

# COP 2535: Data Structures

## Exercise 05, ListPlay

### 1 Instructions

This programming exercise has four parts consisting of four requirements. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run.

For this programming exercise, you have three test arrays of type integer and two strings (arrays of type character), shown below. Write a program that performs the following tasks, listed below. **Do not** use convenience functions. (You may use the built in length functions. Write the code by hand.

**Array A:** 0, 2, 4, 6, 8, 10

**Array B:** 1, 3, 5, 7, 9

**Array C:** 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 9

**String D:** "abcdefghij"

**String E:** "Your Name" (whatever your real name is, for example, John Jones or Darth Vader)

**Counting, summing, computing the mean: 70 points** Write a method counts the number of elements in an integer array, and then sums the elements in an integer array. It should return the average of the array elements as a double. Using the count and sum, compute and print the mean of each array.

**Reversing arrays: 80 points** Write a method that accepts an array as an argument and prints the reversed array. For example, if you pass the method Array B, it should print [9, 7, 5, 3, 1]. Print the reverse of all five arrays.

**Rotating arrays: 90 points** Arrays can be rotated to the right or to the left by any number of places. Rotating a array to the right by two places puts the first element in position three, the second element in position four, and so on, with the last element in position two and the next to last element in position one. Array A rotated to the right by two places results in [8, 10, 0, 2, 4, 6]. Likewise, rotating an array to the left places the first elements at the end. Array B rotated to the left by two places results in [5, 7, 9, 1, 3].

Write a method that accepts three parameters, a direction (right or left), a number of places, and an array, which prints the appropriate rotation. Call the method on all three arrays. Rotate array A two places to the left. Rotate array B two places to the right. Rotate array C four places to the left.

**Sorting arrays: 100 points** Write a method that takes an unsorted integer array and prints a sorted array. Use Array C as your test array. Do you recognize this list of numbers?

### 2 Template

Here s a template you can use. I have included all your function prototypes and function calls. You are responsible for defining the functions.

```

/*****
    Name:
    Author:
    Date:
    Purpose:
    Input:
    Output:
*****/

#include <iostream>
#include <string>
#include <cctype>

using namespace std;

void printIntList(const int [], const int);
void printCharList(const string, const int);
void Average(const int[], const int);
void printReverseIntList(const int[], const int);
void printReverseCharList(const string, const int);
void printRotatedIntList(const int[], const int, char, int);
void printRotatedCharList(const string, const int, char, int);
void printSortedIntArray(int [], const int);
void printSortedCharArray(string, const int);

int len;

int main()
{
    int x1[] = { 0, 2, 4, 6, 8, 10 };
    int 2[] = { 1, 3, 5, 7, 9 };
    int 3[] = { 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5 };
    string x4 = "abcdefghij";
    string x5 = "Your Name";
    int* p;

    cout << "-----" << endl;
    cout << "   Playing with Arrays   " << endl;
    cout << "-----" << endl;

    /*****/
    cout << "\nprinting arrays forward\n\n";

    len = sizeof(x1) / sizeof(int);
    cout << "printing array x1" << endl;
    printIntList(x1, len);
    Average(x1, len);

    len = sizeof(x2) / sizeof(int);
    cout << "printing array x2" << endl;
    printIntList(x2, len);
    Average(x2, len);

```

```

len = sizeof(x3) / sizeof(int);
cout << "printing array x3" << endl;
printIntList(x3, len);
Average(x3, len);

len = x4.length();
cout << "String x4 is:" << endl;
printCharList(x4, len);

len = x5.length();
cout << "String x5 is:" << endl;
printCharList(x5, len);

/*****/
cout << "\nprinting arrays backward\n\n";

len = sizeof(x1) / sizeof(int);
cout << "reversing array x1" << endl;
printReverseIntList(x1, len);

len = sizeof(x2) / sizeof(int);
cout << "reversing array x2" << endl;
printReverseIntList(x2, len);

len = sizeof(x3) / sizeof(int);
cout << "reversing array x3" << endl;
printReverseIntList(x3, len);

len = x4.length();
cout << "reversing String x4 is:" << endl;
printReverseCharList(x4, len);

len = x5.length();
cout << "reversing String x5 is:" << endl;
printReverseCharList(x5, len);

/*****/
cout << "\nprinting rotated arrays\n\n";

len = sizeof(x1) / sizeof(int);
cout << "rotating array x1 2 places to the left" << endl;
printRotatedIntList(x1, len, 'L', 2);

len = sizeof(x2) / sizeof(int);
cout << "rotating array x2 2 places to the right" << endl;
printRotatedIntList(x2, len, 'R', 2);

len = sizeof(x3) / sizeof(int);
cout << "rotating array x3 4 places to the left" << endl;
printRotatedIntList(x3, len, 'L', 4);

len = x4.length();
cout << "rotating string x4 3 places to the right" << endl;
printRotatedCharList(x4, len, 'R', 3);

```

```

    len = x5.length();
    cout << "rotating string x5 1 places to the left" << endl;
    printRotatedCharList(x5, len, 'L', 1);

    cout << "\nprinting rotated arrays\n\n";

    len = sizeof(x3) / sizeof(int);
    cout << "sorting array x3" << endl;
    printSortedIntArray(x3, len);

    len = x5.length();
    cout << "sorting string" << endl;
    printSortedCharArray(x5, len);

    return 0;
}

//your function definitions here

```

### 3 Output

Your deliverable is a text file that will look similar to this. You should be able to select your output screen, copy it (with Ctl-C), and insert it into the text entry box (Ctl-V).

```

-----
    Playing with Arrays
-----

printing arrays forward

printing array x1
0 2 4 6 8 10
Average is 5, which is 30 divided by 6
printing array x2
1 3 5 7 9
Average is 5, which is 25 divided by 5
printing array x3
3 1 4 1 5 9 2 6 5 3 5
Average is 4, which is 44 divided by 11
String x4 is:
a b c d e f g h i j
String x5 is:
Y o u r   N a m e

printing arrays backward

reversing array x1
10 8 6 4 2 0
reversing array x2
9 7 5 3 1
reversing array x3
5 3 5 6 2 9 5 1 4 1 3
reversing String x4 is:

```

```
j i h g f e d c b a
reversing String x5 is:
e m a N   r u o Y
```

```
printing rotated arrays
```

```
rotating array x1 2 places to the left
4 6 8 10 0 2
rotating array x2 2 places to the right
7 9 1 3 5
rotating array x3 4 places to the left
5 9 2 6 5 3 5 3 1 4 1
rotating string x4 3 places to the right
h i j a b c d e f g
rotating string x5 1 places to the left
o u r   N a m e Y
```

```
printing rotated arrays
```

```
sorting array x3
sorted int list
1 1 2 3 3 4 5 5 5 6 9
sorting string
sorted string
  N Y a e m o r u
```