You can view this report online at : https://www.hackerrank.com/x/tests/1663581/candidates/55334253/report

Full Name:	Breeha Qasim
Email:	bq08283@st.habib.edu.pk
Test Name:	**CS 224 Lab# 02 - Fall 2023**
Taken On:	30 Aug 2023 12:20:47 PKT
Time Taken:	3188 min 6 sec/ 5760 min
Work Experience:	< 1 years
Invited by:	Asad
Skills Score:	
Tags Score:	C++ 120/120 switch-c++ 60/60

100%

250/250

scored in **CS 224 Lab# 02 - Fall 2023** in 3188 min 6 sec on 30 Aug 2023 12:20:47 PKT

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review it in detail here - https://www.hackerrank.com/x/tests/1663581/candidates/55334253/report

	Question Description	Time Taken	Score	Status
Q1	Simple Calculator - Using Switch > **Coding**	23 min 21 sec	60/ 60	✓
Q2	Library Information > **Coding**	50 min 46 sec	60/ 60	⚠
Q3	Chronological Human history > **Coding**	57 min 29 sec	60/ 60	⚠
Q4	Vector Operations > **Coding**	2 hour 5 min 30 sec	50/ 50	✓
Q5	Matrix Arithmetic > **Coding**	1 hour 44 min 19 sec	20/ 20	⚠

QUESTION 1

✓

Correct Answer

Score 60

Simple Calculator - Using Switch > Coding C++

QUESTION DESCRIPTION

You are to implement a simple calculator in C++ that gives out the result of addition, subtraction, multiplication and division of two integers depending on the operation that is asked to be performed.

For example, *int_one = 4, int_two = 5, op = '+'*. The result should then be *4 + 5 = 9*.

If an operator other than the four above is input then you should output the following: "Error! The operator is not correct"

Constraints

- All inputs are positive integers.

Note: You must solve this question using the switch statement in C++

▼ Input Format For Custom Testing

The first two lines contain the integers upon which you must perform the operation.

The third line contains a character denoting which operation to perform (+, -, *, /).

▼ Sample Case 0

Sample Input For Custom Testing

```
4
5
+
```

Sample Output

```
4 + 5 = 9
```

Explanation

4 + 5 = 9.

▼ Sample Case 1

Sample Input For Custom Testing

```
13
3
/
```

Sample Output

```
13 / 3 = 4
```

Explanation

13 / 3 = 4.333 = 4 (integer division in C++)

INTERVIEWER GUIDELINES

```
int main() {
    char op;
    int num1, num2;
    cin >> num1 >> num2;
    cin >> op;

    switch (op) {
        case '+':
            cout << num1 << " " << op << " " << num2 << " = " << num1 +
num2;
            break;
        case '-':
            cout << num1 << " " << op << " " << num2 << " = " << num1 -
num2;
            break;
        case '*':
            cout << num1 << " " << op << " " << num2 << " = " << num1 *
num2;
            break;
        case '/':
            cout << num1 << " " << op << " " << num2 << " = " << num1 /
num2;
```

```
            break;
        default:
            // operator is doesn't match any case constant (+, -, *, /)
            cout << "Error! The operator is not correct";
            break;
    }

    return 0;
}
```

