You can view this report online at : https://www.hackerrank.com/x/tests/1687777/candidates/55569229/report

Full Name:	Breeha Qasim
Email:	bq08283@st.habib.edu.pk
Test Name:	**CS224 Lab# 03 - Fall 2023 V2**
Taken On:	6 Sep 2023 12:29:37 PKT
Time Taken:	3126 min 29 sec/ 7200 min
Work Experience:	< 1 years
Invited by:	Shafaq Fatima
Skills Score:	
Tags Score:	

100%

130/130

scored in **CS224 Lab# 03 - Fall 2023 V2** in 3126 min 29 sec on 6 Sep 2023 12:29:37 PKT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Swapping values > **Coding**	8 min 15 sec	30/ 30	✓
Q2	Points on a line? > **Coding**	13 hour 35 min 2 sec	70/ 70	✓
Q3	Bad Intern > **Coding**	21 min	30/ 30	✓

QUESTION 1

✓ Correct Answer

Score 30

Swapping values > Coding

QUESTION DESCRIPTION

Write a function that takes pointers to two integers and swaps the values of the integers.

CANDIDATE ANSWER

Language used: **C++**

```

void swap(int* a, int* b)
{
    int tempo = *a ;
    *a = *b;
    *b = tempo;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0313 sec	8.84 KB
Testcase 1	Easy	Hidden case	✓ Success	10	0.0211 sec	8.85 KB
Testcase 2	Easy	Hidden case	✓ Success	10	0.0506 sec	8.73 KB

No Comments

QUESTION 2

✓ Correct Answer

Score 70

Points on a line? › Coding

QUESTION DESCRIPTION

Given an array of points in a two dimensional space, find out if all of the points lie on a line?

e.g., both arrayOfPoints = [(0,1), (1,2), (3,4), (3.5,4.5)] and arrayOfPoints = [(-6,-1), (-5,-2), (-4,-3), (-2,-5), (-1,-6)] contains points that lie on a line.

Constraints: Assume that points are already sorted by increasing order of their x coordinate.

Hint: This has something to do with 1st derivative, or the difference between consecutive points of arrayOfPoints.

Sample Test Case:

```
4   // total number of points
0   // x coordinate of 1st point
1  // y coordinate of 1st point
1  // x coordinate of 2nd point
2   // y coordinate of 2nd point
3   // x coordinate of 3rd point
4   // y coordinate of 3rd point
3.5   // x coordinate of 4th point
4.5   // y coordinate of 4th point
```

INTERVIEWER GUIDELINES

```
bool isLine(vector<Point> arrayOfPoints) {
// use arrayOfPoints[i].x to access x coordinate of ith point
// use arrayOfPoints[i].y to access y coordinate of ith point
  int length = arrayOfPoints.size(); // length is the total number of points inside arrayOfPoints
  float dydx = (arrayOfPoints[1].y-arrayOfPoints[0].y)/(arrayOfPoints[1].x-arrayOfPoints[0].x);
  for (int i = 1; i < length; i++)
  {
    float newdydx =  (arrayOfPoints[i].y-arrayOfPoints[i-1].y)/(arrayOfPoints[i].x-arrayOfPoints[i-1].x);
    if (newdydx!=dydx)
      return false;
  }
  return true;
}
```

CANDIDATE ANSWER

Language used: **C++**

```

#include <cmath>
```

```
bool isLine(vector<Point> arrayOfPoints)
{
// use arrayOfPoints[i].x to access x coordinate of ith point
// use arrayOfPoints[i].y to access y coordinate of ith point
    int length = arrayOfPoints.size(); // length is the total number of
points inside arrayOfPoints
    float dy;
    float dx;
    float gradient , slope;
    dy = arrayOfPoints[1].y -  arrayOfPoints[0].y;
    dx = arrayOfPoints[1].x -  arrayOfPoints[0].x;
    gradient = dy / dx ;

    for (int i = 0 ; i < length - 1 ; i++)
    {
        dy = arrayOfPoints[i+1].y -  arrayOfPoints[i].y;
        dx = arrayOfPoints[i+1].x -  arrayOfPoints[i].x;
        slope = dy / dx;
        if (slope != gradient)
        {
            return false;
        }
    }
    return true;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0446 sec	8.75 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0566 sec	8.94 KB
Testcase 2	Medium	Hidden case	✓ Success	10	0.0461 sec	8.82 KB
Testcase 3	Hard	Hidden case	✓ Success	10	0.0457 sec	8.79 KB
Testcase 4	Hard	Hidden case	✓ Success	10	0.0305 sec	9.01 KB
Testcase 5	Hard	Hidden case	✓ Success	10	0.0409 sec	8.96 KB
Testcase 6	Easy	Hidden case	✓ Success	10	0.0366 sec	8.87 KB

