

University of Ruhuna- Faculty of Technology
Bachelor of Information & Communication Technology degree
Level 1 (Semester 1) Examination, July 2017

Course Unit: ICT1133- Fundamental of Computer Programming

Time Allowed: 2 hours

Answer all four (04) questions

This question paper contains **07 pages**.

(1)

- a. Define four problem solving phases used in computer programming and briefly describe each of the phase.
- b. Write down three general rules for naming an identifier in C Language.
- c. Explain four basic built-in data types in C language by using suitable examples.
- d. Write down the value of **x** after each of the following C statements are executed.

i) $x = 7 \% 4 * 6 / (12 - 6) \% 3 ;$

ii) $x = 10 \% 3 * 4 + 5 * 2 ;$

- e.
 - i) What is an algorithm?
 - ii) Draw a flow chart which print “**Hello World**”10 times by using a repetition control structure.
- f. Write a C program to input principle amount (P), time (T) and rate (R) and calculate Compound Interest (CI).

Formula to calculate compound interest (CI) is given below,

$$CI = P \times \left(1 + \frac{R}{100}\right)^T - 1$$

Output should be formatted to **two decimal** places.

(2)

a. Identify and correct the errors in each of the following C code segment.

```
i) while ( d<= 4 ) {  
    product *= d  
    ++d;
```

```
ii) if ( gender == 1 );  
    puts( "Woman" );  
    else;  
    puts( "Man" );
```

b. Following incomplete C program is written to check whether a given character is a alphabet, digit or special character. Complete four **blank-lines** with suitable C statements to produce the following output.

Sample outputs are shown below;

```
#include <stdio.h>  
int main()  
{  
    char ch;  
    printf("Enter any character: ");  
    ----- //blank-line 1  
  
    // Alphabet checking condition  
    ----- //blank-line 2  
    {  
        printf("%c is ALPHABET.", ch);  
    }  
    // Check digit  
    ----- //blank-line 3  
  
    {  
        printf("%c is DIGIT.", ch);  
    }  
    else  
    {  
        ----- //blank-line 4  
    }  
    return 0; }
```

Enter any character: 6

6 is DIGIT

Enter any character: s

s is ALPHABET

Enter any character: #

is a special character

c. i) Explain two differences between a **while loop** and a **do-while loop** in C program with aid of simple examples.

ii) Write down the value of the variable (**sum** and **count**) when cone is executed.

```

int count=1;
sum=5;
while (count < 10) {
    sum += count;
    count +=3;
}

```

iii) Translate the above c(ii) **while loop** into an equivalent *for loop*.

- d. Generate the following nested if statements to equivalent switch-statement according to C language syntax.

Assume **Digit** variable is an integer and **value** variable is a character type.

```

if (Digit ==6)
    value= 'A';
else if ((Digit>=0) && (Digit<=2))
    value= 'C';
else if (( Digit == 4) || ( Digit == 5))
    value = 'B';
else
    value= 'D';

```

- e. Write a C program to print the following square stars (*) pattern using nested *for loops*.

```

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

```

(3)

- Write down three main benefits of using functions in a C program.
- What is the difference between **pass by value** and **pass by reference** approaches use in functions?
- In a school 10 students were selected for a volleyball team. The heights were recorded in an array to do an analysis about the student heights.

- i) Write a function called *inputHeights()* which will take *array name* and the *size of the array* as parameters and fill the array with height of students.
- ii) Write a function called *printHeights()* to print the height of all the students selected and the function should take the *array name* and the *size of the array* as parameters.
- iii) Write a function called *findAvgHeight()* to calculate the average height of the students and return it (return type is float). The function should take the *array name* and the *size of the array* as parameters.
- iv) Implement the main function to do the followings:
 - i. Declare prototypes of the function.
 - ii. Create an array with the size 10.
 - iii. Enter 10 heights of the students to the array using *inputHeights()* function.
 - iv. Print the height of the students using *printHeights()*.
 - v. Find and print the average heights of the students in the team using *findAvgHeight()*.
- d. Write down the difference between structures and arrays in C language.
- e. Following C program is declared *car_part* structure to store the details about car parts. Examine the program and briefly explain meaning of numbered lines shown in the table.
(Copy given table to your answer sheet)

Line No.

```
1 # include <stdio.h>
2 struct car_part
3 {
4     int part_num;
5     char color;
6     double price;
7 };
8 int main() {
9     int temp;
10    struct car_part partInfor;
11    partInfor.part_num = 100012;
12    partInfor.price = 4000.00;
13    printf("Enter color of the car");
14    scanf("%c",&partInfor.color);
15    temp = partInfor.part_num;
16    printf("Color of the car: %c \n",partInfor.color);
17    printf("Part number is :%d", temp);
18    return 0;
19 }
```

Line No.	Meaning
2	
4,5,6	
10	
11,12	
14	
15	
16	
17	

(4)

- a. What is a pointer variable?
- b. Examine the C code below and write the output of **LINE A**, **LINE B**, **LINE C** and **LINE D**.

```
# include<stdio.h>
int main(){
    int x, *p;
    p=&x;
    *p=0;
    printf("x is %d\n",x); //LINE A
    printf("*p is %d\n", *p); //LINE B
    *p+=1;
    printf("x is %d\n",x); //LINE C
    (*p)+ +;
    printf("x is %d\n",x); // LINE D
return 0;}
```

- c. Discuss about command line arguments in C language.

d.

```
./a.out hello world

Program name ./a.out
The argument supplied are hello world
```

Figure 1

Above **Figure 1** shows sample output of the program run at command line. Write a C program to produce the same output by examining above results. (You are required to pass the values from the command line).

e.

i) Following C program is used to count number of lines in a given file.

```
int main()
{
    FILE *fp;
    char ch;
    int linesCount=0;

    fp=fopen("ruhuna.txt","r");
    if(-----) // blank-line A
    {
        printf("File \"%s\" does not exist!!!\n","ruhuna.txt");
        return -1;
    }
    //read character by character and check for new line
    while((-----)// blank-line B
    {
        if(-----) // blank-line C

            linesCount++;
    }
    fclose(-----); // blank-line D

    printf("Total number of lines are: %d\n",-----);// blank-line E

    return 0;
}
```

Fill the blank-lines (**A, B, C, D and E**) in the program by getting eight code segments from the pool of code given below,

Pool of code:

```
{
fclose(fp), linesCount, ((ch=fgetc(fp)) !=EOF), ( (ch=fgetc(f))=EOF),
(ch=='\n'), ch !='\n', fp==NULL, fp!=NULL, ch =='\t', Linecount, fclose(f)
}
```

- ii) Explain four main steps in file processing by using the C program given above.

----- **End of the Paper** -----