**Контрольная работа по темам  
                   “Одномерные  массивы, матрицы и функции”**

**Уровень А ( max 8 баллов)**

**1.** Составить две функции. Первая, встроенная, отличная от *void*, для *одного целого десятичного числа u* (а не для индексированной переменной) находит количество единиц в его двоичном представлении. Во второй функции типа *void* с прототипом с помощью первой функции в *одномерном массиве* найти число с наибольшим количеством единиц в двоичном представлении (одно, любое, если таких чисел несколько) и наибольшее количество единиц. В *main* объявить массив, определить его при объявлении с помощью констант, вызвать вторую функцию и вывести полученные с ее помощью результаты.

2.  Составить две функции. Первая типа *void*с прототипом вводит *одномерный массив* оценок с контролем ввода. Оценки должны быть в диапазоне 1 — 10. Если ввели i-ю оценку меньше 1 или больше 10, повторить ввод оценки для этого же индекса i. Вторая встроенная функция, отличная от *void,* *в одномерном массиве* находит количество плохих оценок. В *main* объявить массив, вызвать первую и вторую функции и вывести полученный во второй функции результат.

3.  Составить две функции. Первая, встроенная, отличная от *void*, для *одного целого десятичного числа u* (а не для индексированной переменной)  находит максимальную цифру (2032 à 3). Во второй функции типа *void*с прототипом с помощью первой функции, в *одномерном массиве* найти число с наименьшей максимальной цифрой (одно, любое, если таких чисел несколько) и наименьшую максимальную цифру. В *main* объявить массив, ввести его, вызвать вторую функцию и вывести полученные с ее помощью результаты.

4.  Составить две функции. Первая функция, встроенная, отличная от *void,* находит (но не выводит, а возвращает в*main)* наименьшее среди положительных чисел одномерного массива. Если положительных чисел нет, вернуть в main какое-нибудь отрицательное число, например, -1. Вторая функция типа *void*с прототипомвыводит только положительные числа массива в том же порядке. В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (наименьшее среди положительных чисел или текст “ Нет положит”, если из функции вернули -1).

5.  Составить две функции. Первая, встроенная, отличная от *void*, для *одного целого десятичного числа u* (а не для индексированной переменной)  находит сумму десятичных цифр (263 à 11). Во второй функции типа *void* с прототипомс помощью первой функции в *одномерном массиве* найти число с наибольшей суммой десятичных цифр (одно, любое, если таких чисел несколько) и наибольшую сумму десятичных цифр среди чисел массива. В *main* объявить массив, определить его при объявлении с помощью констант, вызвать вторую функцию и вывести полученные с ее помощью результаты.

6.  Составить две функции. Первая функция, встроенная,  отличная от *void,* находит (но не выводит, а возвращает в*main)* второе наибольшее число одномерного массива (10, 20, -30, 40, 55 à 40, но    для массива 5, 5, 5, 5, 5  вернуть в *main* какое-нибудь отрицательное число, например,  -1000).  Вторая функция типа *void*с прототипомвыводит массив в обратном порядке. В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (второе наибольшее число или текст *“нет второго наибольшего”,*если вернули -1000).

7. Составить две функции. Первая, встроенная, отличная от *void*, для *одного целого десятичного числа u* (а не для индексированной переменной) находит последнюю справа (6943 à 3) десятичную цифру. Во второй функции типа *void*с прототипом с помощью первой функции в *одномерном массиве* найти для каждого числа последнюю справа десятичную цифру и вывести их “лесенкой” (число  -- последняя цифра):

   156  -- 6

               20 -- 0

                        5 -- 5

                                  -123 -- 3

В *main* объявить массив, ввести его и вызвать вторую функцию.

8. Составить две функции. Первая логическая, встроенная для *одного целого десятичного числа u* (а не для индексированной переменной) возвращает *true*, если число простое и *false* в противном случае. Во второй функции типа *void*с прототипом с помощью первой функции в *одномерном массиве* найти количество простых чисел и наибольшее простое число. В *main* объявить массив, ввести его, вызвать вторую функцию и вывести полученные с ее помощью два результата.

9.      Составить две функции. Первая, встроенная, отличная от *void,* находит (но не выводит, а возвращает в*main)* среднеарифметическое значение среди положительных чисел одномерного массива. Если положительных чисел нет, вернуть в main какое-нибудь отрицательное число, например, -1. Вторая функция типа *void*с прототипомвыводит массив по *к* чисел в одной строке (массив, его размерность и *k—*входные параметры функции). В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (среднеарифметическое или текст ‘ Нет положит”, если из функции вернули -1).