No Comments

QUESTION 3

✓ Correct Answer

Score 30

Bad Intern › Coding

QUESTION DESCRIPTION

An intern at the bio lab you work at is not the most motivated and often makes errors. They are supposed to send you genome sequences which are the results of certain ongoing experiments. A correct genome sequence is a string consisting only of the letters *A C G T*. Yet, you have received sequences from your intern that contained other characters or lower case characters.

You want to write a program to correct the received sequence as follows. A lower case *a c g t* is converted to upper case. All other characters are omitted.

Function Description

Write a function that takes two strings as parameters. The first is the potentially bad sequence. The second is where you will save the corrected sequence.

▼ Input Format For Custom Testing

The input is a single line containing the potentially bad sequence.

▼ Sample Case 0

Sample Input For Custom Testing

```
AGCTAGCT
```

Sample Output

```
AGCTAGCT
```

Explanation

There was no error in the input sequence.

▼ Sample Case 1

Sample Input For Custom Testing

```
AgCTAGCTqAGTCCGA 3gATC
```

Sample Output

```
AGCTAGCTAGTCCGAGATC
```

Explanation

The input sequence has been corrected.

INTERVIEWER GUIDELINES

Solution

```
#include <iostream>

// Return the length of string.
int string_length(char string[]) {
  // The length is the number of characters in the string before '\0'.
  int i = 0;
  while (string[i] != '\0') {
    i++;
  }
  return i;
}

// Store a corrected copy of bad_sequence in good_sequence.
void fix_sequence(char bad_sequence[], char good_sequence[]) {
  int idx = 0;  // Index in good_sequence.
  bool valid_letter = true; // Is current letter in bad_sequence valid?
  for (int i = 0; bad_sequence[i] != '\0'; i++) {
    // Copy valid letters from bad_sequence to good_sequence.
    valid_letter = true;
    switch (bad_sequence[i]) {
    case 'a':
    case 'A':
      good_sequence[idx] = 'A';
      break;
    case 'g':
    case 'G':
      good_sequence[idx] = 'G';
      break;
    case 't':
    case 'T':
      good_sequence[idx] = 'T';
      break;
    case 'c':
    case 'C':
      good_sequence[idx] = 'C';
      break;
    default:
        valid_letter = false;
      }
      if (valid_letter) {
        idx++;
      }
    }
    // Terminate good_sequence.
    good_sequence[idx] = '\0';
  }

  int main(int argc, char** argv) {
    // Allocate sufficient space for the input sequence, take input.
    // length.
    char input_sequence[1000];
    std::cin.get(input_sequence, 1000);
    // Create a new string of the same length as the input sequence and
  store the
    // correct sequence in the new string.
    char correct_sequence[string_length(input_sequence)];
    fix_sequence(input_sequence, correct_sequence);
    std::cout << correct_sequence;
    return 0;
  }
```

CANDIDATE ANSWER

Language used: **C++**

```

/*
 * Complete the 'fix_sequence' function below.
 *
 * The function accepts following parameters:
 *  1. CHARACTER_ARRAY bad_sequence
 *  2. CHARACTER_ARRAY good_sequence
 */

void fix_sequence(char bad_sequence[], char good_sequence[]) {
    int ASCII;
    int counting=0;
    char null='\0';

    for (int i;i<1000;i++){
        if (bad_sequence[i]=='A' || bad_sequence[i]=='C' ||
bad_sequence[i]=='G' || bad_sequence[i]=='T'){
            good_sequence[counting]=bad_sequence[i];
            counting++;
        }
        else if (bad_sequence[i]=='a' || bad_sequence[i]=='c' ||
bad_sequence[i]=='g' || bad_sequence[i]=='t'){
            ASCII=int(bad_sequence[i])-32;
            good_sequence[counting]=char(ASCII);
            counting++;
        }
        //null_value
        else if (bad_sequence[i] == null){
            break;
        }
        else
        {
            counting=counting;
        }
    }
    good_sequence[counting]=null;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0233 sec	8.64 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0275 sec	8.78 KB
Testcase 2	Easy	Sample case	✓ Success	10	0.0237 sec	8.56 KB

No Comments