

Л.р. № 6. Подпрограммы.

Цель работы - _____

Исходное задание лабораторной работы №1:

Краткая теория:

Подпрограмма - это _____

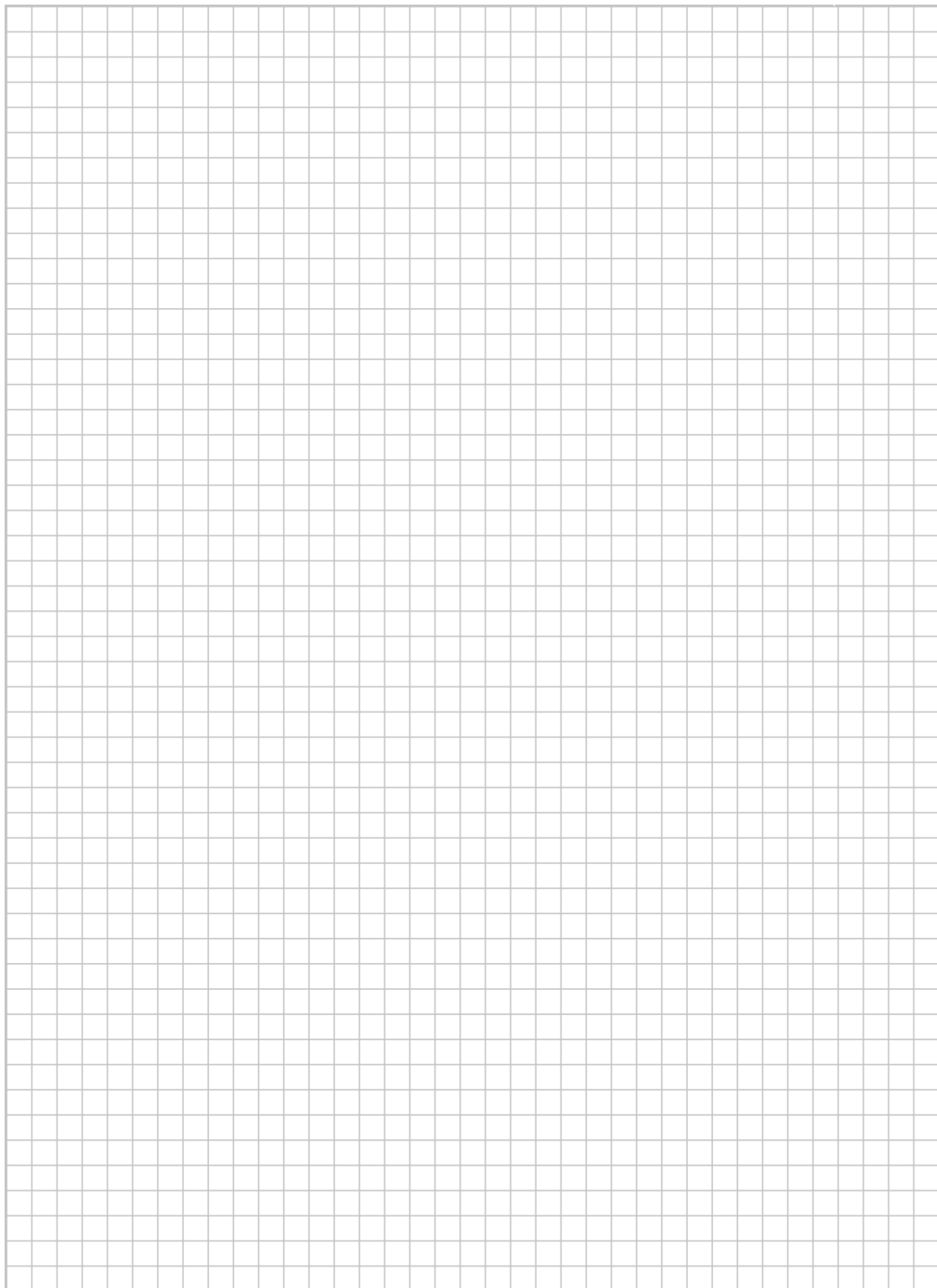
Разбейте ваше задание лабораторной работы № 1 на функционально законченные части: ввода данных, обработки данных(поиск максимального значения, нахождения суммы элементов, транспонирование, перестановка местами элементов матрицы, смена местами строк, смена местами столбцов, сортировка и т.д.), вывода данных. Каждая из этих частей в программе будет реализована в виде отдельной подпрограммы.