10.   Составить две функции. Первая функция, встроенная, отличная от *void,* находит (но не выводит, а возвращает в*main)* количество чисел одномерного массива из промежутка *[u,v]* (массив, его размерность, *u, v* *—*входные параметры функции). Вторая функция типа *void* с прототипом*выводит* только числа массива из промежутка *[u,v]* в обратном порядке (массив, его размерность, *u, v* *—*входные параметры функции). В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (количество чисел массива из промежутка *[u,v]* или текст “ Нет чисел ”, если из функции вернули 0).

11.   Составить две функции. Первая, встроенная, отличная от *void*, для *одного целого десятичного числа u* (а не для индексированной переменной)  находит первую слева (6943 à 6) десятичную цифру. Во второй функции типа *void* с прототипомс помощью первой функции в *одномерном массиве* найти число с наибольшей первой слева десятичной цифрой (одно, любое, если таких чисел несколько) и наибольшую первую слева десятичную цифру среди чисел массива. В *main* объявить массив, ввести его, вызвать вторую функцию и вывести полученные с ее помощью результаты.

12.   Составить две функции. Первая функция, встроенная, отличная от *void,* находит (но не выводит, а возвращает в*main)* номер наибольшего числа одномерного массива (один, наименьший номер, если наибольших чисел несколько). Вторая функция типа *void*с прототипомвыводит массив лесенкой:

                                                                               a[0]

                                                                    a[1]

                                                        a[2]

                                               …

                      a[n-1]

В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (номер наибольшего числа и по номеру это наибольшее число).

13. Составить две функции. Первая логическая функция, встроенная, возвращает *true*, если в одномерном целочисленном массиве оценок есть хотя бы одна плохая оценка из диапазона 1 – 3 и *false* в противном случае. Вторая внешняя функция типа *void*с прототипомвыводит массив в обратном порядке в “столбик”, то есть по одному числу в строке. В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести с помощью полученного в первой функции результата текст *“есть плохие оценки”*или*“нет плохих оценок”*.

14. Составить две функции. Первая типа *void*с прототипом вводит одномерный массив оценок одного студента с контролем ввода. Оценки должны быть в диапазоне 1 — 10. Если ввели *i*-ю оценку меньше 1 или больше 10, повторить ввод оценки для этого же индекса *i*. Вторая функция логического типа, встроенная, возвращает *false*, если в массиве есть хотя бы одна оценка меньше 9, то есть это не отличник и *true* в противном случае, если нет оценок меньше 9, то есть если в массиве оценки отличника. В *main* объявить массив, вызвать первую и вторую функции и вывести с помощью полученного во второй функции результата текст *“отличник”*или*“не отличник”*.

15. Составить две функции. Первая функция, встроенная, отличная от *void,* находит расстояние между двумя точками плоскости с координатами (u, v) и (w, z). *Входные параметры — четыре простые, неиндексированные вещественные переменные*. Во второй функции типа *void*, оформленной как внешняя с прототипом, даны точки плоскости своими координатами в виде двух одномерных вещественных массивов *x*[*n*] и *y*[*n*], где (*xi*, *yi*) – координаты *i*-й точки плоскости. Используя первую функцию, найти наибольшее расстояние от начала координат и координаты одной, любой точки с наибольшим расстоянием от начала координат. В *main* объявить и определить при объявлении два одномерных вещественных массива, вызвать вторую функцию и вывести полученный с ее помощью результат, то есть три вещественных числа.

16. Составить две функции. Первая функция, встроенная, отличная от *void,* находит (но не выводит, а возвращает в*main)* последнее положительное число одномерного массива четной размерности. Предусмотреть случай, когда положительных чисел нет. (10, 20, -30, 40, -55 0  à 40,   для массива 5, 5, 5, 5, 5, 5  вернуть в *main* 5, но для массива -11, -20, -3, -400, -5, -66 вернуть -1).  Вторая функция типа *void*с прототипомвыводит массив по два числа в строке лесенкой:

                                                                               a[0]   a[1]

                                                               a[2]   a[3]

                                                a[4]   a[5]

                                      …

                      a[n-2]  a[n-1]

В *main* объявить массив, определить его при объявлении, вызвать первую и вторую функции и вывести полученный в первой функции результат (последнее положительное число или текст *“нет положительных чисел”,*если вернули -1).

