utama untuk kedua-dua sub-fungsi di atas. Nilai jejari **diterima daripada pengguna** dan maklumat yang telah diproses oleh sub-fungsi hendaklah dipaparkan.

(20 Markah)

- S4 Tulis satu aturcara C bagi mengira nilai sisihan piawai, **s**. Data kemasukan adalah **n** yang mewakili bilangan nombor, dan beberapa nombor perpuluhan sebanyak **n**. Rumus **s** adalah diberi seperti yang berikut:

$$s = \sqrt{\frac{(x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2) - (n \times m^2)}{n-1}}$$

Pembolehubah **m** adalah purata daripada **n** nombor perpuluhan. Jujukan $x_1, x_2, x_3, \dots, x_n$ adalah nilai setiap nombor perpuluhan. Bagi menyelesaikan persoalan ini, anda dikehendaki menggunakan carta alir seperti dalam **Rajah S4**.

(20 Markah)

BAHAGIAN B: JAWAB SATU (1) DARIPADA DUA (2) SOALAN YANG BERIKUT

S5 Kaedah *Simpson Sepertiga* (RAJAH S5) boleh digunakan untuk mengira nilai kamiran secara numerik. Rumus kaedah berkaitan diberikan seperti yang berikut:

$$\int_{x_L}^{x_U} f(x) dx = \frac{h}{3} [f(x_L) + 4f(x_M) + f(x_U)] \quad (1)$$

Dimana:

$$h = \frac{x_U - x_L}{2} \quad \text{dan} \quad x_M = x_L + h \quad (2)$$

- (a) Tulis satu fungsi menggunakan C program bagi $f(x) = \frac{x}{1 + \sqrt{x}}$, dimana x tidak boleh negatif.
- (b) Tulis satu fungsi ***simpson(parameter)*** dengan parameter x_L untuk had bawah dan x_U untuk had atas. Fungsi ini memanggil fungsi ***f(x)*** untuk mengira nilai kamiran seperti di dalam persamaan (1).
- (c) Tulis satu fungsi utama yang memanggil fungsi ***simpson()***. Fungsi utama memerlukan nilai had bawah (x_0) dan had atas (x_n) bagi kamiran. Untuk menambah ketepatan numerik, bahagikan ***f(x)*** kepada N selang. Lebar setiap selang, w , adalah:

$$w = \frac{x_n - x_0}{N} \quad \text{dan} \quad x_n = x_{n-1} + w$$

Kaedah *Simpson Sepertiga* perlu diterapkan di dalam setiap selang. Papar nilai kamiran pada skrin dengan ketepatan enam (6) digit perpuluhan.

(30 Markah)

S6 Lakar satu carta alir yang sesuai bagi menggambarkan kod sumber C seperti yang berikut:

```
#include<stdio.h>
#include<conio.h>

int main(){
    int i,j;
    for ( i = 1; i <= 17; i += 1) {
        if ( i <= 3 || i >= 15) {
            for ( j = 1; j <= 20; j += 1)
                printf("*");
        }
        else if ( i > 3 && i <= 7 || i > 10 && i <= 14) {
            for ( j = 1; j <= 4; j += 1)
                printf("*");
        }
        else if ( i > 7 && i <= 10) {
            for ( j = 1; j <= 13; j += 1)
                printf("*");
        }
        printf("\n");
    }
    getch();
    return 0;
}
```

Keluaran aturcara adalah seperti yang di bawah ini:

```
*****
*****
*****
****
****
****
*****
*****
*****
****
****
****
*****
*****
*****
```

(30 Markah)

PART A: ANSWER ALL QUESTIONS (70 MARKS)

Q1 Write the output after execution of the C codes in part (a) and (b) below:

(a)

```
#include<stdio.h>
void fn (int, int);
int main (void) {
    int x = 3, y = 8;
    fn (x, y);
    printf("\nThe total value is : %d", x + y);
    return 0;
}
void fn (int k, int l){
    k = 6; l = 9;
    printf("\nThe total value is : %d", k + l);
}
```

(5 Marks)

(b)

```
#include<stdio.h>
int func (int );

int main (void) {
    int n, count;
    for (count = 1; count <= 4; count++){
        n = func (count);
        printf("%d\t",n);
    }
    return 0;
}
int func (int z){
    int y;
    y = z * z;
    return (y);
}
```

(5 Marks)

- Q2** Each offend against proper parking ruled by MPBP costs RM 30.00. On a Discount Day, MPBP offers the cost reduction of offends depending on the number of summon tickets as shown in Table Q2

Jadual S2: Number of tickets and discount charge by MPBP

Number of Tickets	Discount [%]
1 – 15	20
16 – 20	50
More than 20	60

Based on Table Q2, write a C program by using the selection control *if-else*, calculate and display the total amount. The amount of ticket must be **enter by user**.

