

Name and, if possible, ID#: _____

AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
COMPI20 Introduction to Object-Oriented Programming
MIDTERM 2 EXAM

Date: Tuesday, March 24 2015

Starting time: 10:30

Duration: 1 hour 20 minutes

Attention: **ANY COMMUNICATION IS STRICTLY PROHIBITED**

Please write down your name at the top of all used pages

Problem 1

The easiest way to implement rotation by 90° of a square array is to transpose it and then reverse all its rows separately. Write a C++ function `void rotate(int *a2D, int size)` that takes as its argument a pointer to the first element of a square array `int *a2D` of the specified `int size` and rotates its. Use already implemented functions `void reverse(int a1D[], int length)` and `void transpose(int *a2D, int size)`:

```
void reverse(int a1D[], int length)
{
    for (int i = 0; i < length / 2; i++)
        swap(a1D[i], a1D[length - 1 - i]);
}

void transpose(int *a2D, int size)
{
    for (int row = 0; row < size; row++)
        for (int col = row + 1; col < size; col++)
            swap(a2D[row * size + col], a2D[col * size + row]);
}
```

```
void rotate(int *a2D, int size) {
    transpose(a2D, size);
    for (int j = 0; j < size; j++)
        reverse(a2D[j], size);
}
```

3

Name and, if possible, ID#: _____

Problem 3

Using, if you wish, `segment()` and `rotate()` functions from the C++ Reference Functions section, write a C++ function `void spiral2(int *a2D, int even_size)` that takes as its argument a pointer to the first element of a square array `int *a2D` of the specified even size `int even_size` and fills its top-left and bottom-right quadrants with spirals of successive values from 1 to $\text{even_size}^2/4$. The remaining two quadrants are filled with zeros. Each spiral propagates horizontally toward the array center, then vertically toward the center, then in opposite directions horizontally and vertically, and so on. Obviously, the spirals do not cross the central axes. A shaded example is shown below:

```
int* segment(int *start, int length, int direction, int increment)
{
    for (; length > 0; length--)
    {
        *(start + direction) = *start + increment;
        start += direction;
    }
    return start;
}
```

	2	3	0	0	
8	9	4	0	0	0
7	6	5		0	0
0	0	0	5	6	7
0		0	4	9	8
0	0	0	3	2	1

void spiral2(int *a2D, int even_size) {
 int *a;

a = segment(a2D, even_size/2-1, 1);

a = segment(a+even_size/2-1, -even_size/2);

a = segment(a+even_size/2-1, -1, 1);

a = segment(a, even_size/2, even_size, 1);

a = segment(a, even_size/2-3, 1); }

2