Таблица описания будущих подпрограмм

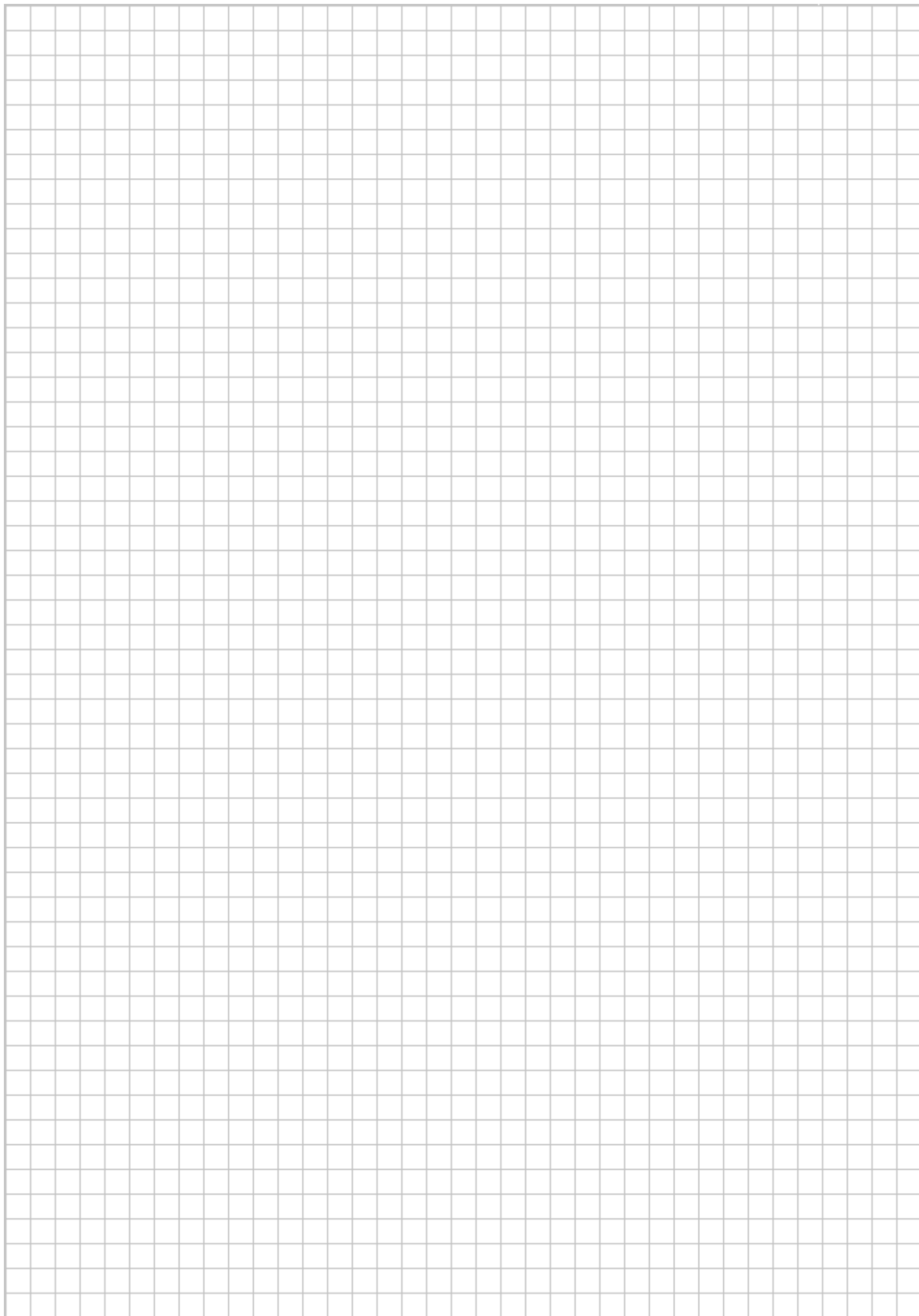
Название (вами придуманное)	Что она делает?	Перечень входных данных	Результат