*17.* Составить две функции. Первая функция, встроенная, отличная от *void,* находит расстояние между двумя точками плоскости с координатами (u, v) и (w, z). *Входные параметры — четыре простые, неиндексированные вещественные переменные*. Во второй функции типа *void*, оформленной как внешняя с прототипом, даны точки плоскости своими координатами в виде двух одномерных вещественных массивов *x*[*n*] и *y*[*n*], где (*xi*, *yi*) – координаты *i*-й точки плоскости. Используя первую функцию, найти и вывести все точки с наибольшим расстоянием от заданной точки. Входные параметры второй функции: два массива, их одинаковая размерность, две координаты фиксированной точки (5 входных параметров); выходных параметров нет. В *main* объявить и определить при объявлении два одномерных вещественных массива, ввести координаты одной точки, от которой будем находить расстояния и вызвать вторую функцию.

18.  Составить две функции.  Первая типа *void,*оформленная как внешняя с прототипом, вводит с экрана два одномерных вещественных массива *x*[*n*] и *y*[*n*] парами (*xi*, *yi*), где (*xi*, *yi*) – координаты *i*-й точки плоскости. Во второй встроенной функции, отличной от *void*, дан массив точек плоскости в виде двух вещественных массивов и надо найти количество точек внутри круга с центром в точке (*u, v*) радиуса *r*. Входные параметры второй функции: два массива, их одинаковая размерность, две координаты центра круга и его радиус (6 входных параметров); выходной параметр один — количество точек, принадлежащих кругу. В *main* объявить два одномерных вещественных массива. С помощью первой функции ввести их, вызвать вторую функцию и вывести полученный результат, то есть одно целое число.

19. Составить две функции. Первая функция, встроенная, отличная от *void,* находит расстояние между двумя точками плоскости с координатами (u, v) и (w, z). *Входные параметры — четыре простые, неиндексированные вещественные переменные.* Во второй функции, оформленной как внешняя с прототипом, даны точки плоскости своими координатами в виде двух одномерных вещественных массивов *x*[*n*] и *y*[*n*], где (*xi*, *yi*) – координаты *i*-й точки плоскости. Используя первую функцию, найти количество точек, принадлежащих сектору первой четверти с центром в начале координат единичного радиуса и вывести только координаты точек первой четверти в том же порядке, что в исходном массиве. В *main* объявить и определить при объявлении два одномерных вещественных массива, вызвать вторую функцию.

20. Составить две функции.  В первой встроенной функции, отличной от *void*, дан массив точек плоскости, то есть два одномерных вещественных массива одинаковой размерности *x*[*n*] и *y*[*n*], где (*xi*, *yi*) – координаты *i*-й точки плоскости. Надо найти количество точек внутри треугольника, вершины которого в заданных точках (-1,0), (0, 1), 1, 0). Вторая функция типа *void,*оформленная как внешняя с прототипом, выводит на экран два массива по два числа в строке лесенкой:

                                                                               x[0]   y[0]

                                                              x[1]   y[1]

                                             x[2]   y[2]

                                      …

                      x[n-1]  y[n-1]

В *main* объявить и определить при объявлении два одномерных вещественных массива, вызвать первую и вторую функции и вывести полученный в первой функции результат, то есть одно целое число.

***Уровень B на 10 баллов***

*Вариант 1.* Составить функцию, которая: в *одномерном массиве* находит среднее значение среди положительных чисел. В *main* определить целочисленную матрицу при объявлении. С помощью функции для каждой строки матрицы найти среднее значение среди положительных чисел строки. Вывести матрицу и справа после каждой строки другим цветом вывести полученное с помощью функции её среднее значение среди положительных чисел этой строки.

*Вариант 2.* Составить логическую функцию, которая для *одного целого числа*  определяет, простое это число или нет. В *main* ввести целочисленную квадратную матрицу и с помощью функции на побочной диагонали матрицы найти и вывести количество простых чисел. При выводе матрицы простые числа главной диагонали выделить другим цветом.

*Вариант 3.* Составить функцию, которая: в *одномерном массиве* находит наибольший элемент и его номер. В *main* определить матрицу при объявлении и с помощью функции в каждой строке матрицы найти наибольший элемент и его номер в строке. Вывести матрицу и справа после каждой строки другим цветом вывести полученные с помощью функции наибольший элемент и его номер в строке.