CANDIDATE ANSWER

Language used: **C++**

```
#include <map>
#include <set>
#include <list>
#include <cmath>
#include <ctime>
#include <deque>
#include <queue>
#include <stack>
#include <string>
#include <bitset>
#include <cstdio>
#include <limits>
#include <vector>
#include <climits>
#include <cstring>
#include <cstdlib>
#include <fstream>
#include <numeric>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <unordered_map>

using namespace std;
int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int int_one;
    int int_two;
    char op;
    int result;
    cin >> int_one ;
    cin >> int_two;
    cin >> op;
    switch(op){
        case '+':
        result = int_one + int_two;
        cout << int_one << " " << "+" << " " << int_two << " = " << result;
        break;

        case '*':
        result = int_one*int_two;
        cout << int_one << " " << "*" << " " << int_two << " = " << result;
        break;

        case '/':
        result = int_one/int_two;
        cout << int_one << " " << "/" << " " << int_two << " = " << result;
        break;

        case '-':
        result=int_one-int_two;
        cout << int_one << " " << "-" << " " << int_two << " = " << result;
        break;

       default:
       cout << "Error! The operator is not correct";
       return 0;
    }
    return 0;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0564 sec	8.79 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0323 sec	8.79 KB
Testcase 2	Easy	Hidden case	✓ Success	10	0.056 sec	8.79 KB
Testcase 3	Easy	Hidden case	✓ Success	10	0.0258 sec	8.88 KB
Testcase 4	Easy	Hidden case	✓ Success	10	0.0209 sec	8.77 KB
Testcase 5	Easy	Hidden case	✓ Success	10	0.0266 sec	8.88 KB

No Comments

QUESTION 2

! Needs Review

Score 60

Library Information › Coding

C++ switch-c++

QUESTION DESCRIPTION

You are tasked with handling the borrowing of books at a library and want the borrowers to receive an informative text which would have the category, subcategory and the duration of borrow of the book being borrowed. A book can be issued for at most 15 days. No one can issue a book for more than 15 days.

For example, Jack is a fan of Murder Mystery books and always issues them from your library. So for the category he picks Novel (*category = 1*), for the subcategory he picks Murder Mystery (*subcategory = 1*), and he usually borrows the book for 4 days (*borrow_days = 4).* Therefore, the message that should be output for Jack is: "Issued you a novel in the subcategory of Murder Mystery for 4 days."

If *borrow_days > 15*, the following string must be output: "You can not issue a book for more than 15 days at our library!"

The current categorization and subcategorization of the items in your library is as follows:

- Category 1: Novels
  - Subcategory 1: Murder Mystery
  - Subcategory 2: Thriller
  - Subcategory 3: Science Fiction
- Category 2: Course Books
  - Subcategory 1: Mathematics
  - Subcategory 2: Physics
  - Subcategory 3: Biology

If a borrower wishes to not borrow from within a particular subcategory then they can input any number (that isn't already reserved for a subcategory) as the subcategory input. The resulting string will not contain a mention of the subcategory in this case. For example, if *category = 2, subcategory = 0 (which doesn't* *denote a subcategory at your library)* and *borrow_days = 10* then the resulting message output should be "Issued you a course book for 10 days."

Function Description

Write a function *book_information* that prints out the informative message for the book being borrowed. The function should take the following parameters:

- *category:* an integer
- *subcategory*: an integer
- *borrow_days:* an integer

Constraints

- All inputs are positive integers.

Note: You <u>must</u> solve this question using the switch statement in C++. Use of if...else is not allowed.

▼ Input Format For Custom Testing

There are three lines of input, each containing one integer each.

The first line contains the *category*, the second line the *subcategory* and the third line the *borrow_days*.

▼ Sample Case 0

Sample Input For Custom Testing

```
1
1
4
```

Sample Output

```
"Issued you a novel in the subcategory of Murder Mystery for 4 days."
```

Explanation

The category input (1) corresponds to Novels, the subcategory input (1) corresponds to Murder Mystery, and the borrow_days input (4) corresponds to the number of days the book is wished to be borrowed for.

▼ Sample Case 1

Sample Input For Custom Testing

```
2
0
10
```

Sample Output

```
"Issued you a course book for 10 days."
```

Explanation

The category input (2) corresponds to Course Books, the subcategory input (0) does not correspond to any predefined subcategory of novels at your library, and the borrow_days input (10) corresponds to the number of days the book is wished to be borrowed for.

INTERVIEWER GUIDELINES