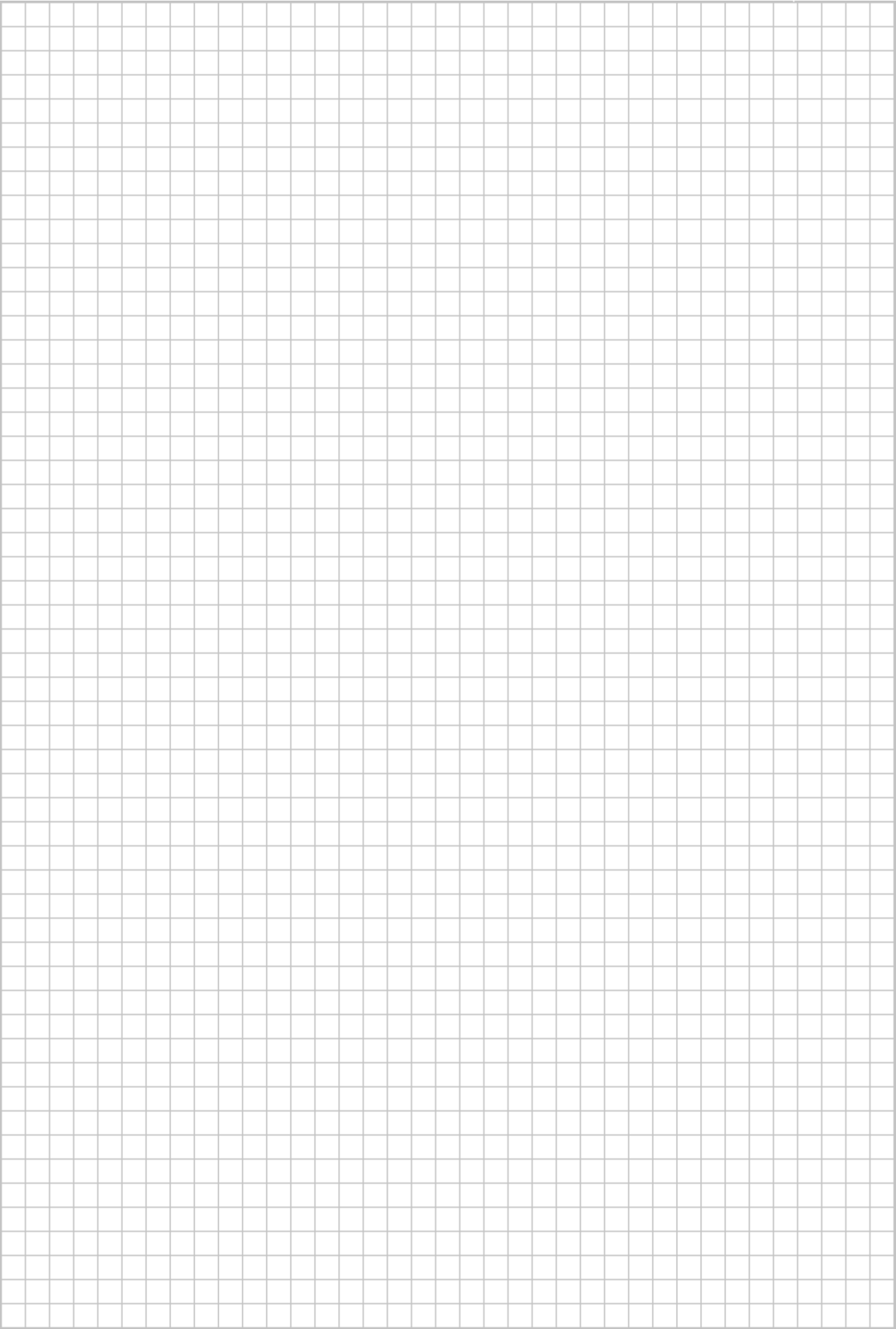
Название (вами придуманное)	Что она делает?	Перечень входных данных	Результат

