ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федерального государственное бюджетное образовательное учреждение
высшего образования
«Иркутский государственный университет путей сообщения»
(ФГБОУ ВО ИрГУПС)

С.А. Махнев

Программирование

Иркутск 2022

Лабораторная работа №7

Подпрограммы

Тема: Создание программ, использующих пользовательские подпрограммы-функции (процедуры).

1. **Цель работы:** Освоить создание и работу с подпрограммами-функциями и процедурами. Освоить методы тестирования программ.

Задание на лабораторную работу

2. Получить задание в соответствии с номером своего варианта.
3. Составить схему алгоритма решения задачи.
4. Написать код программы на языке C++
5. Осуществить компиляцию, и отладку программы. Провести ***структурное*** тестирование программы (тестирование как ***белого*** ящика): составить тестовые наборы для тестирования:
   - в ***нормальных*** условиях,
   - при ***граничных*** условиях,
   - при ***некорректных*** входных данных,
   - по критерию ***покрытия решений*** – результат проверки каждого условия принимал значение ***истина*** или ***ложь*** хотя бы один раз,
   - по критерию ***покрытия условий*** – все возможные результаты каждой логической операции в условиях были получены хотя бы по одному разу.
6. Получить результат.
7. Оформить отчет по лабораторной работе.

Пример оформления работы

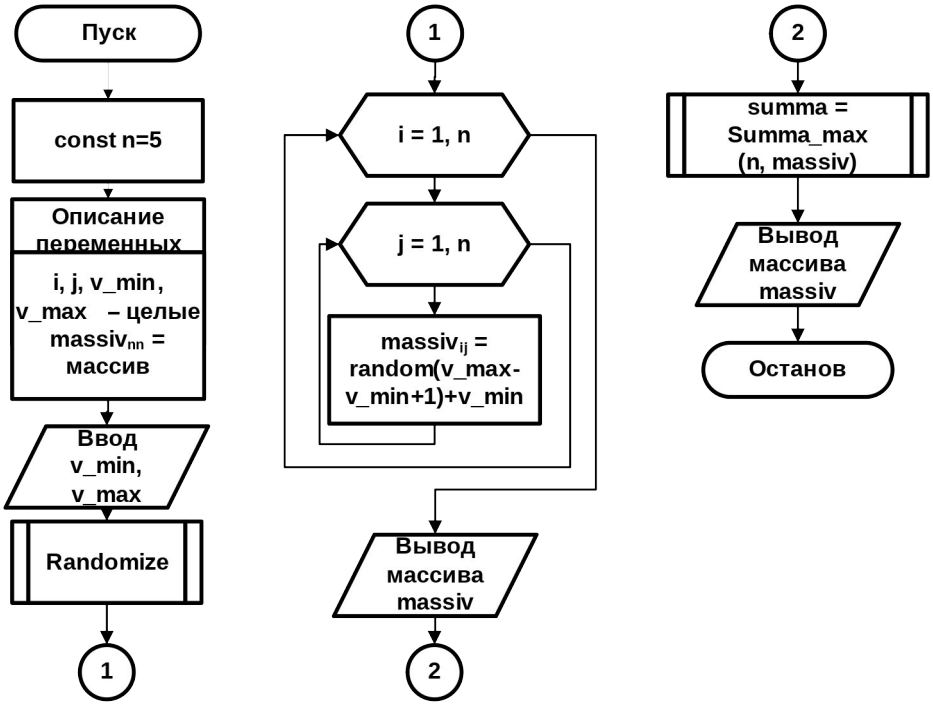
1) Создать функцию, определяющую сумму максимальных элементов главной и побочной диагоналей массива $massiv_{nn}$.

Перед началом поиска максимального элемента ***max_1*** главной диагонали массива принимаем за него элемент $massiv_{11}$. После этого сравниваем остальные элементы главной диагонали $massiv_{ii}$ $(i=2,\ldots,n)$ с текущим максимумом: если очередной элемент больше текущего максимума ***max_1***, то сейчас он – текущий максимум.

