North South University FINAL ID: _____
Summer, 2021 Total: 60 Date: September 18, 2021

The allocated points for each questions are provided on the left. The students are to **answer 3 the questions out of 4** worth 60 points. The submission method will be explained before the exam starts.

1. **(15 pts)** Write a program that reads a noun word (string) and then produces the following outputs::
   (a) What is the length of the word?
   (b) Form the plural of the noun by- adding "es" if that word ends with "o" , "s", "sh" or "ch". Otherwise simply add "s"; Print the following output.
   (c) Now read your full name as input and print your name in reverse order such that it follows "*Last_Name, First_Name*" format.

   Sample Input: *Noun:* volcano *Your Name:* George Orwell
   Sample Output:
   a) Length of the noun is 7
   b) Plural: volcanoes
   c) Orwell, George

2. **(20 pts)** Solve the following questions using array and pointers:
   (a) **(10 pts)** You are given an array of bulbs which are initially turned off.
   In the 1^{st} round you turn on all the bulb.
   In the 2^{nd} round you turn off every alternate bulb.
   In the 3^{rd} round you toggle (if lights are on you turn it off and vice versa) every third bulb .
   In i^{th} round you toggle the i^{th} bulb.
   Now print how many bulbs are turned on. Take the number of bulbs and number of rounds from the user.
   Sample Input: #of bulb= 7 #of rounds=4
   Sample Output: 5
   round01→[1 1 1 1 1 1 1]
   round02→[1 0 1 0 1 0 1]
   round03→[1 0 0 0 1 1 1]
   round04→[1 0 0 1 1 1 1]
   (b) **(10 pts)** Given an integer array of size n, find the longest sub-array where the absolute difference between any two consecutive elements is less than or equal to 1. Use *int abs(int x)* function from the *stdlib.h* to find the absolute value.
   Sample Input: [1 1 2 2 4 4 5 5 6]
   Sample Output: [4 4 5 5 6] is the longest sub-array with maximum size 5.

3. **(20 pts)** Suppose you are stuck in a maze of rooms and each room has one door. Now, you want to determine if there is an exit from the room that you are currently at or is it a infinite loop. You found a map that you decided to simplify using a 2D array (figure:1) and trace your options. To do so you jolted down the following steps:
   (a) **(10 pts)** Create a 2D array, *"maze"*, of $n \times m$ size. Populate it as shown in figure:1 and print it.
   (b) **(10 pts)** Take the users position and determine if you can get out of the maze or if you are stuck in a loop. If you can get out print the index number of that room.[Hint: Mark the rooms you already visited.]

   Sample Input 01: Starting point→ maze[0][0] **Sample Output 01:** You got out at maze[0][4]
   Sample Input 02: Starting point→ maze[4][0] **Sample Output 02:** You are stuck in a loop!!

1→	4↓	1→	1→	3↑	2←
4↓	2←	3↑	2←	1→	3↑
1→	1→	1→	3↑	3↑	2←
1→	1→	1→	1→	4↓	3↑
3↑	4↓	3↑	2←	2←	3↑

Figure 1: Maze [1 means Right, 2 meas Left, 3 means Up and 4 means Down]

4. **(25 pts)** Write a structure **Product** with the following attributes: *pName(string), quantity(integer), cost_price(float), selling_price(float).* Now answer the following questions:
   (a) **(5 pts)** Create an array of structure of arbitrary size and populate the array. All the information must be user inputs.
   (b) **(10 pts)** Write a function that will take *name* and *array of structure* as parameter and return the total profit that the shop will incur from that product. Here, **total profit= quantity** $\times$ **(selling_price - cost_price).**
   (c) **(10 pts)** Write a function that takes the structure array as input and returns a structure variable *bestProduct* that contains the product that incurs the most profit.