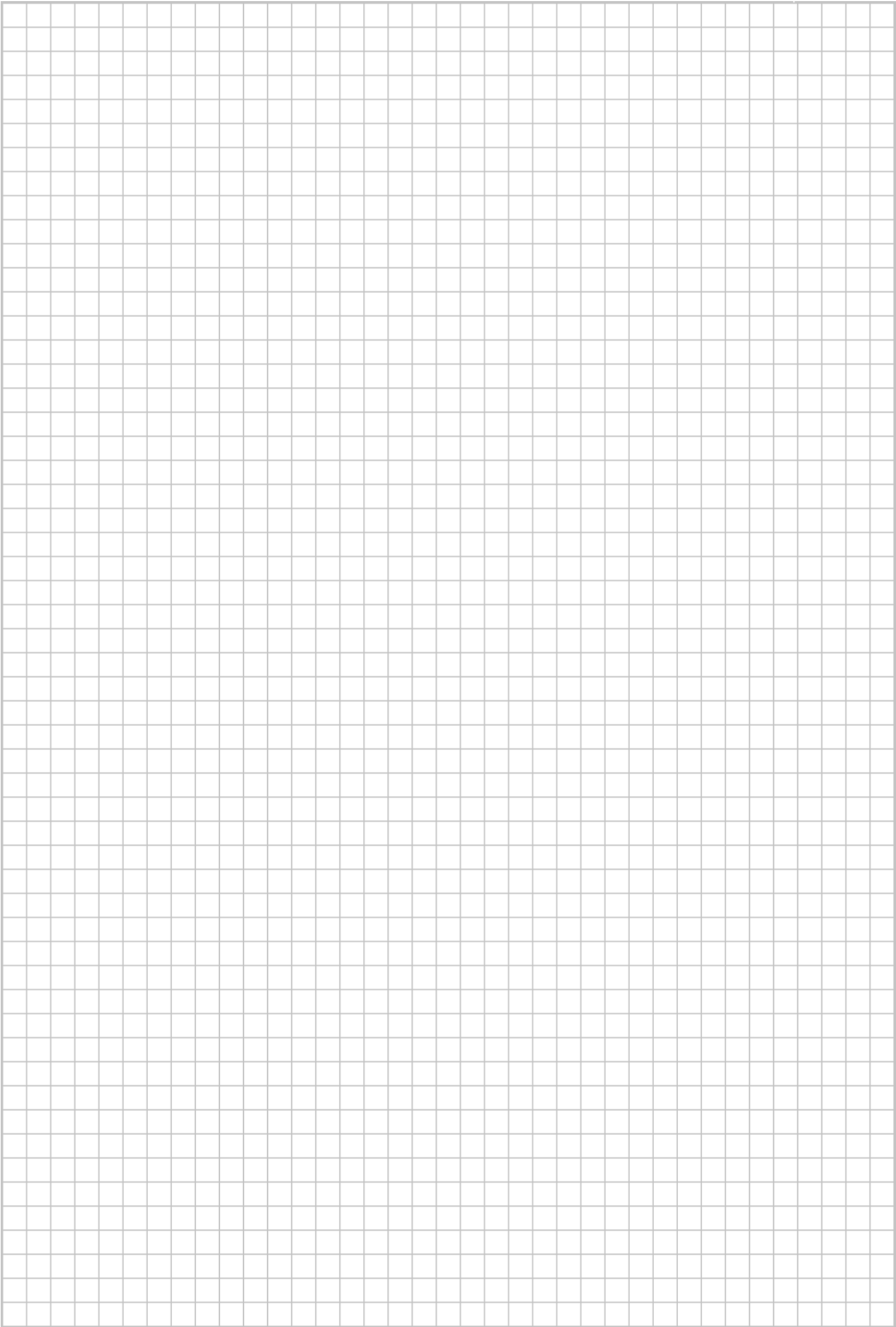
Начертите диаграмму UML последовательности вызова подпрограмм вашей будущей программы



