Object Oriented Programming
Home Work 03

Home Work 03 **Marks 10**

Instructions

Work on this home work individually. Absolutely NO collaboration is allowed. Any traces of plagiarism would result in a ZERO marks in this homework and possible disciplinary action. All files should have the same name as of the tasks and coded in **C++**.

Due Date

Paste the solution(s) folder of the problems (source code .cpp files only) labeled with your complete roll number in SEM – HW 03 and SEA – HW 03 folders for SE Morning and SE Afternoon sections respectively on Thursday, March 17, 2016 before 05:00 PM. These folders are available at \\printsrv\Teacher Data\Umair Babar\Students.

Task 01 [02]

Write and test the function that attempts **to remove an item** from an array:

```
bool removeFirst(float a[], int &size, float key);
```

The function searches the array **a** for the item **key**. If **key** is found, its first occurrence is removed; all the elements above that position are shifted down, **size** is decremented, and **true** is returned to indicate a successful removal. If **key** is not found, the array is left unchanged and **false** is returned.

Task 02 [02]

Write and test the following function:

```
void rotate(int a[], int size, int positions);
```

The function **rotates** the elements of the array **a**, number of **positions** to the right (or number of positions to the left if **k** is negative). The last **positions** elements are **wrapped** around to the beginning of the array.

For example,

1. The call **`rotate(a,8,3)`** would transform the array

   `{22,33,44,55,66,77,88,99}` into `{77,88,99,22,33,44,55,66}`

2. The call **`rotate(a,8,-4)`** would transform the array

   `{22,33,44,55,66,77,88,99}` into `{66,77,88,99,22,33,44,55}`

Task 03 [02]

Write a program that implement the following functions:

createArray – accepts an **integer array** and the **array's size** as arguments. The function should **create a copy of the array**, except that the **element values should be reversed** in the copy. The function should **return a pointer** to the **new array**.

showData – accept a **pointer to array** with its **size** and **display** all the **contents** of the array.

Task 04 [02]

Write and test the function that attempts to add the contents of two equal size arrays:

```
float * add(const float a[], const float b[], int sizeofa, int sizeofb);
```

The function adds the adjacent elements of two array **a** and **b** and places there sum to the newly created array and return it if possible, return **NULL** otherwise. The addition is only possible if the sizes of both the arrays are equal.

For example,

1. **a** contains `{1, 2, 3, 4, 5}` and array **b** contains `{5, 4, 3, 2, 1}` then the resultant array is `{6, 6, 6, 6, 6}`
2. **a** contains `{1, 2, 3, 4, 5}` and array **b** contains `{5, 4, 3, 2}` then the addition is not possible so return **NULL**

NOTE: - No submission will be accepted after the due date and time.

Best of Luck