

***Instructions:***

- This lab is based on the topics: Looping with for, while and do-while.
- You should save each program with the file name as specified against each problem as <Lab#>-<Assignment#>-<Roll#>.c. For example, **04-01-24NA10006.c** to save Program to 1<sup>st</sup> assignment in Lab 4 with Roll Number 24NA10006
- You should upload each program to the Moodle system. Also, copy + paste your programs to the text window on the test page.
- A few test cases against each problem is given for your references and but not limited to.
- There are three problems and the maximum time allowed is 120 minutes.
- **Do not use arrays or any function in this lab.**

1. Write a program that takes a positive integer n and displays the first n terms of the following series: 1, -1, 2, -3, 5, -8, 13, -21, ...

You need to handle cases when the user enters 0 or negative numbers by giving appropriate messages.

[25]

**Test cases:**

#	INPUT	OUTPUT
1	3	1, -1, 2
2	0	N is invalid
3	10	1, -1, 2, -3, 5, -8, 13, -21, 34, -55
4	-2	N is invalid

```
// Code creator: Nishkal Prakash (nishkal@iitkgp.ac.in)
// Code to print fibonnachi sequence
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n); // Taking input from the user

    if (n <= 0) // If the input is 0 or negative, then it is invalid
        return printf("N is invalid");

    int a = 0, b = 1, c; // a, b, c are the first three terms of the series
    printf("%d", b);
    for (int i = 1; i < n; i++) // Looping from 1 to n-1 (1st term is already
        printed)
    {
        c = a + b;
        a = b;
        b = c;
        (i % 2) ? printf(", %d", -c) : printf(", %d", c); // If i is odd, then
        print -c, else print c
    }

    return 0;
}
```

- Write a program that will prompt the user to enter a sequence of positive integer values. At any time, your program should display the two largest values so far entered. When the user has entered only one value, for the second largest number your program should display "Value not yet entered". Your program should terminate when the user enters any negative number.

[40]

For example, one test case may be as follows:

```
Enter numbers: 1 ↵
Largest number: 1
Second largest number: Value Not yet entered
Enter numbers: 1 2 5 4 ↵
Largest number: 5
Second largest number: 4
Enter numbers: 1 2 5 4 8 2 ↵
Largest number: 8
Second largest number: 5
Enter numbers: 1 2 5 4 8 2 1 1 ↵
Largest number: 8
Second largest number: 5
Enter numbers: 1 2 5 4 8 2 1 1 -10 ↵
<Program terminates>
```

Here ↵ denotes pressing of enter key.

#### Test cases:

(assume enter is pressed after each number of the INPUT)

#	INPUT	OUTPUT
1	1 2 3 -1	Largest number: 1 Second largest number: Value not yet entered  Largest number: 2 Second largest number: 1  Largest number: 3 Second largest number: 2
2	10 -1	Largest number: 10 Second largest number: Value not yet entered
3	1 10 10 10 -1	Largest number: 1 Second largest number: Value not yet entered  Largest number: 10 Second largest number: 1  Largest number: 10 Second largest number: 10  Largest number: 10 Second largest number: 10
4	-1	

```

// Code creator: Nishkal Prakash
// Code to print Largest and Second Largest
#include <stdio.h>

int main()
{
    // max1 stores the largest number, max2 stores the second largest number
    int max1 = -1, max2 = -1, num;
    while (1)
    {
        scanf("%d", &num); // Taking input from the user
        if (num < 0) // If the input is negative, then terminate the program
            break;
        if (num > max1){ // If the input is greater than the largest number,
            then update the largest and second largest numbers
                max2 = max1;
                max1 = num;
        }
        else if (num > max2) // If the input is greater than the second largest
            number, then update the second largest number
                max2 = num;
        printf("\nLargest number: %d\n", max1);
        printf("Second largest number: ");
        if (max2 == -1) // If the second largest number is -1, then the second
            largest number is not yet entered
                printf("Value not yet entered\n");
        else
            printf("%d\n", max2);
    }
    return 0;
}

```

3. Write a program to print the “Numero-Pyramid” for a given height (also called the number of rows) and with digits from 0, 1, ..., 9 as the constituent elements (see the output in test cases). Your program should read the height of a Numero-Pyramid and then display it on the screen.

[Hint: Develop a formula to generate the output for each row.]

[35]

**Test cases:**

#	INPUT	OUTPUT
1	Enter the number of rows: 7	<pre> 1 232 34543 4567654 567898765 67890109876 7890123210987 </pre>
2	Enter the number of rows: 10	<pre> 1 232 34543 4567654 567898765 67890109876 7890123210987 890123454321098 90123456765432109 0123456789876543210 </pre>

```

// Code creator: Nishkal Prakash (nishkal@iitkgp.ac.in)
// Code to print the Numero-Pyramid
#include <stdio.h>
int main()
{
    // n is the number of rows, i, j, k, l are loop variables
    int n, i, j, k, l;
    printf("Enter the number of rows: ");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n - i; j++)
            printf(" "); // Printing spaces
        for (k = i; k < 2 * i; k++)
            printf("%d", k % 10); // Printing the first half of the numbers
        for (l = 2 * i - 2; l >= i; l--)
            printf("%d", l % 10); // Printing the second half of the numbers
        printf("\n");
    }
    return 0;
}

```

---\*---