Блок - схемы алгоритмов подпрограмм вашей программы (включая main)







Имеется следующий фрагмент кода:

```
int k;

int add (int a, int b) {
    return (a + b) * k;
}
float average (int a, int b) {
    return ((a+b) / 2.0);
}
void exchange (float &c, int *d) {
    float prom;
    prom = c;
    c = *d;
    *d = prom;
}
void show_number (int a) {
    printf ("Значение параметра равно %i\n", a);
}
int main(int argc, char* argv[]) {
    k = 2;
    int M = 2, G = 4;
    float E = 7;
    E = add(M, G);
    exchange (E, &M);
    show_number(average(E,G));
}
```

Входные параметры подпрограммы – это переменные (указанные в её заголовке), через которые подпрограмма принимает исходные данные для своей работы.

Для каждой подпрограммы найденной Вами в приведённом фрагменте кода перечислите формальные и фактические параметры.

Имя подпрограммы	Формальные параметры	Фактические параметры

У процедуры, записывающей в элементы двумерного параметра **0**-ли, пропали малые параметры. Восстановите их, пожалуйста.

```
void Zero (
    int i, j;
    for (i = 0; i < N; i = i + 1) {
        for (j = 0; j < M; j = j + 1) {
            A[i][j] = 0;
        }
    }
}
```

Прототип подпрограммы - это _____

Все необходимые данные подпрограммам должны передаваться явно, в качестве параметров (в том числе и массив). Использование глобальных переменных для передачи параметров подпрограммам не допускается.

Напишите прототипы подпрограмм своего задания:

Прототипы подпрограмм

Пример создания и использования динамического двумерного массива

Как ни хотелось Петрову чего-либо делать. Может само образуется, думал он. Целыми днями он сидел и думал, какие его плохие люди окружают, все эти математики, физики и т.д. постоянно от него чего-то требуют и просят. Я буду лежать и не сдвинусь с места, думал он, назло им. Чтобы им стало дурно. Не хорошие они, а я от природы самый лучший, самый гениальный. И мне всё это не нужно – я и так всё знаю. Лежал, лежал и долежался он до белой горячки. Стал кукарекать по утрам, а по ночам как собака стал забираться под кровать, думая, что он находится в новой купленной им квартире. «Я гений!!!» – говорил он себе. Потеряв способность к мыслительной деятельности, как-то раз, он решил подшутить над товарищем на его взгляд глупым, так как тот много трудился, не пропуская занятий, разорван на мелкие кусочки его тетрадь по информатике. Сказав: Раз я гений, то ты должен быть таким же, как и я – тоже гением без всякой лишней писанины!!!”.

На одной из страниц был приведён пример создания и использования двумерного динамического массива. Восстановите, пожалуйста, этот пример.

на матрицу

*A;

памяти

new

ление памяти

new int

```
[ ] A[0];  
] A;
```

A[i] = A[i-1] + M;

под массив указателей

* [N] ;

i++)

M; j++)

j;

// Указатель

int *

// Расстановка указателей:

под элементы матрицы:

int [N*M] ;

// Работа с матрицей:

for (i = 0;

for (j =

A[i][j]

// Выделение

A[0] =

i < N;

0; j <

= i +

// Удаление:

