PLACEMENTS KEY'S

Coding Challenge #1 (Question)

Day - 1

Ranjan wants to develop an application which prints the following pattern based on the required size. Ranjan got struck while building the logic and somewhere he missed the code. So, help him in order to print the required pattern as shown below.

Input format:

The first line contains the two integers "m" and "n".

Output format:

Pattern of required size.

Sample input 0:

3 4

Sample output 0:

```
  *
 *3*
*44*
*555*
*6666*
*555*
*44*
 *3*
  *
```

Explanation:

The pattern should start and end with same number as mentioned as "m". Where size of the pattern depends on value of "n".

PLACEMENTS KEY'S

Coding Challenge #1 (Question contd.)

Day - 1

Sample input 1:

10 10

Sample output 2:

```
   *
*10*
*1111*
*121212*
*13131313*
*1414141414*
*151515151515*
*16161616161616*
*1717171717171717*
*181818181818181818*
*19191919191919191919*
*181818181818181818*
*1717171717171717*
*16161616161616*
*151515151515*
*1414141414*
*13131313*
*121212*
*1111*
*10*
   *
```

PLACEMENTS KEY'S

Coding Challenge #1 (Question contd.)

Day - 1

Complete the code in the below Pattern_print() function:

```
#include<stdio.h>
int Pattern_print(int start,int size);
int Pattern_print(int start,int size)
{
        /*Complete the code*/
}
int main()
{
        int m,n,count;
        printf("Enter the size of m and n: "); //enter m and n
        scanf("%d %d",&m,&n);
        Pattern_print(m,n);  //the function calls is here
}
```

PLACEMENTS KEY'S

Coding Challenge #1 (C Solution)

Day - 1

```c
#include<stdio.h>
int Pattern_print(int start,int size);
int Pattern_print(int start,int size)
{
        printf(" *\n");
        for(int i=0;i<size;i++,printf("\n"))
        {
                printf("*");
                for(int j=0;j<=i;j++)
                {                                       //first part of pattern logic
                        printf("%d",start);
                }
                printf("*");
                start++;
        }
```

PLACEMENTS KEY'S

Coding Challenge #1 (C Solution contd.)

Day - 1

```
start=start-1;                    //the start value is decremented
        for(int i=size-1;i>=1;i--,printf("\n"))
        {
                start--;
                printf("*");
                for(int j=i;j>=1;j--)     //second part of the pattern logic
                 printf("%d", start);
                printf("*");
        }
        printf(" *");
}
int main()
{
        int m, n, count;
        printf("Enter the size of m and n: "); //enter m and n
        scanf("%d %d",&m,&n);
       Pattern_print(m,n);  //the function calls is here
}
```

PLACEMENTS KEY'S

Coding Challenge #1 (JAVA Solution)

Day - 1

```
import java.util.*;
public class PatternPrint
{
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.println("Enter starting number and no of succeding numbers..");
		int x,y;
		x = input.nextInt();
		y = input.nextInt();
		pattern(x,y);
	}
	public static void pattern(int x,int y){
		System.out.println("*");
			for(int i = 1;i<y;i++){
				System.out.print("*");
				for(int j=0;j<i;j++){
					System.out.print(x);
				}
				System.out.println("*");
				x++;
			}
```

PLACEMENTS KEY'S

Coding Challenge #1 (JAVA Solution contd.)

Day - 1

```
for(int i =y;i>0;i--){
                        System.out.print("*");
                        for(int j=0;j<i;j++){
                                System.out.print(x);
                        }
                        System.out.println("*");
                        x--;
                }
                System.out.println("*");
        }
}
```

PLACEMENTS KEY'S

Coding Challenge #2 (Question)

Day - 1

Given an array, of size, reverse it.

Example: If array = [1,2,3,4,5], after reversing it, array = [5,4,3,2,1].

Input Format:

The first line contains an integer "size", denoting the size of the array. The next line contains n(=size) space-separated integers denoting the elements of the array.

Constraints:

$1 \leq n \leq 1000$

$1 \leq arr_{ith} \leq 1000$, where is the arr_{ith} element of the array.

Output Format:

The output should contain the reverse array of the given array.

Sample Input 0:

6

16 13 7 2 1 12

Sample Output 0:

12 1 2 7 13 16

Explanation 0:

Given array = [16,13,7,2,1,12]. After reversing the array = [12,1,2,7,13,16]

PLACEMENTS KEY'S

Coding Challenge #2 (Question contd.)

Day - 1

Sample Input 1:

7

1 13 15 20 12 13 2

Sample Output 1:

2 13 12 20 15 13 1

Sample Input 2:

8

15 5 16 15 17 11 5 11

Sample Output 2:

11 5 11 17 15 16 5 15

PLACEMENTS KEY'S

Coding Challenge #2 (Question contd.)

Day - 1

Complete the logic in below Reverse_array function():

```
#include<stdio.h>
int Reverse_array(int size,int arr[]);
int Reverse_array(int size,int arr[])  //the function start here
{
        /*complete the code*/
}
int main()                    //main()
{
int arr[1000],size;
printf("Enter the size: ");
scanf("%d",&size);            //enter the size of array
for(int i=0;i<size;i++)
 scanf("%d",&arr[i]);         //enter the numbers into array
Reverse_array(size,&arr[0]);   //the function call is here
for(int i=0;i<size;i++)       //After reversing the array
 printf("%d ",arr[i]);        //Array gets printed
}
```

PLACEMENTS KEY'S

Coding Challenge #2 (C Solution)

Day - 1

```
#include<stdio.h>
int Reverse_array(int size,int arr[]);
int Reverse_array(int size,int arr[])  //the function start here
{
        if(size%2==1)                  //when size is odd
        {
        for(int i=0;i<=size/2;i++)
        {
         int temp=arr[size-i-1];    //the last element is stored in temp
        arr[size-i-1]=arr[i];  //the ith index array element is stored in (size-i-1)th array index
        arr[i]=temp;         //the temp value is stored in ith array index
           }
        }
        else        //when size is even
        {
                for(int i=0;i<=(size/2)-1;i++)
                {
                        int temp=arr[size-i-1];
                        arr[size-i-1]=arr[i];
                        arr[i]=temp;
                }
        }
}
```

PLACEMENTS KEY'S

Coding Challenge #2 (C Solution contd.)

Day - 1

```
int main()                    //main()
{
        int arr[1000],size;
        printf("Enter the size: ");
        scanf("%d",&size);          //enter the size of array

        for(int i=0;i<size;i++)
         scanf("%d",&arr[i]);       //enter the numbers into array

        Reverse_array(size,&arr[0]);   //the function call is here

        for(int i=0;i<size;i++)       //After reversing the array
         printf("%d ",arr[i]);        //Array gets printed
}
```

PLACEMENTS KEY'S

Coding Challenge #2 (JAVA Solution)

Day - 1

```
public class ReverseString
{
	public static void reverseString(int A[],int n){
		for(int j=0;j<n/2;j++){
			int temp = 0;
			temp = A[n-1-j];
			A[n-1-j]=A[j];
			A[j] = temp;
		}
	}
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.println("Enter the size of string");
		int size = input.nextInt();
		System.out.println("Enter the elements");
		int array[] = new int[size];
		for(int i=0;i<size;i++){
			array[i] = input.nextInt();
		}
		ReverseString m = new ReverseString();
		m.reverseString(array,size);
		for(int i=0;i<size;i++){
			System.out.print(array[i] + " ");
		}
	}
}
```