```
void book_information(int category, int subcategory, int borrow_days) {
    switch(borrow_days > 15) {
        case 0:
            switch (category) {
                case 1:
                    cout << "Issued you a novel ";
                    switch (subcategory) {
                        case 1:
                            cout << "in the subcategory of Murder Mystery
for " << borrow_days << " days." << endl;
                            break;
                        case 2:
                            cout << "in the subcategory of Thriller for "
<< borrow_days << " days." << endl;
                            break;
                        case 3:
                            cout << "in the subcategory of Scince Fiction
for " << borrow_days << " days." << endl;
                            break;
                        default:
                            cout << "for " << borrow_days << " days." <<
endl;
                            break;
                    }
                    break;
                case 2:
                    cout << "Issued you a course book ";
                    switch (subcategory) {
                        case 1:
                            cout << "in the subcategory of Mathematics
for " << borrow_days << " days." << endl;
                            break;
                        case 2:
                            cout << "in the subcategory of Physics for "
<< borrow_days << " days." << endl;
                            break;
                        case 3:
                            cout << "in the subcategory of Biology for "
<< borrow_days << " days." << endl;
                            break;
                        default:
                            cout << "for " << borrow_days << " days." <<
endl;
                            break;
                    }
                    break;
            }
            break;
        case 1:
            cout << "You can not issue a book for more than 15 days at
our library!" << endl;
            break;
    }
}
```

CANDIDATE ANSWER

Language used: **C++**

```
/*
 * Complete the 'book_information' function below.
 *
 * The function accepts following parameters:
 *  1. INTEGER category
 *  2. INTEGER subcategory
 *  3. INTEGER borrow_days
 */

void book_information(int category, int subcategory, int borrow_days) {
    //borrow days should be less than 15 days
    switch(borrow_days>15){
        case true:
```

```cpp
        cout<<"You can not issue a book for more than 15 days at our library! ";
        break;
        case false:
        switch(category){
            case 1:
            switch (subcategory) {
                case 1:
                cout<<"Issued you a novel in the subcategory of Murder Mystery for "<< borrow_days<<" days.";
                break;
                case 2:
                cout<<"Issued you a novel in the subcategory of Thriller for "<< borrow_days<<" days.";
                break;
                case 3:
                cout<<"Issued you a novel in the subcategory of Science Fiction for "<< borrow_days<<" days.";
                break;
                default:
                cout<<"Issued you a novel for "<< borrow_days<<" days.";
        }
        break;
        case 2:
        switch(subcategory){
            case 1:
            cout<<"Issued you a course book in the subcategory of Mathematics for "<< borrow_days<<" days.";
            break;
            case 2:
            cout<<"Issued you a course book in the subcategory of Physics for "<< borrow_days<<" days.";
            break;
            case 3:
            cout<<"Issued you a course book in the subcategory of Biology for "<< borrow_days<<" days.";
            break;
            default:
            cout<<"Issued you a course book for "<< borrow_days<<" days.";
        }
        break;
    }
    break;
}
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0425 sec	8.91 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0236 sec	8.8 KB
Testcase 2	Easy	Sample case	✓ Success	10	0.0541 sec	8.86 KB
Testcase 3	Easy	Hidden case	✓ Success	10	0.0544 sec	8.82 KB
Testcase 4	Easy	Hidden case	✓ Success	10	0.0237 sec	8.76 KB
Testcase 5	Easy	Hidden case	✓ Success	10	0.0207 sec	8.68 KB

No Comments

QUESTION 3

Chronological Human history › Coding

Needs Review

Score 60

QUESTION DESCRIPTION

Historian AA wants to input all the important years in the human history, and chronologically order them from the beginning. The years will be saved in an integer array, and use Bubble Sort to sort the integer array

Function Description

You have to create the following functions:

1. ***input_years*** that takes two arguments, **yrs** - an array of integers that will store the years taken as input from the user, and **num_yrs** - an integer that represents the size of the array. The function does not returns any value. The function should take the year input from the user. If the entered year is less than 0 or greater than 9999, then it should print an error message stating **"Year can be between 0 and 9999! Try Again!"** on a separate line, and prompt for the input again. This should continue until a correct year value has not been entered. Once the correct year is entered, it should be added to the **yrs** array
2. ***print_yrs_array*** that takes two arguments, **yrs** - an array of integers containing the years inputted by the user, and **num_yrs** - an integer that represents the size of the array. The function does not returns any value. The function should print the list of years (**yrs**) separated by a comma, as specified in the Sample outputs
3. ***chronological_order*** that takes two arguments, **yrs** - an array of integers containing the years inputted by the user, and **num_yrs** - an integer that represents the size of the array. The function does not returns any value. This function will perform Bubble Sort to sort the array of years (**yrs**) in ascending order
4. ***main*** - The main function should first take an integer input from the user to represent the number of years to be entered, as well as to define the size of the array later. If the user enters a value less than 2, it should print an error message stating "**Need at least 2 years to sort! Try Again!**" on a separate line, and prompt for the input again. This should continue until a correct number has not been entered. Once the correct number is entered, it should call the function ***input_years*** to take the years value inputs from the user. After the array has been filled, it should print the initial array by calling the ***print_yrs_array*** function with a preceding message stating: "**The initial array is:**", as specified in the Sample outputs. Then call the ***chronological_order*** function to sort the array. Finally, print the sorted array by calling the ***print_yrs_array*** function again, with a preceding message stating: "**The sorted array is:**", as mentioned in the Sample Outputs

