

Homework 2: Arrays and Structs

Due: Tuesday, Feb 4, 11:55pm

How to turn in:

- Write three programs and submit them on iLearn
- If you use repl.it, the files will be named main.cpp; you will need to rename them
- The names are in the assignment; incorrect names will be graded as 0

Part 1. Array Operations (10 points)

Write a program that conducts three operations to an array. For the problem, your program should have an array with 10 positive integer values such as 10, 20, 30, 40, 50, 15, 25, 35, 45, and 55. Then, your program should ask for an option from three choices; (1) delete the max, (2) append a number, and (3) reverse array. This program should be submitted as hw2-1.cpp.

A sample run of your program should look like this:

```
Array values: 10 20 30 40 50 15 25 35 45 55
```

```
This is a list of operations
```

1. Delete the max
2. Append a number
3. Reverse array

```
Enter your option: 3
```

```
New array values: 55 45 35 25 15 50 40 30 20 10
```

```
Continue? (Y/N): Y
```

```
Enter your option: 2
```

```
Array is full and can't append anymore.
```

```
Array values: 55 45 35 25 15 50 40 30 20 10
```

```
Continue? (Y/N): Y
```

```
Enter your option: 1
```

```
New array values: 45 35 25 15 50 40 30 20 10 0
```

```
Continue? (Y/N): Y
```

```
Enter your option: 1
```

```
New array values: 45 35 25 15 40 30 20 10 0 0
```

```
Continue? (Y/N): Y
```

```
Enter your option: 1
```

```
New array values: 35 25 15 40 30 20 10 0 0 0
```

```
Continue? (Y/N): Y
```

```
Enter your option: 1
```

```
New array values: 35 25 15 30 20 10 0 0 0 0
```

```
Continue? (Y/N): Y
```

```
Enter your option: 3
New array values: 10 20 30 15 25 35 0 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 1
New array values: 10 20 30 15 25 0 0 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 2
Enter the number to append: 30
New array values: 10 20 30 15 25 30 0 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 1
New array values: 10 20 15 25 30 0 0 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 2
Enter the number to append: 27
New array values: 10 20 15 25 30 27 0 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 2
Enter the number to append: 12
New array values: 10 20 15 25 30 27 12 0 0 0
Continue? (Y/N): Y
```

```
Enter your option: 3
New array values: 12 27 30 25 15 20 10 0 0 0
Continue? (Y/N): N
```

Read the sample run carefully to understand the requirements of the problem correctly. Especially, you should know which number you should remove if you have duplicated max numbers in the array.

Part 2. Quiz Averages with Structs

Develop a program to determine the average of quiz scores of all students in a course. The input data of the course will be stored in an input file, and a record of each student will include the following

- The student's name
- The student's ID
- Five quiz scores
- An average of the students' quiz scores

The file will have up to (but no more than) 30 records, and will have a special string (STOP) to indicate the end of file. An example file is available on iLearn, but you should make your own test files, too.

You must use a structure (aka struct) to store each student's record. Since there are several students in a course, you have to use an array of structure to keep all records. This program should be submitted as hw2-2.cpp.

Your program should:

- Ask for an input filename and read the input data
- Calculate the average of the top four quiz scores for each person (discard the lowest score)
- Print the results for each student to the screen, per the format below

Student (ID#): Average

Sample Input File (sample_t1.txt):

```
Tom    1000  9.5  9.0  8.5  8.0  8.5
Alice  2000 10.0  6.7 10.0 10.0 10.0
John   3000  6.9  8.0  8.0  8.0  8.0
Alice  1500  8.0  9.0  7.0  6.5  8.0
Jason  2500  5.0  5.0  5.0  3.5  5.0
STOP   -1    -1.0 -1.0 -1.0 -1.0 -1.0
```

Sample Run (user input in **bold**):

```
Enter an input file: sample_t1.txt
```

```
-----
Course Report: Quiz Average
-----
```

```
Tom (1000): 8.875
Alice (2000): 10
John (3000): 8
Alice (1500): 8
Jason (2500): 5
-----
```

Part 3. Struct Search (10 points)

Write a program to find the customer's information from a bank data. For the problem, the input data will be stored in an input file, and a record of each customer will include the customer's name, account number, and current balance. You must store each customer's record in a struct. Since there are several customers in a bank, you have to use an array of structs to keep all records. This program should be submitted as hw2-3.cpp.

The file will have up to (but no more than) 30 records, and will have a special string (STOP) to indicate the end of file. An example file is available on iLearn, but you should make your own test files, too.

Your program should:

- Ask for an input filename and read the input data
- Present the user with a record-finding function
 - Let the user enter a name

- If the name is found once, present all of the customer information
- If the name is found multiple times, present all of the customer information
- If the name is not found, present an error message to the user
- After a search, allow the user to choose to continue (search again) or not (exit)

Sample Input File (sample_t1.txt):

```
Tom      1000   10.55
Alice    2000   200.75
John     3000   69.57
Alice    1500    8.0
Jason    2500  555.55
STOP     -1     -1.0
```

Sample Run (user input in **bold**):

```
Enter an input file: sample_t2.txt
-----
Record Finder - Enter a customer name: Alice
-----
Name: Alice
Account: 2000
Balance: 200.75

Name: Alice
Account: 1500
Balance: 8.0
-----

Do you want to continue? (y/n) y

-----
Record Finder - Enter a customer name: Johnson
-----
Fail. Johnson doesn't exist.
-----

Do you want to continue? (y/n) y

-----
Record Finder - Enter a customer name: Jason
-----
Name: Jason
Account: 2500
Balance: 555.55
-----

Do you want to continue? (y/n) n

BYE!
```