

International Islamic University Chittagong
Department of Computer Science & Engineering
B. Sc. in CSE Semester Final Examination, Spring 2019
Course Code: CSE 1121 Course Title: Computer Programming I
Total marks: 50 Time: 2 Hours 30 minutes

[Answer any two questions from **Group-A** and any three questions from **Group-B**; Separate answer script must be used for Group-A and Group-B; Figures in the right hand margin indicate full marks.]

Group-A

- 1.a) Write C code segment to print all the *even* numbers from 1 to 100 inclusive in *decreasing* order using **for** loop. Rewrite the same using **while** and **do..while** loop. 3
- b) Describe the output generated by the following program. 2.5
- | | |
|--|--|
| i)
int main()
{
int i = 0;
for(;i<=10;i++);
printf("%d\n",i);
return 0;
} | ii)
int main()
{
int i,j;
for(i = 0; i < 3; i++)
for(j = 0; j < 3; j++)
printf("%d %d\n", i, j);
return 0;
} |
|--|--|
- c) Explain why we should avoid *goto* statement in big programs. 1.5
- d) You will be given some positive integer inputs followed by a negative value. You have to print the sum of the given positive inputs except the ones which are divided by 5 or 7. You have to calculate their average also. Write a C program for this by using *break* statement after getting the negative input and *continue* statement if the input is divided by 5 or 7. 3
- 2.a) Finding errors in a program is easier when we use functions. Do you agree? Give argument in support of your answer. 2
- b) What are the differences between *local variable* and *global variable*? Explain with examples. 2.5
- c) What do you mean by *recursion*? Explain in which cases the given recursive function will cause a program crash. Also explain in which cases it will work properly. 2.5
- ```
void f(int x)
{
 if(x!=0)
 f(x+1);
 printf("%d ",x);
}
```
- d) Write a function called **divisor** that determines for a pair of integers whether the second integer is a divisor of the first. The function should take two integer arguments and return 1 (true) if the second is a divisor of the first, and 0 (false) otherwise. Demonstrate your function in a complete program. 3
3. a) What do you mean by *storage class*? Briefly discuss static storage class with example. 1.5
- b) What do you mean by *function prototypes*? Where within a program are function prototypes normally placed? Give example. 1.5
- c) Write the output generated by the following programs- 4

```

i)
#include<stdio.h>
int x=5;
int fun1()
{ x = x +20; return x; }

int fun2()
{ int x = 10; return x; }

int fun3()
{ x = x - 10; return x; }
int main()
{
 x = 1;
 printf("x = %d\n", x);
 printf("x = %d\n", fun1());
 printf("x = %d\n", fun2());
 printf("x = %d\n", fun3());
 return 0;
}

```

```

ii)
#include<stdio.h>
void f1(int a);
void f2(int b);
int main()
{
 f1(10);
 return 0;
}
void f1(int a)
{
 if(a!=0)
 f2(a-1);
 printf(".");
}
void f2(int b)
{
 if(b!=0)
 f1(b-1);
 printf("%d",b);
}

```

- d) Write a C program that will read a positive integer  $n$  and determine whether  $n$  is *prime* or not. 3

### Group-B

4. a) Write an appropriate array definition for each of the following situations. 2
- i) Define a one dimensional, character array called **direction**. Assign the string "WEST" to the array elements. End the string with null character.
  - ii) Define a two-dimensional, 3 X 4 integer array called **table**. Assign the following values to the array elements.
 

|    |    |    |   |
|----|----|----|---|
| 10 | 12 | 14 | 0 |
| 0  | 20 | 22 | 0 |
| 0  | 30 | 32 | 0 |
- b) How an array can be *passed* to a function? Why does it *passed by reference* by default? 2
- c) Let, two strings are given as user input. Perform the following operations on the strings by writing one or two lines of C code: 3
- a) Find out sum of **length** of the two strings.
  - b) **Concatenate** second string onto the end of first string.
  - c) **Copy** second string to the first string.
  - d) **Compare** the two strings and determine which one is greater.
- d) Take  $n$  numbers as input and store them in an array. Then input another number  $X$ . Now, print all those numbers which are larger than  $X$  in a single line. In the next line, print all those numbers which are smaller than  $X$ . For Example, let  $n = 6$  and the numbers are 5, 7, 4, 6, 2, 4 and  $X = 6$ . So, the output will be 3
- ```

5 7
4 2 4

```

5. a) What is a *pointer*? What are the purposes of *address operator* (&) and the *indirection operator* (*)? Explain with example. 2
- b) When passing an argument to a function, what are the differences between *passing by value* and *passing by reference*? Explain with a simple C program. 3
- c) What is the difference between *structure* and *union*? Explain using a suitable example. 2
- d) Write a C program using *array of structure* that will allow you to enter and display the following information about the students of a class: 3
 i) *name* ii) *ID* iii) *CGPA* iv) *department* v) *section*
6. a) What is a *data file*? Briefly describe different types of data files in C. 2
- b) Suppose, a and b are unsigned, 16-bit integer quantity whose hexadecimal values are **0x6db7** and **0x8ae1**. Evaluate each of the following bitwise expressions: 3
 i) $a | b$ ii) $a \wedge b$ iii) $a \& b$ iv) $\sim a$ v) $b = a \gg 4$ vi) $b = a \ll 3$
- c) What is a *macro*? Write the advantages and disadvantages of using macro over *function*. 2
- d) Open a file and write 10 numbers taken from keyboard. Next, close the file and open it again in read mode. Now, read the numbers from the file and print them in *reverse* order. 3
7. a) If we want to create an array of size N where N will be taken as input, how the *dynamic memory allocation* helps us with that? Write proper code segment. 1
- b) Write a function to *copy* the content of a given string to another string without using library function *strcpy()*. 2.5
- c) Write the output generated by the following program- 2.5
- ```
#include<stdio.h>
int n = 3, found = 0;
char maze[5][5] = { "R..",
 "##.",
 ".#G" };

void visit(int i, int j)
{
 printf("%d %d\n", i , j);
 if(maze[i][j] == 'G') {
 found = 1; return;
 }
 if(maze[i+1][j] != '#' && i+1 < n) visit(i + 1, j);
 if(maze[i][j+1] != '#' && j+1 < n) visit(i, j + 1);
}

int main()
{
 visit(0,0);
 return 0;
}
```
- d) Write a C program that reads n numbers from keyboard, store in an array and *rearrange* the numbers in *ascending* order and then display the list. 4