numbers[numbers[0],...numbers[n-1]]: an array of integers

Constraints

- ***yrs >= 0 and yrs <= 9999***
- ***num_yrs >= 2***

▼ Input Format For Custom Testing

The first line contains an integer, *num_years*, denoting the number of elements in the array, *years*.

Each line *i* of the *num_years* subsequent lines (where $0 \leq i < num\_years$) contains an integer describing $years_i$.

▼ Sample Case 0

Sample Input For Custom Testing

```
5
3500
1443
9965
7605
400
```

Sample Output

```
The initial array is: 3500, 1443, 9965, 7605, 400
The sorted array is: 400, 1443, 3500, 7605, 9965
```

Explanation

The user enters the number of years followed by the values of the years. It prints the unsorted array, sort it and finally print the sorted array.

▼ Sample Case 1

Sample Input For Custom Testing

```
-9
3
9999
2021
1980
```

Sample Output

```
Need at least 2 years to sort! Try Again!
The initial array is: 9999, 2021, 1980
The sorted array is: 1980, 2021, 9999
```

Explanation

The user enters incorrect number of years, so the program prints the error message and prompts to enter again. Then the correct number of years is entered followed by the year values. It prints the unsorted array, sort it and finally print the sorted array.

▼ Sample Case 2

Sample Input For Custom Testing

```
3
10000
9999
2021
1980
```

Sample Output

```
Year can be between 0 and 9999! Try Again!
The initial array is: 9999, 2021, 1980
The sorted array is: 1980, 2021, 9999
```

Explanation

The user enters the number of years, and then enters the wrong year value, so the program prints the error message and prompts to enter again. Then the correct values are entered. It prints the unsorted array, sort it and finally print the sorted array.