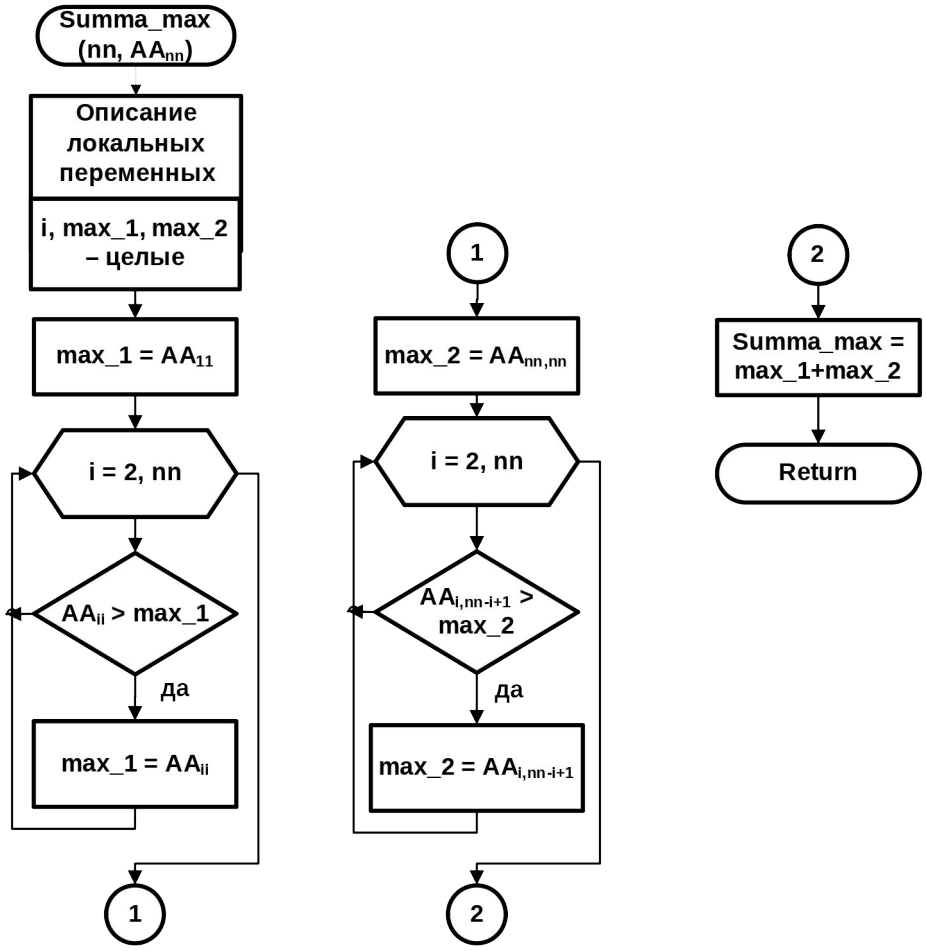
Перед началом поиска максимального элемента ***max_2*** побочной диагонали массива принимаем за него элемент $massiv_{nn}$. После этого сравниваем остальные элементы побочной диагонали $massiv_{ij}$ $(i=2,\ldots,n;\ j=n-i+1)$ с текущим максимумом: если очередной элемент больше текущего максимума ***max_2***, то сейчас он – текущий максимум.

2) Схема алгоритма решения задачи

Головная программа



Подпрограмма-функция:



3) Код программы

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <time.h>

                    // прототипы функций

void vorm_mass(int nn, int mass_max, int mass_min, int
mass[nn][nn]);
int summa_max(int nn, int mass[nn][nn]);

int main()
{
     int i, j, n, massiv_min, massiv_max, summa;

     printf("\nЛабораторная работа №7");
     printf("\nВариант №0");
     printf("\nПИ-22-1");
     printf("\nПупкин Василий\n");

     printf("\nВведите нижнюю границу диапазона:");
     scanf("%d", &massiv_min);
     printf("\nВведите верхнюю границу диапазона:");
```

```
     scanf("%d", &massiv_max);
     printf("\nВведите размер массива:");
     scanf("%d", &n);

     int massiv[n][n];

     printf("\nИсходный массив:\n");

   vorm_mass(n, massiv_max, massiv_min, massiv);

     for (i=0; i<n; i++)
     {
          for (j=0; j<n; j++)
               printf("%5d", massiv[i][j]);
         printf("\n");
     }

     printf("\n");

   summa = summa_max(n, massiv);

     printf("\nСумма максимальных элементов диагоналей: %d",
summa);

     return 0;
}
                         //описание функций

                         // 1. формирование массива
void vorm_mass(int nn, int mass_max, int mass_min, int
mass[nn][nn])
{
     int i, j, m;
     time_t t;      // текущее время для инициализации
                         // генератора случайных чисел

     srand((unsigned) time(&t));   // инициализация генератора
                                        // случайных чисел

     m = mass_max-mass_min + 1;