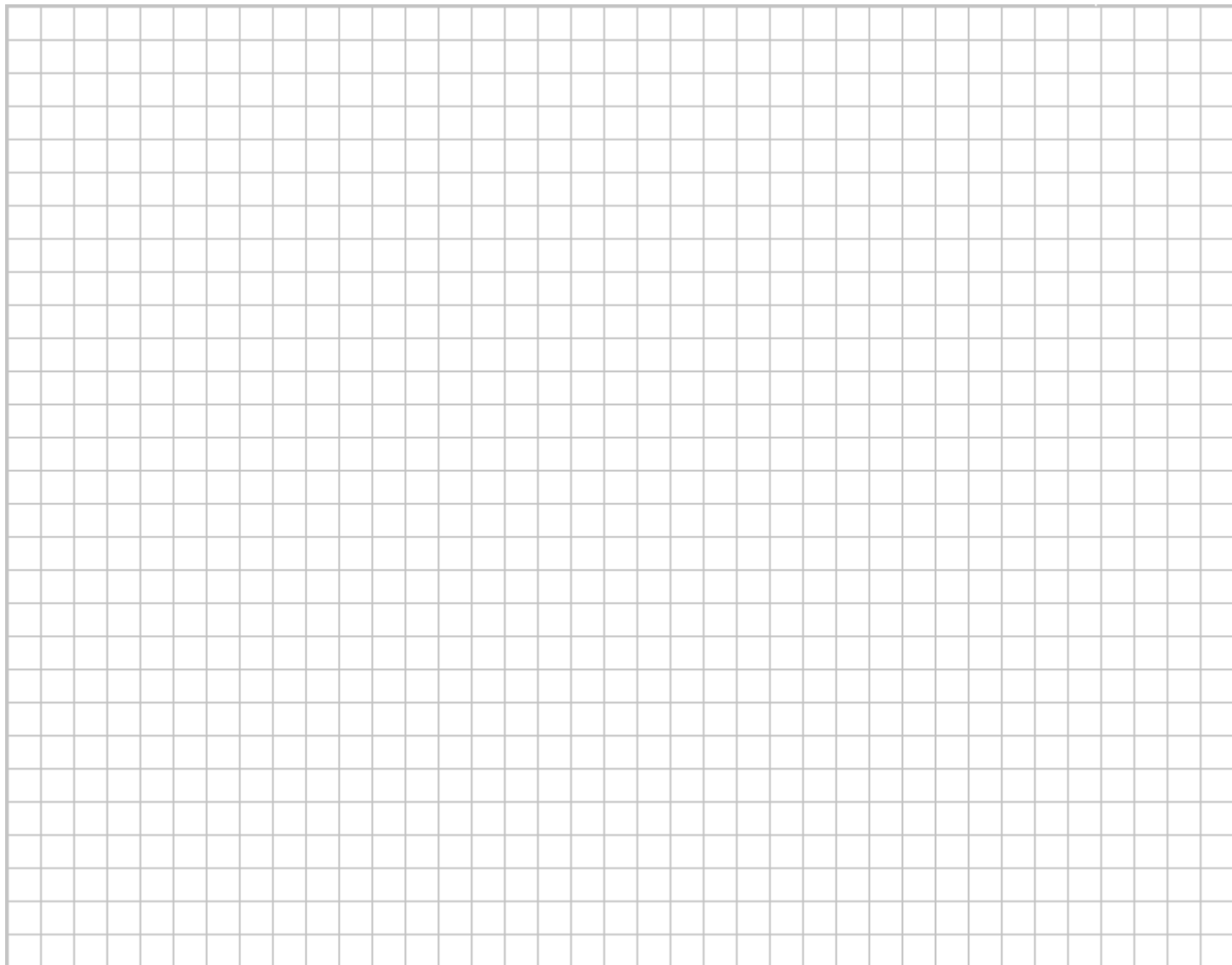
delete

delete [

// Выде

A =

for (i = 1; i < N; i++)



на матрицу памяти ление памяти

*A; new new int

A[i] = A[i-1] + M;

// Расстановка указателей:

i++
M; j++)
j;

// Указатель

*
int

// Выделение

A[0] =

// Выде

A =

под массив указателей

* [N] ;

// Работа с матрицей:

for (i = 0;

for (j =

A[i][j]

под элементы матрицы:

int [N*M] ;

for (i = 1; i < N; i++)

