

**UNIVERSITY OF RUHUNA**  
**BACHELOR OF COMPUTER SCIENCE (GENERAL) DEGREE**  
**LEVEL I (SEMESTER I) EXAMINATION – JUNE 2022**  
**COURSE UNIT: CSC 1113 (Programming Techniques) – Theory** **TIME: 2 HOURS**

---

Answer All four (04) Questions.

1.

a. Figure Q1a illustrate the stages of the C compilation Process.

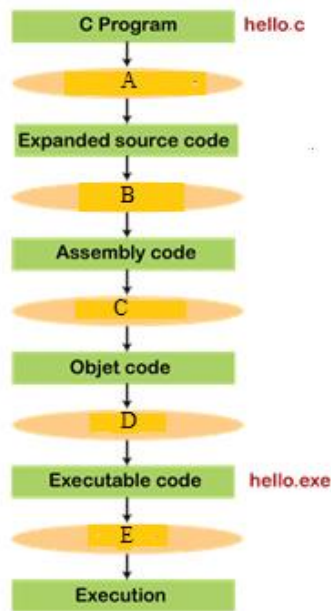


Figure Q1a

Name the component that associate with converting C Program to Execution stage given by A, B, C, D and E in Figure Q1a and briefly explain their tasks.

- b. Variables are simply names used to refer to some location in memory.
- i. State whether following variable declarations are valid or invalid and provide reasons for invalid declarations.
    - A. `int firstInput;`
    - B. `char $input;`
    - C. `float First Input=3.13f;`
    - D. `int _firstput, secput;`
    - E. `int first_input=10;`
  - ii.
    - A. Briefly explain what is meant by the scope of a variable.
    - B. Discuss the difference between scope of local variables and global variables using two (02) facts.

c. State whether following expressions are true or false if  $x = 10$ ,  $y=5$  and  $z=15$  in each expression.

- A.  $!(x != y)$
- B.  $(x == y) \parallel (x <= z)$
- C.  $(x > y) \&\& (y <= z)$
- D.  $!(x++ < z)$
- E.  $x+y < z$
- F.  $x+z <= y*y$
- G.  $!(z+2*y <= \text{pow}(y,2))$

d.

- i. Write a C program to check whether user is an adult or not according to below given instructions.
  - prompt user a message “Enter your age:” and take the entered age to a variable.
  - if entered age is 18 years or over than that, print the message “You are an adult”.
- ii. Implement the algorithm given in below Figure Q1d using C programming language to swap given values of X and Y.

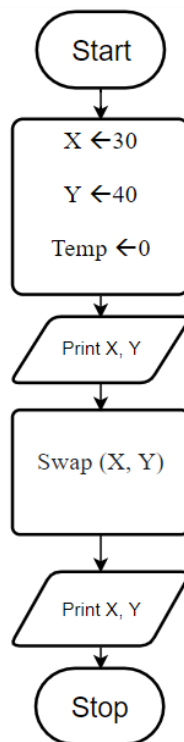


Figure Q1d

2.

a. Write down the expected output of the following programs given in A - D.

```
#include <stdio.h>

void main()
{
    for(int i=0;i<10;i++)
    {
        if(i==7)
            continue;
        printf("%d\n",i);
    }
}
```

A

```
#include <stdio.h>

void main()
{
    for(int i=0;i<10;i++)
    {
        if(i==6)
            break;
        printf("%d\n",i);
    }
}
```

B

```
#include <stdio.h>

void main()
{
    for(int i=0;i<10;i+=2)
    {
        printf("%d\n",i);
    }
}
```

C

```
#include <stdio.h>

void main()
{
    int a=13;
    int b = 4;
    printf("%d",a%b);
}
```

D

b. Write code segments required to obtain the output given below using repetition control structures.

**Note:** Use of pure sequential control structure to obtain the output will not be given marks.

i.     \*\*\*\*\*  
       \*\*\*\*\*  
       \*\*\*\*\*

ii.     \*  
       \*\*\*  
       \*\*\*\*\*  
       \*\*\*\*\*

- c. Write C programs to satisfy following requirements
- Obtain numbers between 1 to 10 using a repetition control structure. Display whether each of the number is odd or even.
  - Obtain two numbers as user inputs and display maximum out of given two numbers.

3.

- a. Briefly describe each part given in the Figure Q3a of a general function declaration.

```
return_type function_name( parameter_list )  
{  
    body of the function  
}
```

Figure Q3a

- b. List three (03) benefits of using functions in C Programs.
- c. There are two ways arguments can be passed to a function in C as **Call by value** and **Call by reference**.
- Briefly explain the difference between *Call by value* and *Call by reference* using two (02) facts.
  - Write a C program with a function to calculate the Area of a circle according to below given instructions. [Hint:  $Area = \pi(radius)^2$ ]
    - user should allow to *enter the radius* of the circle
    - function should *accept radius of the circle as a parameter* to calculate the Area of the circle
    - function should *call by value* with user input from main program
- d.
- Briefly explain what an array is.
  - Write a C program to initialize an array with list of numbers from 1 to 5 and display numbers given in the array in reverse order.
- e. A pointer is a variable whose value is the address of another variable.
- The Figure Q3d illustrates a C program written using pointers.

```

#include <stdio.h>

int main()
{
    int a=5;
    int b;

    int *ptr;    /** Statement A **/
    ptr=&a;      /** Statement B **/
    b=*ptr;      /** Statement C **/

    *ptr=10;

    printf("The value of a is: %d \n",a);
    printf("The value of b is: %d \n",b);

    return 0;
}

```

Figure Q3d

- A. Explain the functionality of statements labeled with A, B and C using comments.
  - B. Write the output of program in Figure Q3d.
- ii. Write C program statements to define an array of five characters and to declare a pointer to that array.

4.

a. State two advantages of using *structures* over *arrays* in C.

b.

- i. Explain two differences between a *structure* and a *union*.
- ii. What will be the size of following *union* declaration? (Assume size of int = 2 bytes, char = 1 byte, and float = 4 bytes)

```

union Test {
    int x;
    char y;
    float z;
}

```

iii. What is the output of the following code?

```
#include <stdio.h>
struct temp
{
    int a;
    int b;
} s;
void change(struct temp);
main()
{
    s.a = 10;
    s.b = 20;
    change(s);
    printf("%d %d \n", s.a, s.b);
}
void change(struct temp s)
{
    s.a = 1;
    s.b = 2;
}
```

iv. Consider the following structure definitions where *struct point* represents the x and y coordinates of a point in XY coordinate system. The points in the *struct rectangle* represent the points at the upper left and lower right corners of a rectangle. Write a function named *area* that takes a rectangle structure as an input argument and returns the area of the rectangle.

```
struct point {
    int x;
    int y;
};
```

```
struct rectangle {
    struct point upperLeft;
    struct point lowerRight;
};
```

c.

i. Define a structure in C to store the following details of a customer attached to a bank. Account Number, Customer Name, Account Balance and Account opening date. Assume Account Number is an integer, Account Name

contains at most 20 characters. Account Balance is of type *float* and opening date is a structure containing date, month and year.

- ii. Write a function in C that takes no input arguments but accepts account details from the keyboard and return a customer structure.
  - iii. Assuming there are 100 customers opened accounts, write a C code segment to count the number of customers whose Account Balance is Rs. 25000 or more.
- d. Explain the use of following functions.
- i. `fopen()`
  - ii. `fseek()`
  - iii. `ftell()`

\*\*\*\*\* END \*\*\*\*\*