Контрольная работа №1: Матрицы, элементарные структуры данных  
\*Условия восстановлены по памяти

Задание 1. Имеется матрица вещественных чисел с двойной точностью, структура которой определена в файле “mylib.h”. Написать функцию func(), которая преобразует матрицу по условию ниже. Функция принимает на вход указатель на матрицу и возвращает указатель на другую матрицу – изменять исходную матрицу нельзя!

typedef struct Line {

int size;

double \*data;

} Line;

typedef struct Matrix {

int size;

Line \*lines;

} Matrix;

Требуется поменять местами первую строку матрицы с той строкой, которая содержит максимальное количество строго положительных чисел.

Задание 2. Имеется стек, реализованный на списке, который имеет структуру, представленную ниже. Требуется написать функцию push(), которая принимает в качестве аргументов указатель на стек и целое значение, которое требуется вставить, и возвращает 0 в случае успеха и 1 в случае ошибки. Программа должна подключать все необходимые библиотеки сама, включая заголовочный файл со стеком “mylib.h”!

typedef struct Node {

int k;

Node \*next;

} Node;

typedef struct Stack {

Node \*head;

} Stack;

Задание 3. (точно такое же, какие были задания в лекциях на тестированиях)

Задана очередь, реализованная на списке, реализованы функции add(el), которая вставляет значение el в очередь, и get(\*pel), которая читает значение из очереди и записывает по указателю pel. Дана некая последовательность действий с очередью. Определить конечное состояние очереди и значение el.

(например, могут быть такие данные):  
Исходное состояние очереди: Q1 {5}, Q2 {-71}, Q3 {27}, Q4 {3}, Q5 {-51}

int el;

add(83);

add(-91);

add(113);

get(&el);

get(&el);

Задание 4 – такое же как и задание 3, но с другими данными (очередь/стек, вектор/список, другое исходное состояние и последовательность действий)

Задание 5.

Дана программа. Определить, что она выведет:

int e=8, p, s;

while (e < 14) {

p = e++;

s = ++e;

}

printf(“%d %d %d”, e, p, s);