

---

---

# Lab Exercise 5

— CSDC102: Intermediate  
Programming —

---

---

# Before your codes...

```
//*****  
// Filename :  
// Date :  
// Subject :  
// Second Semester, SY 2019 - 2020  
// Activity : Lab 1A  
// Problem Title :  
// Input :  
// Output :  
//  
// Honor Code : *insert honor code here*  
//  
// Complete Name :  
// ID Number :  
// Year-Course : 1-BSCS  
// DCS, College of Computer Studies  
// Ateneo de Naga University  
//*****
```

Honor Code : This is my own program. I have not received any unauthorized help in completing this work. I have not copied from my classmate, friend, nor any unauthorized resource. I am well aware of the policies stipulated in the handbook regarding academic dishonesty. If proven guilty, I won't be credited any points for this exercise.

# Lab 5A: Non-Fib

## Program Description:

- Alex is a teacher of hacking class and was supposed to take an extra lecture of 5th sem IT. He had already taken two lectures that day so he was very tired. He could not give free lecture to students because they were very much talkative. So he decided to give the students a task.
- He quickly asked the class to find the *nth non-fibonacci number*.

# Lab 5A: Non-Fib

- Specifications:
  - Filename: **Lab5A\_SURNAME1\_SURNAME2.cpp**
  - Input file: **alex.in**
  - Output: standard output

# Lab 5A: Non-Fib

- Formats:

- Input:

- First line contains number  $t$ , total number of test cases.
    - Then next  $t$  lines contain one number  $n$  for which you have to find  $n$ th non-fibonacci number.

- Constraints:

- $0 < t \leq 1000000$
    - $0 < n \leq 100000000$

- Output:

- $t$  lines containing  $n$ th non-fibonacci number for each test cases.

# Lab 5A: Non-Fib

Sample Input

5

1

2

3

4

5

Sample Output

4

6

7

9

10

# Lab 5B: Counting Mountains

## Program Description:

- Let:

$\wedge$  be a mountain.

Also, let:



be a bigger mountain.

Using this, we can create a mountain range,



The mountain range above is composed of 3 mountains, two in the front and one at the back. Create a program that counts the number of front mountains in a given mountain range.



# Lab 5B: Counting Mountains

- Formats:
  - Input:
    - Each judge file will contain a mountain range. Assume that (1) you will never see the foot of all mountains located at the back and (2) the front mountains do not overlap with each other.
    - The input file starts with an integer  $N$  denoting the height of the tallest mountain.

# Lab 5B: Counting Mountains

- Formats:
  - Constraints:
    - $1 \leq H \leq 7$  where  $H$  is the height of the tallest mountain
  - Output:
    - Print a single integer showing the count of all mountains in front.

# Lab 5B: Counting Mountains

- Specifications:
  - Implement using arrays/vectors. Program should iterate through the vector using the standard pointer dereference operator (\*). Using any *for loop* elements is NOT allowed. Program should contain functions that perform subtasks of your solution.
  - Filename: **Lab5B\_SURNAME1\_SURNAME2.cpp**
  - Input file: standard input
  - Output: standard output

# Lab 5B: Counting Mountains

## Sample Input

4



## Sample Output

4

# Lab 5C: Buildings

## Program Description:

- Yen was assigned in Metro Manila, Philippines to advertise her company's product. There are  $S$  streets in the city. Each day she would travel one street. There are  $N$  buildings in a street which are located at points  $1, 2, 3, \dots, N$ , respectively. Each building has a corresponding height  $h$  in meters. Yen stands at point  $0$  and her height is  $0.5m$ . Yen then starts communicating with the people of each building.

# Lab 5C: Buildings

## Program Description:

- She can communicate with people of a particular building only if she can see that building. If she succeeds in communicating with any building, then her boss gives her  $P$  Pesos, i.e., if she successfully communicates with  $K$  buildings in a day, then she will earn  $K * P$  Pesos. Now, She wants to know her maximum possible earning for each day.
- Note:
  - All the buildings are on a straight line and Yen is always standing at point  $O$ .

# Lab 5C: Buildings

- Formats:
  - Input:
    - Input begins with an integer  $N$  denoting the number of streets in the city.
    - Details for each street is described as follows:
      - First line contains an integer  $N$  and a double  $P$ , where  $N$  is the number of buildings found on the street and  $P$  is the corresponding earning for communicating successfully with one building.
      - The next line contains  $N$  space-separated integers  $h$  denoting the heights of the buildings.

# Lab 5C: Buildings

- Formats:
  - Output:
    - Print  $S$  lines, each indicating the maximum earning in the corresponding street. Earning printed should have two values after the decimal point.
  - Constraints:
    - $1 \leq S \leq 100$
    - $1 \leq N \leq 1000$
    - $1.00 \leq P \leq 1000.00$
    - $1 \leq h \leq 1000$



# Lab 5C: Buildings

- Specifications:
  - Implement using arrays/vectors. Program should iterate through the vector using the standard pointer dereference operator (\*). Using any *for loop* elements is NOT allowed. Program should contain functions that perform subtasks of your solution.
  - Filename: **Lab5C\_SURNAME1\_SURNAME2.cpp**
  - Input file: **yen.in**
  - Output: standard output

# Lab 5C: Buildings

Sample Input

2

6 3.00

8 2 3 11 11 10

5 12.00

12 20 39 45 89

1

Sample Output

6.00

60.00