(20 Marks)

- Q3** Given are two functions for calculating the circumference and the area of a circle. Radius is the input data.

```
#include<stdio.h>

double circumference (double rad){
    const double PI = 3.1415;
    return (2 * PI * rad);
}

double area (double rad){
    const double PI = 3.1415;
    return (PI * pow (rad, 2));
}
```

Write a *main function* for the sub functions. The value of radius is **inputted by user** in the *main function*. Display the results receive from the sub functions.

(20 Marks)

- Q4** Write a C program that first reads an integer value for n and then reads n floating –point values. After entering the last input, compute the standard deviation using the formula:

$$s = \sqrt{\frac{(x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2) - (n \times m^2)}{n-1}}$$

where m is the average of n numbers, and $x_1, x_2, x_3, \dots, x_n$ represent the floating point values. You shall use the flowchart given in **Rajah S4** to guide your answer.

(20 Marks)

PART B: ANSWER ANY TWO (2) OF THREE (3) QUESTIONS

- Q5** Method of *Simpson One-Third* (**RAJAH S5**) can be used to evaluate of a numerical integration of any function. The formulas of this method are shown as follows:

$$\int_{x_L}^{x_U} f(x) dx = \frac{h}{3} [f(x_L) + 4f(x_M) + f(x_U)] \quad (1)$$

Where:

$$h = \frac{x_U - x_L}{2} \quad \text{and} \quad x_M = x_L + h \quad (2)$$

- (a) Write a function using C programming for $f(x) = \frac{x}{1 + \sqrt{x}}$, where x must be positive.
- (b) Write a function of ***simpson(parameter)*** with a parameter x_L for the lower limit and x_U for the upper limit. This function calls ***f(x)*** in order to calculate the integration.
- (c) Write a main program to call ***simpson()***. The main program needs a lower limit (x_0) and an upper limit (x_n). For more accurate, divide the integration domain of $f(x)$ in N intervals. The interval width, w , is then:

$$w = \frac{x_n - x_0}{N} \quad \text{and} \quad x_n = x_{n-1} + w$$

The *Simpson One-Third* needs to be implemented in every interval. Display the value of the integration with six (6) decimal digits accuracy.

(30 Marks)

Q6 Draw the corresponding flow chart based on the source code below:

```
#include<stdio.h>
#include<conio.h>

int main(){
    int i,j;
    for ( i = 1; i <= 17; i += 1) {
        if ( i <= 3 || i >= 15) {
            for ( j = 1; j <= 20; j += 1)
                printf("*");
        }
        else if ( i > 3 && i <= 7 || i > 10 && i <= 14) {
            for ( j = 1; j <= 4; j += 1)
                printf("*");
        }
        else if ( i > 7 && i <= 10) {
            for ( j = 1; j <= 13; j += 1)
                printf("*");
        }
        printf("\n");
    }
    getch();
    return 0;
}
```

The output of the coding is given below:

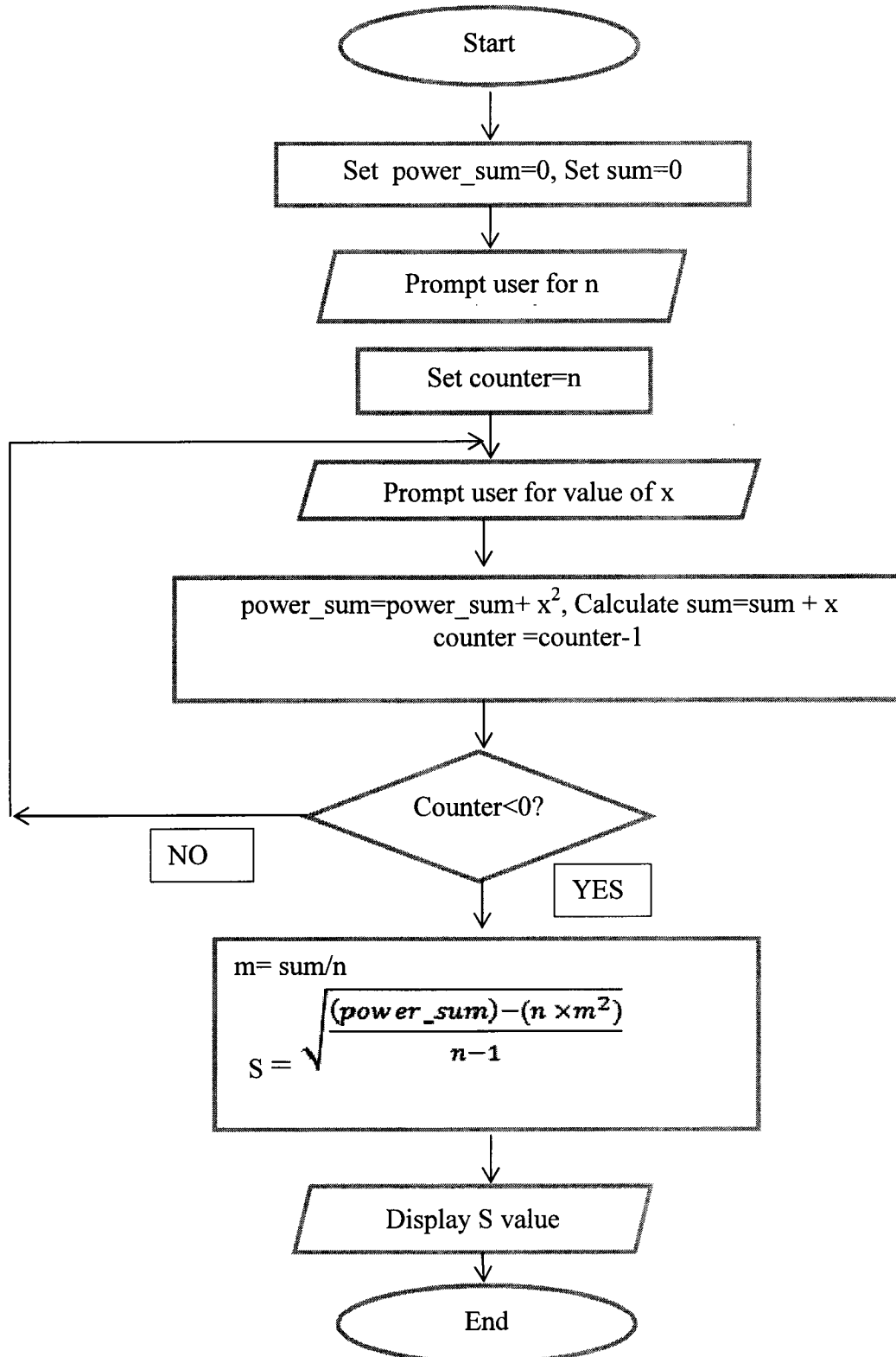
```
*****
*****
*****
****
****
****
*****
*****
*****
****
****
****
*****
*****
*****
```

(30 Marks)

PEPERIKSAAN AKHIR

SEMESTER / SESI : SEM II / 2010/2011
 NAMA KURSUS : PENGATURCARAAN KOMPUTER

PROGRAM : 1 BDD
 KOD KURSUS : BTI 10202

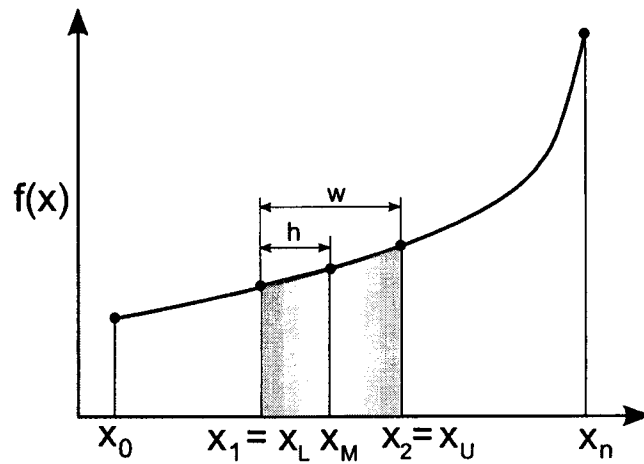


RAJAH S4

PEPERIKSAAN AKHIR

SEMESTER / SESI : SEM II / 2010/2011
NAMA KURSUS : PENGATURCARAAN KOMPUTER

PROGRAM : 1 BDD
KOD KURSUS : BTI 10202

**RAJAH S5**