*Вариант 4.* Составить логическую функцию, которая для *одного целого числа* определяет, является ли оно кратным числу K. В *main* определить целочисленную матрицу при объявлении и вывести матрицу и справа от неё после каждой строки другим цветом номер первого чётного числа этой строки, полученный с помощью логической функции. Если чётных чисел в строке нет, вывести *“Нет чётных чисел”.*

*Вариант 5.* Составить функцию, которая: в *одномерном* массиве (а не в матрице) находит сумму модулей до первого нуля. Если нуль в начале массива, получить нуль, а если нулей нет — сумму модулей всех чисел массива. В *main* определить матрицу при объявлении и вывести матрицу и справа от неё после каждой строки другим цветом полученную с помощью функции сумму модулей до первого нуля.

*Вариант 6.* Составить логическую функцию, которая для *одного целого числа* определяет, есть ли нуль в его двоичной системе счисления. В *main* определить целочисленную матрицу при объявлении и с помощью функции на главной диагонали найти и вывести количество чисел, у которых нет нулей в двоичной системе счисления (например, 7-> 111, 15 -> 1111, 63 -> 111111). При выводе матрицы такие числа выделить другим цветом.

*Вариант 7.* Составить функцию, которая *для одномерного массива* оценок возвращает единицу, если это оценки отличника (то есть в массиве только 9 и (или) 10 и других оценок нет) и нуль в противном случае. В *main* ввести матрицу *A*, в которой *Ai,j* — оценка *i*–го студента на *j*–м экзамене по 10–балльной системе. Предусмотреть контроль ввода, то есть если введём несуществующую оценку, то повторить ввод элемента матрицы с этим же индексом. В матрице с помощью функции найти и вывести количество отличников, то есть количество строк с соответствующими числами, а не количество отличных оценок. Вывести матрицу, строки отличников выделив другим цветом.

*Вариант 8.* Составить логическую функцию, которая для *одного целого числа* определяет, есть ли буква в его шестнадцатеричной системе счисления. В *main* определить целочисленную квадратную матрицу при объявлении, вывести её правый нижний треугольник и с помощью функции в правом нижнем треугольнике квадратной матрицы найти  и вывести количество чисел, у которых в 16-й системе счисления есть хотя бы одна буква. Такие числа в правом нижнем треугольнике квадратной матрицы выделить другим цветом.

*Вариант 9.* Составить функцию, которая для *одномерного  массива* оценок  возвращает единицу, если это оценки двоечника (есть хотя бы одна оценка 1 и(или) 2 и(или) 3 независимо от других оценок) и нуль в противном случае. В *main* ввести целочисленную матрицу *A*, в которой *Ai,j* — оценка *i*–го студента на *j*–м экзамене по 10–балльной системе. Предусмотреть контроль ввода, то есть если введём несуществующую оценку (меньше единицы или больше 10), то повторить ввод элемента матрицы с этим же индексом. Для матрицы с помощью функции найти и вывести  количество двоечников, то есть количество строк, в которых есть хотя бы одна оценка 1 и(или) 2 и(или) 3. Вывести матрицу, строки двоечников выделив другим цветом.

*Вариант 10.* Составить логическую функцию, которая для *одного целого числа* определяет, все ли буквы в шестнадцатеричной системе счисления. В *main* определить целочисленную квадратную матрицу при объявлении, вывести её левый нижний треугольник и с помощью функции в левом нижнем треугольнике квадратной матрицы найти  и вывести количество чисел, у которых в шестнадцатеричной системе счисления числа все буквы. Такие числа в левом нижнем треугольнике квадратной матрицы при выводе выделить другим цветом.

*Вариант 11.* Составить функцию, которая *для одномерного  массива* оценок  возвращает;

2, если в одномерном массиве есть хотя бы одна неудовлетворительная оценка;

3, если есть хотя бы одна 4 и (или) 5, но нет неудовлетворительных оценок;

4, если есть хотя бы одна 6, 7, и (или) 8, но нет неудовлетворительных и удовлетворительных оценок,

5, если все оценки отличные (9 и (или) 10).

В *main* ввести матрицу *A*, в которой *Ai,j* — оценка *i*–го студента на *j*–м экзамене по 10–балльной системе. Предусмотреть контроль ввода, то есть если введём несуществующую оценку, то повторить ввод элемента матрицы с этим же индексом. Для каждой строки матрицы с помощью функции получить рейтинг и вывести матрицу, а справа от каждой строки другим цветом вывести одно из полученных чисел (2 или 3 или 4 или 5).