          // получение случайного числа в диапазоне
          // от mass_min  до  mass_max

     for (i=0; i<nn; i++)
          for (j=0; j<nn; j++)
               mass[i][j]=rand()% m + mass_min;
}

// 2. определение суммы максимальных элементов диагоналей
массива
int summa_max(int nn, int mass[nn][nn])
{
```

```
    int i, max_1, max_2;

    max_1 = mass[0][0];
    for (i=1; i<nn; i++)
        if (mass[i][i] > max_1)
            max_1 = mass[i][i];

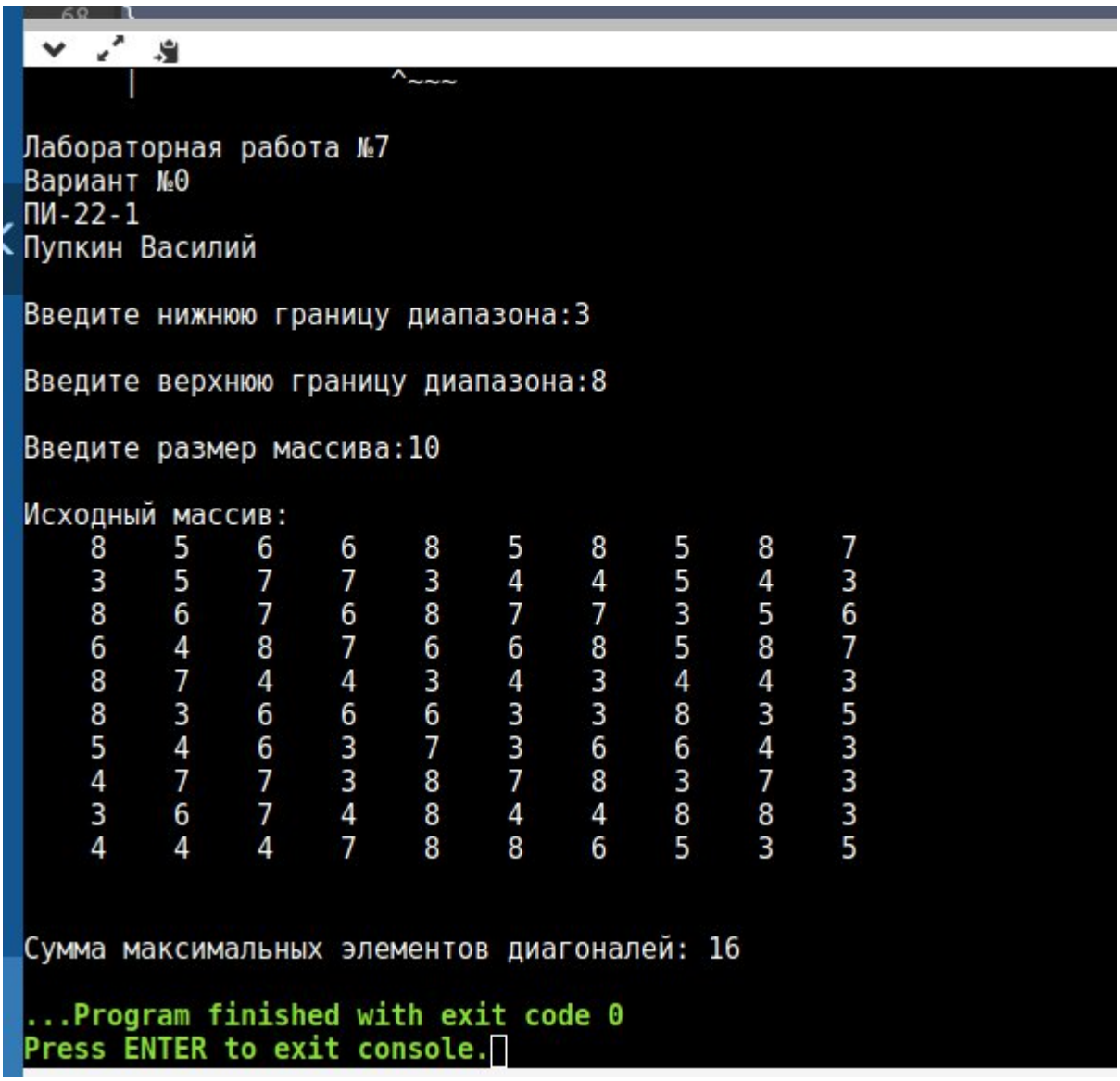
    max_2 = mass[0][nn-1];
    for (i=1; i<nn; i++)
        if (mass[i][nn-i-1] > max_2)
            max_2 = mass[i][nn-i-1];