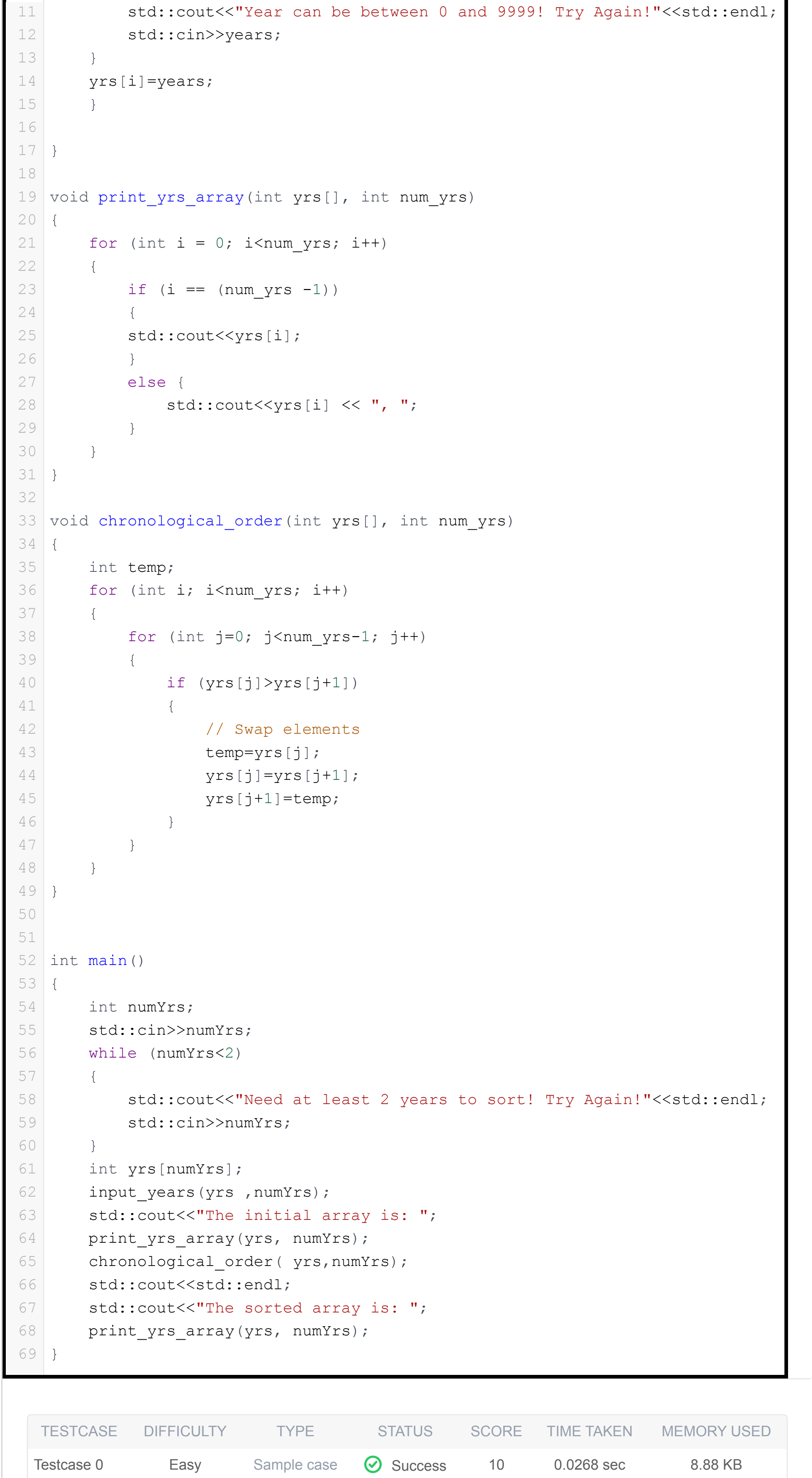
INTERVIEWER GUIDELINES

```
#include<iostream>
using namespace std;

void input_years(int yrs[], int size)
{
    int yr;
    for (int i = 0; i < size; i++)
    {
        cin >> yr;

        while (yr < 0 || yr > 9999)
        {
            cout << "Year can be between 0 and 9999! Try Again!\n";
            cin >> yr;
        }

        yrs[i] = yr;
    }
}

void print_yrs_arr(int yrs[], int size)
{
    for (int i = 0; i < size - 1; i++)
    {
        cout << yrs[i] << ", ";
    }

    cout << yrs[size - 1] << "\n";
}
void chronological_order (int yrs[], int size)
{
    for (int i = 0; i < size; i++)
    {
        for (int j = i+1; j < size; j++)
        {
            int temp;
            if (yrs[i] > yrs[j])
            {
                temp = yrs[i];
                yrs[i] = yrs[j];
                yrs[j] = temp;
            }
        }
    }
}

int main()
{
    int num_years;
    cin >> num_years;

    while(num_years < 2)
    {
        if(num_years < 2)
        {
            cout << "Need at least 2 years to sort! Try Again!\n";
            cin >> num_years;
        }
    }

    int years[num_years];

    input_years(years, num_years);

    cout << "The initial array is: ";
    print_yrs_arr(years, num_years);

    chronological_order(years, num_years);
    cout << "The sorted array is: ";
    print_yrs_arr(years, num_years);
}
```

