

# ESS111 : Programming 1 (C Programming)

## LAB - 7

Due: 27 January, 2021 @ 11:59 pm

### Part A (to be submitted)

**Problem 1:** Rinku and Ravi are playing a game using a bunch of coins  $n$ . Rules for the game are as follows:

1. A player can only make one move at a time.
2. In each move, a player gets to pick  $t$  number of coins, where  $t = 2^m, m \geq 0$  i.e. each player can pick  $t$  number of coins from the sequence  $\{1, 2, 4, 8, \dots\}$  while  $t \leq n$ .
3. The player who picks the last coin in their move, wins the game.

Given each player plays optimally, write a (C) program that takes 2 positive integers  $n, id$  as input where  $n$  is the total number of coins at the start of the game and  $id$  depicts which player has started the game, output the name of the player would win the game.

*Note:* Rinku is player1 hence  $id$  being 1 depicts Rinku started the game and while  $id$  being 2 depicts Ravi started the game. It is guaranteed that,  $1 \leq n \leq 30$  and  $id \in \{1, 2\}$ .

**Sample Input 1:**

10 1

**Output 1:**

Rinku

**Sample Input 2:**

3 2

**Output 2:**

Rinku

**Problem 2:** Write a (C) program with a recursive function with prototype `int is_palindrome(int x)`. The function `is_palindrome` returns 1 if the input number  $x$  is a palindrome, and 0 elsewhere. The function `is_palindrome` should be called in the main function. If the function `is_palindrome` returns 1, output Yes else output No.

*Note:* A number is a palindrome if it remains the same when read from either right to left or left to right. The input  $x$  is guaranteed to be in the range of `int` datatype.

**Sample Input 1:**

1805781

**Output 1:**

No

**Sample Input 2:**

975579

**Output 2:**

Yes

**Problem 3:** Write a (C) program to compute the greatest common divisor of two positive integers  $m$  and  $n$  given by Euclid's algorithm using a recursive function.

*Hint:* Euclid's algorithm works on the principle  $\text{gcd}(a, b) = \text{gcd}(a, b \% a)$  when  $a \leq b$ .

**Sample Input 1:**

91 56

**Output 1:**

7

**Sample Input 2:**

67 50

**Output 2:**

1

**Problem 4:** There are three pegs labeled A, B and C. Initially there are  $n$  disks placed on peg A. The bottom-most disk is largest, and disks go on decreasing in size with the topmost disk being smallest. The objective of the game is to move the disks from peg A to peg C, using peg B as an auxiliary peg. The rules of the game are as follows:

1. Only one disk may be moved at a time, and it must be the top disk on one of the pegs.
2. A larger disk should never be placed on the top of a smaller disk.

Write a (C) program to count the minimum number of steps required to move all disks from peg A to peg C given the  $n$ , the initial number of disks on peg A.

*Note:* It is guaranteed that  $1 \leq n \leq 30$ .

**Sample Input 1:**

1

**Output 1:**

1

**Sample Input 2:**

4

**Output 2:**

15

## **Part B (need not be submitted)**

1. What is the output of the following programs:

- (a)
- ```
#include <stdio.h>
int main()
{
    int x = 1, y = 2;
    int *ip;
    ip = &x;
    y = *ip;
    *ip = 0;
    printf("%d %d\n", x, y);
}
```
- (b)
- ```
#include <stdio.h>
int i = 0;
void val();
int main()
{
    printf("main's i = %d\n", i);
    i++;
    val();
    printf("main's i = %d\n", i);
    val();
    return 0;
}

void val( )
{
    i = 100;
    printf ( "val's i = %d\n", i );
    i++;
}
```
- (c)
- ```
#include <stdio.h>
int main()
{
```

```

        static int count = 5;
        printf("count=%d\n", count--);
        if(count != 0)
            main();
        return 0;
    }

```

```

(d)  #include <stdio.h>
      void func();
      int main()
      {
          func();
          func();
          return 0;
      }
      void func()
      {
          auto int i = 0;
          register int j = 0;
          static int k = 0;
          i++; j++; k++;
          printf("%d %d %d\n", i, j, k);
      }

```

2. Consider the following statements and indicate which of them is/are correct :

- (a) Comparing two pointers is a valid operation.
- (b) Adding a pointer with a positive or negative number is a valid operation.
- (c) Adding two pointers is a valid operation.
- (d) Multiplying two pointers is a valid operation

3. State whether the following statements are True or False:

- (a) The value of an automatic storage class variable persists between various function invocations.
- (b) If the CPU registers are not available, the register storage class variables are treated as static storage class variables.
- (c) The register storage class variables cannot hold float values.
- (d) If we try to use register storage class for a **float** variable the compiler will report an error message.

- (e) The default value for automatic variable is zero.
- (f) The life of static variable is till the control remains within the block in which it is defined.
- (g) If a global variable is to be defined, then the **extern** keyword is necessary in its declaration.
- (h) The address of register variable is not accessible.