    return max_1 + max_2;
}
```

4) Тестирование

	Вход	Вывод
в ***нормальных*** условиях,	2;3;3	Случайный, но предполагаю, что сумма максимальных элементов диагонали будет 6
при ***граничных*** условиях,	1;1;1	1;1;0;0
при некорректных входных данных,	Год;1;1 2.3;3;2	Error Error

5) Компиляция, отладка, тестирование и получение результата:


^~~~
Лабораторная работа №7
Вариант №0
ПИ-22-1
Пупкин Василий
Введите нижнюю границу диапазона:3
Введите верхнюю границу диапазона:8
Введите размер массива:10
Исходный массив:
8 5 6 6 8 5 8 5 8 7
3 5 7 7 3 4 4 5 4 3
8 6 7 6 8 7 7 3 5 6
6 4 8 7 6 6 8 5 8 7
8 7 4 4 3 4 3 4 4 3
8 3 6 6 6 3 3 8 3 5
5 4 6 3 7 3 6 6 4 3
4 7 7 3 8 7 8 3 7 3
3 6 7 4 8 4 4 8 8 3
4 4 4 7 8 8 6 5 3 5
Сумма максимальных элементов диагоналей: 16
...Program finished with exit code 0
Press ENTER to exit console.

Пример оформления титульника отчета

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федерального государственное бюджетное образовательное учреждение высшего образования

«Иркутский государственный университет путей сообщения»

(ФГБОУ ВО ИрГУПС)

Факультет «Управление на транспорте и информационные технологии»

Кафедра «Информационные системы и защита информации»

Дисциплина «Программирование»

Тема

Лабораторная работа № 7

Вариант № 1

ЛР.430500.09.04.04.299.ПЗ

Выполнил:
Студент гр. …
….
«___» _____2022 г.

_____

Проверил:
ассистент
Махнев С. А.
«___» _____2022 г.

_____

Иркутск 2022

Задания на лабораторную работу №7

Размеры исходных массивов задавать именованными константами в головной программе. Элементы исходных массивов задавать целыми случайными числами в диапазоне от **v_min** до **v_max**. Границы диапазона **v_min** и **v_max** задать с консоли. Ввод исходных данных и вывод результатов производить в головной программе.

Запрещается использовать в подпрограммах глобальные переменные и рабочие массивы. Векторы передавать в подпрограммы как открытые массивы. Не использовать конкретные числа в качестве фактических параметров процедур – только имена переменных, массивов или имена констант.

Вариант 1

Задание 1

Создать функцию, определяющую максимальный элемент вектора $vector_n$, находящийся между вторым по порядку положительным и предпоследним отрицательным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2.

Создать процедуру, поворачивающую в квадратном массиве $matrix_{nn}$ (n – нечетное) средний наибольший ромб элементов на **90** градусов по часовой стрелке.

Вариант 2

Задание 1

Создать функцию, определяющую минимальный элемент вектора $vector_n$, находящийся между вторым по порядку четным и предпоследним нечетным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2.

Создать процедуру, поворачивающую в квадратном массиве $matrix_{nn}$ (n – нечетное) средний наибольший ромб элементов на **90** градусов против часовой стрелки.

Вариант 3

Задание 1

Создать функцию, определяющую среднее арифметическое значение элементов вектора $vector_n$, находящихся между вторыми по значению максимальным и минимальным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2.

Создать процедуру, определяющую в целочисленном векторе $vector_n$ два элемента, наиболее близкие к среднему арифметическому всех элементов вектора. Элементы, расположенные между ними, отсортировать по возрастанию.

Вариант 4

Задание 1

Создать функцию, определяющую максимальный элемент k-го слоя среднего наибольшего ромба элементов в квадратном массиве $matrix_{nn}$ (n – нечетное). Номер слоя k задать с консоли.

Задание 2.

Создать процедуру, вычисляющую в целочисленном векторе $vector_n$ сумму нечетных элементов с четными индексами и сумму четных элементов с нечетными индексами.

Вариант 5

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве $matrix_{nm}$ наибольший элемент, встречающийся более одного раза. Предусмотреть случай, когда такого элемента нет.

Задание 2

Создать процедуру, определяющую в целочисленном векторе $vector_n$ два любых одинаковых элемента. Элементы вектора, расположенные между ними, отсортировать по возрастанию. Предусмотреть случай, когда таких элементов нет.

Вариант 6