CANDIDATE ANSWER

Language used: **C++**

```

// Function to input years
void input_years(int yrs[], int num_yrs)
{
    for (int i=0; i<num_yrs; i++)
    {
        int years;
        std::cin>>years;
        while (years<0 || years>9999)
    {
```



```
        std::cout<<"Year can be between 0 and 9999! Try Again!"<<std::endl;
        std::cin>>years;
    }
    yrs[i]=years;
    }

}

void print_yrs_array(int yrs[], int num_yrs)
{
    for (int i = 0; i<num_yrs; i++)
    {
        if (i == (num_yrs -1))
        {
        std::cout<<yrs[i];
        }
        else {
            std::cout<<yrs[i] << ", ";
        }
    }
}

void chronological_order(int yrs[], int num_yrs)
{
    int temp;
    for (int i; i<num_yrs; i++)
    {
        for (int j=0; j<num_yrs-1; j++)
        {
            if (yrs[j]>yrs[j+1])
            {
                // Swap elements
                temp=yrs[j];
                yrs[j]=yrs[j+1];
                yrs[j+1]=temp;
            }
        }
    }
}


int main()
{
    int numYrs;
    std::cin>>numYrs;
    while (numYrs<2)
    {
        std::cout<<"Need at least 2 years to sort! Try Again!"<<std::endl;
        std::cin>>numYrs;
    }
    int yrs[numYrs];
    input_years(yrs ,numYrs);
    std::cout<<"The initial array is: ";
    print_yrs_array(yrs, numYrs);
    chronological_order( yrs,numYrs);
    std::cout<<std::endl;
    std::cout<<"The sorted array is: ";
    print_yrs_array(yrs, numYrs);
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0268 sec	8.88 KB

Testcase 1	Medium	Sample case	✓ Success	10	0.0347 sec	8.85 KB
Testcase 2	Medium	Sample case	✓ Success	10	0.0431 sec	8.8 KB
Testcase 3	Easy	Hidden case	✓ Success	10	0.0262 sec	8.8 KB
Testcase 4	Medium	Hidden case	✓ Success	10	0.027 sec	8.82 KB
Testcase 5	Hard	Hidden case	✓ Success	10	0.0902 sec	8.94 KB

No Comments

QUESTION 4

✓

Correct Answer

Score 50

Vector Operations › Coding

QUESTION DESCRIPTION

We are going to write a program to perform arithmetic on *vectors*, specifically to add and subtract them. We will store a vector using an integer array.

Function Description

You have to write the following functions.

- *print_vector*: prints the contents of the vector passed to it.
- *input_vector*: populates the vector passed to it with values input form the console.
- *add_vectors*: takes 3 vectors as parameters and stores the sum of the first 2 vectors in the third one
- *subtract_vectors*: takes 3 vectors as parameters and stores the difference of the first 2 vectors in the third one
- *compare_vectors*: takes 2 vectors as parameters and returns *true* if they are equal and *false* otherwise.

You may pass any additional parameters to your functions as needed.

▼ Input Format For Custom Testing

The first line contains an integer, *n*, denoting the dimension of the 2 vectors that follow.

The next line contains *n* space separated integers which represent the first vector.

The next line contains *n* space separated integers which represent the second vector.

The next line contains a single character which is one of: + - =. Depending on the character, you have to output the sum or the difference of the vectors or if the vectors are equal.

▼ Sample Case 0

Sample Input For Custom Testing

```
3
6 5 4
1 2 3
+
```

Sample Output

```
7 7 7
```

Explanation

The sum of the vectors [6 5 4]^T and [1 2 3]^T is [7 7 7]^T.

▼ Sample Case 1

Sample Input For Custom Testing

```
3
6 5 4
1 2 3
-
```

Sample Output

```
5 3 1
```

Explanation

The difference of the vectors $[6\ 5\ 4]^T$ and $[1\ 2\ 3]^T$ is $[5\ 3\ 1]^T$.

INTERVIEWER GUIDELINES

Solution

```
#include <iostream>

// Print vec of size.
void print_vector(int vec[], int size) {
  for (int i = 0; i < size; i++) {
    std::cout << vec[i] << " ";
  }
}

// Input vec of size from console.
void input_vector(int vec[], int size) {
  for (int i = 0; i < size; i++) {
    std::cin >> vec[i];
  }
}

// Store the sum of vec1 and vec2 in result. All are of size.
void add_vectors(int vec1[], int vec2[], int result[], int size) {
  for (int i = 0; i < size; i++) {
    result[i] = vec1[i] + vec2[i];
  }
}

// Store the difference of vec1 and vec2 in result. All are of size.
void subtract_vectors(int vec1[], int vec2[], int result[], int size) {
  for (int i = 0; i < size; i++) {
    result[i] = vec1[i] - vec2[i];
  }
}

