<u>IMPORTANT NOTE</u>: For all questions,first read the instructions written in here,then also check the instructions that appear in the source code files attached to this document. There is no need to write the whole code by yourselves, we gave you a little head start by writing those source code files - use them!

<u>assume valid input</u>

<u>instructions:</u>

In this exercise you will answer 4 questions.

In order to avoid excess files,the course team prepared a simple user interface where it will be possible to fill in the solution of the three questions in one file.

How the interface will work:

1. At the beginning of the running of the code,the question "Which question would you like to check?: " will appear.
2. The user (you in the phase of writing the assignment / we in the phase of checking the submission) will enter the number of the question he wants to run.
3. The message "Please enter a number: " will appear.
4. The user will enter the input he would like to check.
5. The user interface will enter the desired function and run it.
6. The output of the desired function will be printed to the screen for inspection by the user.

<u>Homework guidelines:</u>

1. To use the c file given to you,follow the next steps:
    a. Open new solutions in your computer
    b. Download the c files from the moodle
    c. Copy the content of each c file that has been given to you to each c file in your solution accordingly
2. Do not change the content of the print function given to you.
    a. we won't “spare” any printing mistakes.
3. Write your code only in the q_1/2/3/4 functions given to you in the c file located in the moodle according to the order below.
4. Do not delete the #define _CRT_SECURE_NO_WARNINGSin the c file.
5. Submit the exersize in its intended submission box in the right format as shown in class:
    a. <ex.num>_<ID.num>.c
    b. for example: ex3_123456789.c

information about the inputs:

- In this exercize you will deal with strings that represent a string of numbers.
- the input will be in the form of : “number1,number2,...”
- which means - every two numbers are separated with one comma (there is no comma in the end).
- a valid input for example: “3,4,7,-1,34”
- good luck.

Question 1

Given a string (arr[]),find the length of the subarray having a maximum sum.

Test cases:

arr[] = “1,-2,1,1,-2,1”: 2

Explanation: the Subarray with consecutive elements and maximum sum will be "1,1". So the length is 2.

arr[] ="-2,-3,4,-1,-2,1,5,-3 ": 5

Explanation : Subarray with consecutive elements and maximum sum will be "4,-1,-2,1,5". So the length is 5.

Question 2

Given a string (arr[]) of positive integers and a number k. Find the minimum number of swaps required to bring all the numbers less than or equal to k together.

Test cases:

arr[] = “2,1,5,6,3”,k = 3: 1

Explanation:

To bring elements 2,1,3 together,swap element ‘5’ with ‘3’ such that final array will be arr[] = "2,1,3,6,5".

arr[] = "2,7,9,5,8,7,4",k = 5: 2

Question 3

Given a string (arr[]) which contains elements from 0 to n-1 where n is the number of integers in the string (you can assume that there won't be any numbers larger or equal to n)

find if there's numbers occurring more than once in the given array, if so return 1,

else return -1.

Test cases:

arr[] ="2,3,1,2,3": 1

Explanation: 2 and 3 occur more than once in the given array.

arr[] = "0,3,1,2": -1

Question 4

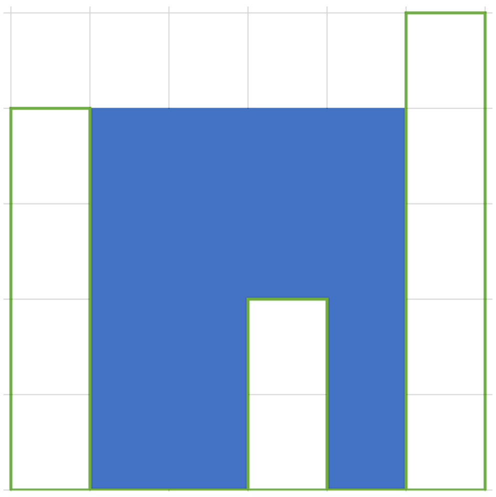
Given a string arr[] of non-negative integers representing the height of a tower made out of blocks. If the width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.

examples:

1) Input:arr[] = "4,0,0,2,0,5"

   Output:14

   Explanation:

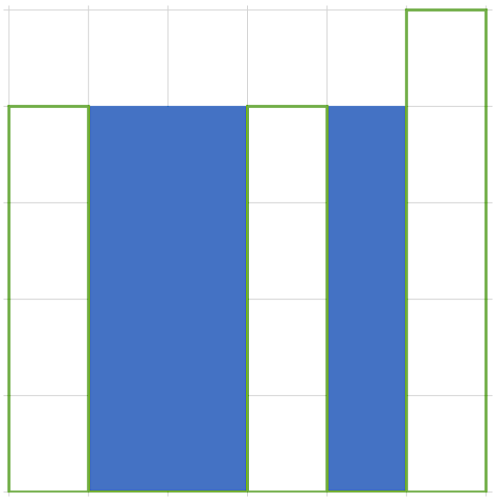


   total trapped water = 4+4+2+4 = 14

2) Input:arr[] = "4,0,0,4,0,5"

   Output:12

   Explanation:

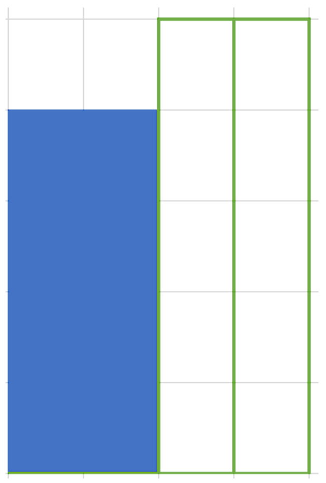


   total trapped water = 4+4+0+4 = 14

3) Input:arr[] = "0,0,5,5"

Output:0

Explanation:



No water will be trapped! (the water will fall)