*Вариант 12.* Составить функцию, которая *для одномерного  массива* оценок  возвращает общее количество плохих (1,2,3) оценок. В *main* ввести матрицу *A*, в которой *Ai,j* — оценка *i*–го студента на *j*–м экзамене по 10–балльной системе. Предусмотреть контроль ввода, то есть если введём несуществующую оценку, то повторить ввод элемента матрицы с этим же индексом. Для матрицы с помощью функции найти и одну любую строку с наибольшим количеством плохих оценок. Вывести матрицу, выделив другим цветом такую строку.

*Вариант 13.* Составить логическую функцию, которая для *одного целого числа* определяет, есть ли буква в шестнадцатеричной системе счисления. В *main* определить целочисленную квадратную матрицу при объявлении, вывести её левый верхний треугольник и с помощью функции в левом верхнем треугольнике квадратной матрицы найти  и вывести количество чисел, у которых в шестнадцатеричной системе счисления числа есть буква. Такие числа в левом верхнем треугольнике квадратной матрицы при выводе выделить другим цветом.

*Вариант 14.* Составить функцию, которая выводит *одно целое число* в шестнадцатеричной системе счисления, не используя формат “*%X*” функции *printf* и модификатор *hex* для *cout*. В *main* определить целочисленную матрицу при объявлении, одним цветом вывести матрицу в десятичной системе счисления, а другим цветом с помощью функции вывести матрицу в шестнадцатеричной системе счисления.

*Вариант 15.* Составить функцию, которая выводит одно целое число в двоичной системе счисления. В *main* ввести матрицу, используя управление курсором, и с помощью функции вывести её в двоичной системе счисления.

*Вариант 16.* Составить функцию, которая: в *одномерном массиве* находит сумму модулей до первого нуля. Если нуль в начале массива, получить нуль, а если нулей нет — сумму модулей всех чисел массива. В *main* определить матрицу при объявлении и вывести матрицу, а справа от неё после каждой строки другим цветом вывести полученную с помощью функции сумму модулей до первого нуля.

*Вариант 17.* Составить одну функцию, которая в *одномерном**массиве* находит наибольший и наименьший элементы. В *main* ввести матрицу как матрицу, используя управление курсором, и с помощью функции в каждой строке матрицы найти эти два числа, вывести их, а справа после этих двух чисел другим цветом вывести соответствующую строку матрицы.

*Вариант 18.* Составить функцию, которая в *одномерном массиве* меняет местами (переставляет) наибольший и наименьший элементы. Предполагается, что наибольший и наименьший элементы не повторяются. В *main* ввести матрицу, с помощью функции преобразовать каждую строку и вывести исходную матрицу, а справа от неё другим цветом вывести изменённую матрицу.

*Вариант 19.* Составить функцию, которая в *одномерном массиве* находит наименьшее значение среди положительных чисел. Если положительных чисел нет, получить и возвратить отрицательное число -1. В *main* определить матрицу при объявлении и с помощью функции в каждой строке найти наименьшее значение среди положительных чисел этой строки, вывести его или текст *“ Нет положительных”*(если функция вернула -1), а справа другим цветом вывести соответствующую стоку матрицы строки матрицы.

*Вариант 20.* Составить логическую функцию, которая для *одного целого числа* определяет, есть ли цифра “нуль” в числе в десятичной системе счисления. В *main* ввести целочисленную квадратную матрицу как матрицу, используя управление курсором, и с помощью функции на побочной диагонали матрицы найти и вывести количество чисел, в которых есть цифра “нуль” в числе. Вывести матрицу, выделив другим цветом такие числа на побочной диагонали.

*Вариант 21.* Составить одну функцию, которая для *одного целого числа*  находит и возвращает сумму цифр числа (2021 à 1+2+0+2=5) и наибольшую цифру (2021 à 2). В *main* ввести целочисленную квадратную матрицу как матрицу, используя управление курсором. С помощью функции для каждого числа побочной диагонали матрицы найти и вывести сумму цифр и наибольшую цифру.

*Вариант 22.* Составить функцию, которая в *одномерном массиве* находит наибольшее число и его номер. Если наибольших чисел несколько, найти номер первого из них. В *main* определить матрицу при объявлении и с помощью функции в каждой строке найти наибольшее число и его номер в строке, вывести эти параметры, а справа после этих двух чисел другим цветом вывести строку матрицы.