// Compute if vec1 and vec2 of size are equal.
bool compare_vectors(int vec1[], int vec2[], int size) {
  for (int i = 0; i < size; i++) {
    if (vec1[i] != vec2[i]) {
      return false;
    }
  }
  return true;
}

int main() {
  // Input size and then create and input vectors of that size.
  int n;
  std::cin >> n;
  int vec1[n], vec2[n];
  input_vector(vec1, n);
  input_vector(vec2, n);
  // Input and perform operation.
  char op;
  std::cin >> op;
  if (op == '+') {
    // Addition.
    int result[n];
    add_vectors(vec1, vec2, result, n);
    print_vector(result, n);
  }
  else if (op == '-') {
    // Subtraction.
    int result[n];
    subtract_vectors(vec1, vec2, result, n);
    print_vector(result, n);
  }
  else if (op == '=') {
    // Comparison.
    if (compare_vectors(vec1, vec2, n)) {
      std::cout << "EQUAL";
    }
    else {
      std::cout << "UNEQUAL";
    }
  }
  return 0;
}
```

CANDIDATE ANSWER

Language used: **C++14**

```
#include <iostream>

void print_vector(int vector[], int size) {
    for (int i = 0; i < size; i++) {
        std::cout<<vector[i]<<' ';
    }
    std::cout << std::endl;
}

void input_vector(int vector[], int size) {
    for (int i = 0; i < size; i++) {
        std::cin >> vector[i];
    }
}

void add_vectors(int vector1[], int vector2[], int result[], int size) {
    for (int i = 0; i < size; i++) {
        result[i] = vector1[i] + vector2[i];
    }
}

void subtract_vectors(int vector1[], int vector2[], int result[], int size) {
    for (int i = 0; i < size; i++) {
        result[i] = vector1[i] - vector2[i];
    }
}

bool compare_vectors(int vector1[], int vector2[], int size) {
    int count;
    for (int i=0; i<size; i++)
    {
        if (vector1[i]==vector2[i])
        {
            count=count+1;
        }

    }
    if (count==size)
    {
        return true;
    }
    else
    {
        return false;
    }
}

int main() {
    int size;
    std::cin>>size;

    int vector1[size];
    int vector2[size];
    int result[size];

    input_vector(vector1, size );
    input_vector(vector2, size );

    int comp = compare_vectors( vector1, vector2, size);

    char op;
    std::cin>>op;

    switch(op)
    {
        case '+':
        add_vectors( vector1, vector2, result, size);
        print_vector(result,size);
        break;

        case '-':
        subtract_vectors( vector1, vector2,  result, size);
        print_vector(result,size);
        break;

        case '=':
        switch(comp)
        {
            case true:
            std::cout<<"EQUAL"<<std::endl;
            break;

            case false:
            std::cout<<"UNEQUAL"<<std::endl;
            break;
        }
        break;
    }
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0395 sec	8.71 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0909 sec	8.78 KB
Testcase 2	Easy	Sample case	✓ Success	10	0.0353 sec	8.9 KB
Testcase 3	Easy	Sample case	✓ Success	10	0.0551 sec	8.9 KB
Testcase 4	Easy	Sample case	✓ Success	10	0.0271 sec	8.78 KB

No Comments

QUESTION 5

Matrix Arithmetic › Coding

Needs Review

Score 20

QUESTION DESCRIPTION

We are going to write a program to perform arithmetic on *matrices*, specifically to add and subtract them. While a 2D array is a more suitable representation for a matrix, there are peculiarities with passing a 2D array to a function. We will therefore simulate the 2D structure of the matrix with a regular (1D) array.

Function Description

You have to write the following functions.

- *print_matrix*: prints the contents of the matrix passed to it.
- *input_matirx*: populates the matrix passed to it with values input form the console.
- *add_matrices*: takes 3 matrices as parameters and stores the sum of the first 2 matrices in the third one
- *subtract_matrices*: takes 3 matrices as parameters and stores the difference of the first 2 matrices in the third one

You may pass any additional parameters to your functions as needed.