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве $matrix_{nm}$ сумму всех элементов, соприкасающихся углами с его максимальным элементом.

Задание 2

Создать процедуру, переворачивающую квадратный массив $matrix_{nn}$ на **90** градусов по часовой стрелке.

Вариант 7

Задание 1

Создать функцию, определяющую в прямоугольном целочисленном массиве $matrix_{nm}$ количество всех его седловых точек. Седловой точкой матрицы называется ее элемент, одновременно наименьший в своей строке и наибольший в своем столбце.

Задание 2

Создать процедуру, сдвигающую элементы k-ого слоя целочисленного прямоугольного массива $matrix_{nm}$ на p шагов вправо. Номер слоя k и количество шагов p задать с консоли.

Вариант 8

Задание 1

Создать функцию, определяющую в квадратном целочисленном массиве $matrix_{nn}$ сумму элементов, находящихся выше и ниже главной и побочной диагоналей ("**песочные часы**"). Элементы диагоналей в сумму не включать.

Задание 2

Создать процедуру, формирующую вектор $vector_{2n-1}$ из максимальных элементов диагоналей целочисленного квадратного массива $matrix_{nn}$, параллельных его главной диагонали.

Вариант 9

Задание 1

Создать функцию, определяющую в целочисленном квадратном массиве $matrix_{nn}$ сумму элементов, находящихся левее и правее главной и побочной диагоналей ("**бабочка**"). Элементы диагоналей в сумму не включать.

Задание 2

Создать процедуру, перемножающую две матрицы по правилам матричного умножения. Предусмотреть случай, когда размеры матриц заданы некорректно.

Вариант 10

Задание 1

Создать функцию, определяющую в квадратном целочисленном массиве $matrix_{nn}$ (n-нечетное) сумму элементов k-го слоя среднего наибольшего ромба элементов. Номер слоя k задать с консоли.

Задание 2

Создать процедуру, приводящую с помощью допустимых преобразований заданную матрицу коэффициентов системы линейных алгебраических уравнений к треугольному виду.

Вариант 11

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве $matrix_{nm}$ сумму элементов, соприкасающихся сторонами с его максимальным элементом.

Задание 2

Создать процедуру, решающую систему линейных алгебраических уравнений вида $A_{nn} * X_n = B_n$

Где A_{nn} - матрица коэффициентов системы, B_n – вектор правых частей системы, X_n – вектор решения.

Вариант 12

Задание 1

Создать функцию, удаляющую из заданной строки все последовательности символов, находящиеся между корректно записанными круглыми скобками, вместе с этими скобками. Скобки могут быть вложенными.

Задание 2

Создать процедуру, формирующую вектор $vector_{2n-1}$ из максимальных элементов диагоналей целочисленного квадратного массива $matrix_{nn}$, параллельных его побочной диагонали.

Вариант 13

Задание 1

Строка – это правильное арифметическое выражение, содержащее только натуральные числа и знаки арифметических операций между ними **(+, -, *, /).** Создать функцию, подсчитывающую результат при выполнении арифметических операции в порядке их следования.

Задание 2

Создать процедуру, сдвигающую элементы k-ого слоя прямоугольного целочисленного массива $matrix_{nm}$ на p шагов влево. Номер слоя k и количество шагов p задать с консоли.

Вариант 14

Задание 1

Ввести две строки, состоящие только из нулей и единиц. Считая их изображениями двоичных чисел, создать функцию, складывающую их по правилам двоичной арифметики. Сумму преобразовать в десятичное число.

Задание 2

Создать процедуру, определяющую все локальные минимумы в целочисленном прямоугольном массиве $matrix_{nm}$ Найденные минимумы занести в вектор $vector_{2n}$ Если таковых нет, то заполнить вектор нулями. **Локальным минимумом** называется элемент массива, который меньше своих соседних элементов, соприкасающихся с ним сторонами.

Вариант 15

Задание 1

Строка состоит из нескольких слов, разделенных пробелом. Создать функцию, определяющую минимальное количество первых символов слов, по которым можно различить слова в строке.

Задание 2

Создать процедуру, которая упорядочивает строки прямоугольного массива $\mathbf{matrix_{nm}}$ по возрастанию количества одинаковых элементов в каждой строке.