भारतीय सूचना प्रौद्योगिकी संस्थान गुवाहाटी

Indian Institute of Information Technology Guwahati

COMPUTER PROGRAMMING LAB (CS110)

ASSIGNMENTS–11 : REVISE

1. Write separate programs in C to print the following patterns. Each of them is associated with a control variable `n`. The examples are associated with `n = 4`.

   i.
   ```
   ****
      *
     *
   ****
   ```

   ii.
   ```
   ****
   *  *
   *  *
   ****
   ```

iii.
```
<<<<(       )   >
<<< ((     ))  >>
<<  (((  ))) >>>
<   ((((()))))>>>>
```

iv.
```
      1
     1 1
    1 0 1
   1 0 0 1
  1  1 1 1 1
```

2. Compute the sum of the series for the first n terms:

$$S = \frac{1}{0!} + \frac{1}{4!} + \frac{1}{8!} + \frac{1}{12!} + \frac{1}{16!} + \cdots$$

3. Compute the sum of the series for the first n terms:

$$S = \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \cdots$$

4. Write a C program to input a decimal integer and convert it to a binary number system. Example: if you have

   ```
   int n = 17;
   ```

   as input, you need to create another `int` variable with the value 10001.

5. Write a function in C that takes a positive integer as input and returns the leading digit in its decimal representation. For example, the leading digit of 234567 is 2.

6. Write a recursive function in C to find the sum of all even numbers in a given range.

7. Write a recursive function in C to find the least common multiple (LCM) of two numbers.

8. Write a function in C to add a new value in a sorted array. After the operation, the modified array needs to be sorted.

9. Write a function in C to delete an element at the desired position from an array, considering it a list.

10. Write a function in C to check whether an array is a subset of another array.

11. Write a c program to sort an array of structure student as per their roll number. Structure contains following members
    char Name[30]; int roll; float CPI;

12. Realize the following program:

```c
#include <stdio.h>

int x; // can be accessed from outside this file by declaring it using extern
static int y; // access restricted to this file

void f() { // can be accessed from outside this file
    static int count; // accessible only to this function; default value is 0
    count++;
    printf("%s is called %d time(s).\n", __func__, count);
    return;
}

static void g() { // access restricted to this file
    printf("Inside %s\n", __func__);
}

void main(void) {
    void f();
    printf("x = %d, y = %d\n", x, y); // default value is 0
    f();
    f();
    f();
}
```

13. Realize the following program:

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

int x = 1, y;
static int a = 2, b;

void f(){}

static void g();

int main(int argc, char *argv[]) {
    static int u = 3, v;
    int s = 4, t;
    printf("CODE/TEXT SEGMENT (LOW MEMORY):\n");
    printf("f      = %p\n", f);
    printf("main   = %p\n", main);
    printf("g      = %p\n", g);
    printf("printf = %p (library function)\n", printf);
    printf("\n");
    printf("DATA SEGMENT (INITIALIZED):\n");
    printf("x = %d, &x = %p (external) \n", x, &x);
    printf("a = %d, &a = %p (static) \n", a, &a);
    printf("u = %d, &u = %p (static, local to main)\n", u, &u);
```

```
    printf("\n");
    printf("DATA SEGMENT (UNINITIALIZED):\n");
    printf("y = %d, &y = %p (external)\n", y, &y);
    printf("b = %d, &b = %p (static)\n", b, &b);
    printf("v = %d, &v = %p (static, local to main)\n", v, &v);
    printf("\n");
    printf("HEAP:\n");
    printf("address = %p\n", calloc(1, 1));
    printf("\n");
    printf("STACK SEGMENT (INITIALIZED/UNINITIALIZED):\n");
    printf("s = %d, &s = %p\n", s, &s);
    printf("t = %d, &t = %p\n", t, &t);
    g();
    printf("\n");
    printf(
        "+-------------------------+\n"
        "|          STACK          | (HIGH MEMORY)\n"
        "+-------------------------+\n"
        "|            |            |\n"
        "|            V            |\n"
        "|                         |\n"
        "|                         |\n"
        "|            ^            |\n"
        "|            |            |\n"
        "+-------------------------+\n"
        "|          HEAP           |\n"
        "+-------------------------+\n"
        "| UNINITIALIZED DATA (BSS)|\n"
        "+-------------------------+\n"
        "| INITIALIZED DATA (DATA) |\n"
        "+-------------------------+\n"
        "|    TEXT/CODE SEGMENT    | (LOW MEMORY)\n"
        "+-------------------------+\n\n"
    );
    printf("Block Starting Symbol (BSS) portion contains "
           "statically-allocated variables."
    );
    return 0;
}

static void g(){
    int i = 1;
    printf("i = %d, &i = %p (stack grows)\n", i, &i);
}
```