▼ Input Format For Custom Testing

The first line contains 2 integers, *r* and *c*, denoting the dimension of the 2 matrices that follow. Both matrices will have *r* rows and *c* columns.

Each of the next *r* lines contains *c* space separated integers corresponding to a row of the first matrix.

Each of the next *r* lines contains *c* space separated integers corresponding to a row of the second matrix.

The next line contains a single character, either + or -. Depending on the character, you have to output the sum or the difference respectively of the matrices.

▼ Sample Case 0

Sample Input For Custom Testing

```
2 3
1 0 3
5 -1 2
6 12 9
14 1 0
+
```

Sample Output

```
7 12 12
19 0 2
```

Explanation

The output is the sum of the 2 matrices.

▼ Sample Case 1

Sample Input For Custom Testing

```
2 3
1 0 3
5 -1 2
6 12 9
14 1 0
-
```

Sample Output

```
-5 -12 -6
-9 -2 2
```

Explanation

The output is the difference of the 2 matrices.

Solution

Solution

INTERVIEWER GUIDELINES

```
// Print matrix of size: rows and cols.
void print_matrix(int matrix[], int rows, int cols) {
  for (int r = 0; r < rows; r++) {
    for (int c = 0; c < cols; c++) {
      std::cout << matrix[r*cols + c] << " ";
    }
    std::cout << "\n";
  }
}

// Input matrix of size: rows and cols.
void input_matrix(int matrix[], int rows, int cols) {
  for (int i = 0; i < rows*cols; i++) {
    std::cin >> matrix[i];
  }
}

// Store the sum of matrix1 and matrix2 in result. All are of size: rows
and
// cols.
void add_matrices(int matrix1[], int matrix2[], int result[], int rows,
     int cols) {
  for (int i = 0; i < rows*cols; i++) {
    result[i] = matrix1[i] + matrix2[i];
  }
}

// Store the difference of matrix1 and matrix2 in result. All are of
size: rows
// and cols.
void subtract_matrices(int matrix1[], int matrix2[], int result[], int
rows,
          int cols) {
  for (int i = 0; i < rows*cols; i++) {
    result[i] = matrix1[i] - matrix2[i];
  }
}

int main() {
  // Input size and then create and input matrices of that size.
  int r, c;
  std::cin >> r >> c;
  int matrix1[r*c], matrix2[r*c];
  input_matrix(matrix1, r, c);
  input_matrix(matrix2, r, c);
  // Input and perform operation.
  char op;
  std::cin >> op;
  int result[r*c];
  if (op == '+') {
    // Addition.
    add_matrices(matrix1, matrix2, result, r, c);
  }
  else if (op == '-') {
    // Subtraction.
    subtract_matrices(matrix1, matrix2, result, r, c);
  }
  print_matrix(result, r, c);
  return 0;
}
```

CANDIDATE ANSWER

Language used: **C++14**

```cpp
#include <iostream>

void print_matrix(int matrix[][100], int rows , int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            std::cout << matrix[i][j]<<' ';
        }
        std::cout << std::endl;
    }
}

void input_matrix(int matrix[][100], int rows , int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            std::cin >> matrix[i][j];
        }
    }
}

void add_matrices(int matrix1[][100],int matrix2[0][100], int result[0]
[100],int rows,int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}

void subtract_matrices(int matrix1[][100],int matrix2[0][100], int result[0]
[100],int rows,int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] - matrix2[i][j];
        }
    }
}

int main(){
    int rows,cols;
    std::cin>>rows;
    std::cin>>cols;

    int matrix1[100][100];
    int matrix2[100][100];
    int result[100][100];

    input_matrix(matrix1 , rows , cols);
    input_matrix(matrix2 , rows , cols);

    char op;
    std::cin>>op;

    switch(op)
    {
        case '+':
        add_matrices(matrix1, matrix2, result, rows ,cols);
        break;

        case '-':
        subtract_matrices(matrix1, matrix2, result, rows,cols);
        break;
    }
    print_matrix(result,  rows , cols);
    return 0;
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✓ Success	10	0.0386 sec	8.88 KB
Testcase 1	Easy	Sample case	✓ Success	10	0.0502 sec	9 KB

No Comments

PDF generated at: 4 Sep 2023 18:25:24 UTC