

AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
COMP120 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)`
`void reverse(int i = 0; i < length int a1D[], int length)`
`void transpose(int *a2D, int size)`
int size = a
int length = b
Array [a][b] ptr [a][b]
ptr {a, b}

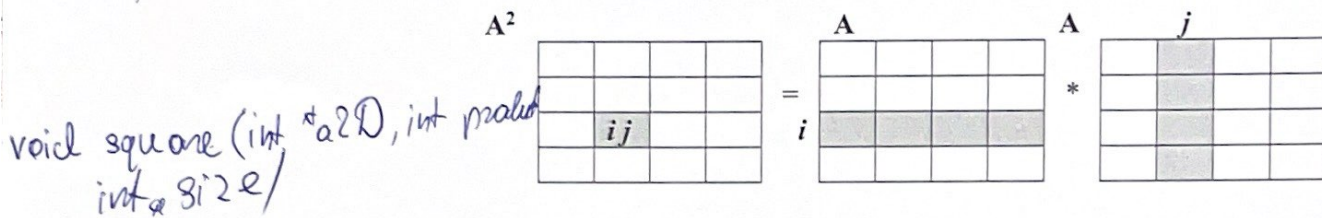
OOP. MT2. 240315. L122

Problem 2

Using functions *transpose()* from **Problem 1** and *scalar()* from below, write a C++ function *void square(int *a2D, int *product, int size)* that takes as its argument a pointer to the first element of a square array *int *a2D* of the specified *int size*, computes its square (multiplies it by itself) and saves it in another square array of the same size, the pointer to the first element of which is given by *int *product*. Each element p_{ij} in the i^{th} row and j^{th} column of the array **product* is the scalar product of the i^{th} row and j^{th} column of the array **a2D* and is calculated by the

$$\text{expression: } p_{ij} = \sum_{k=0}^{\text{size}-1} a_{ik} a_{kj}$$

```
int scalar(int a[], int b[], int length)
{
    int result = 0;
    for (int i = 0; i < length; i++)
        result += a[i] * b[i];
    return result;
}
```



void square(int *a2D, int product
int size)

Array [a b].
ptr a, b
int product = a.b
int a = int b

void transpose (int *a2D, int size)
int scalar (int a[], int b[], int length).

1

QOP.MT2. 240315. 122