; N;

< j;

i;

0

delete

delete

[

< i;

;

=

// Удаление:

delete

delete

delete

[] A[0];

] A;

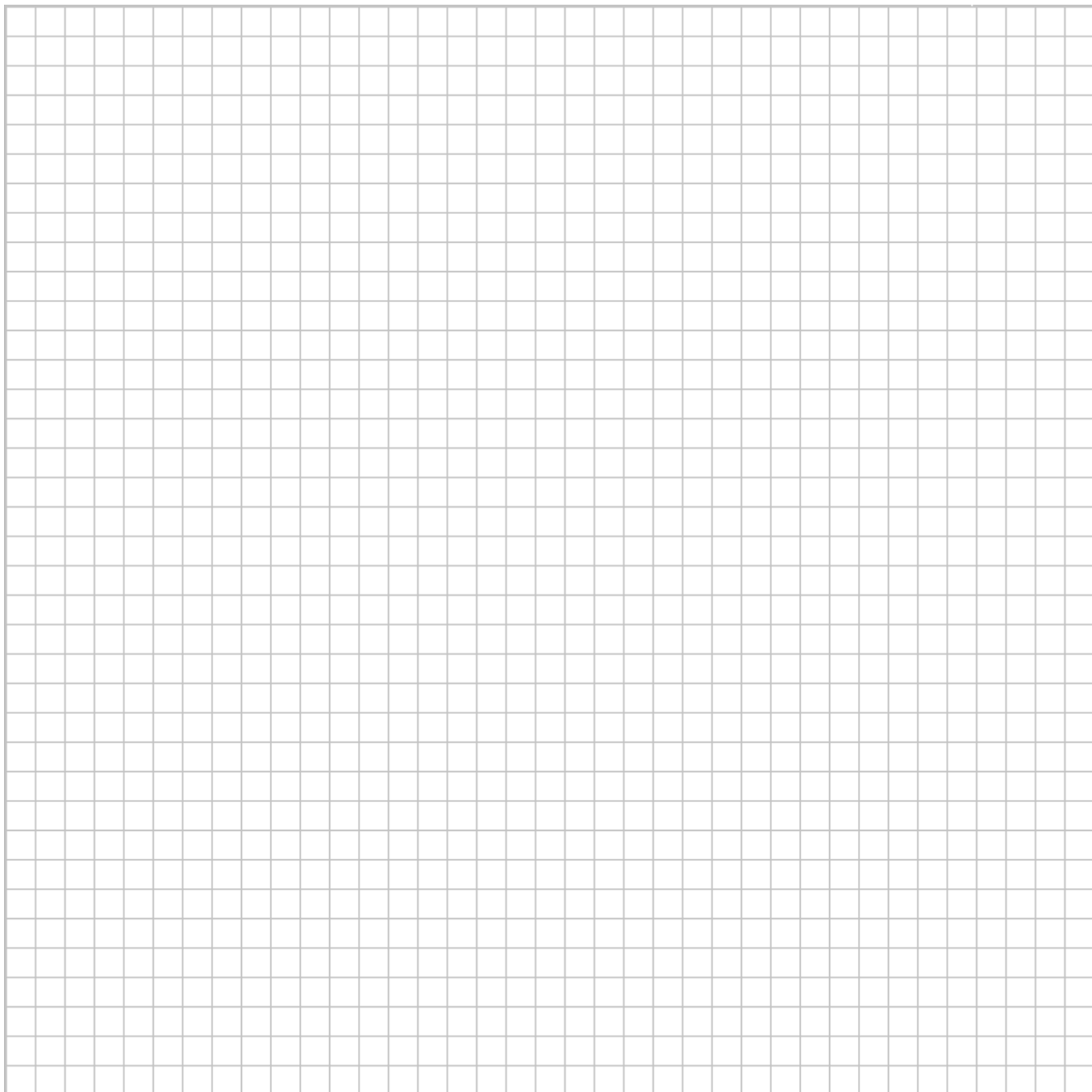
Текст программы

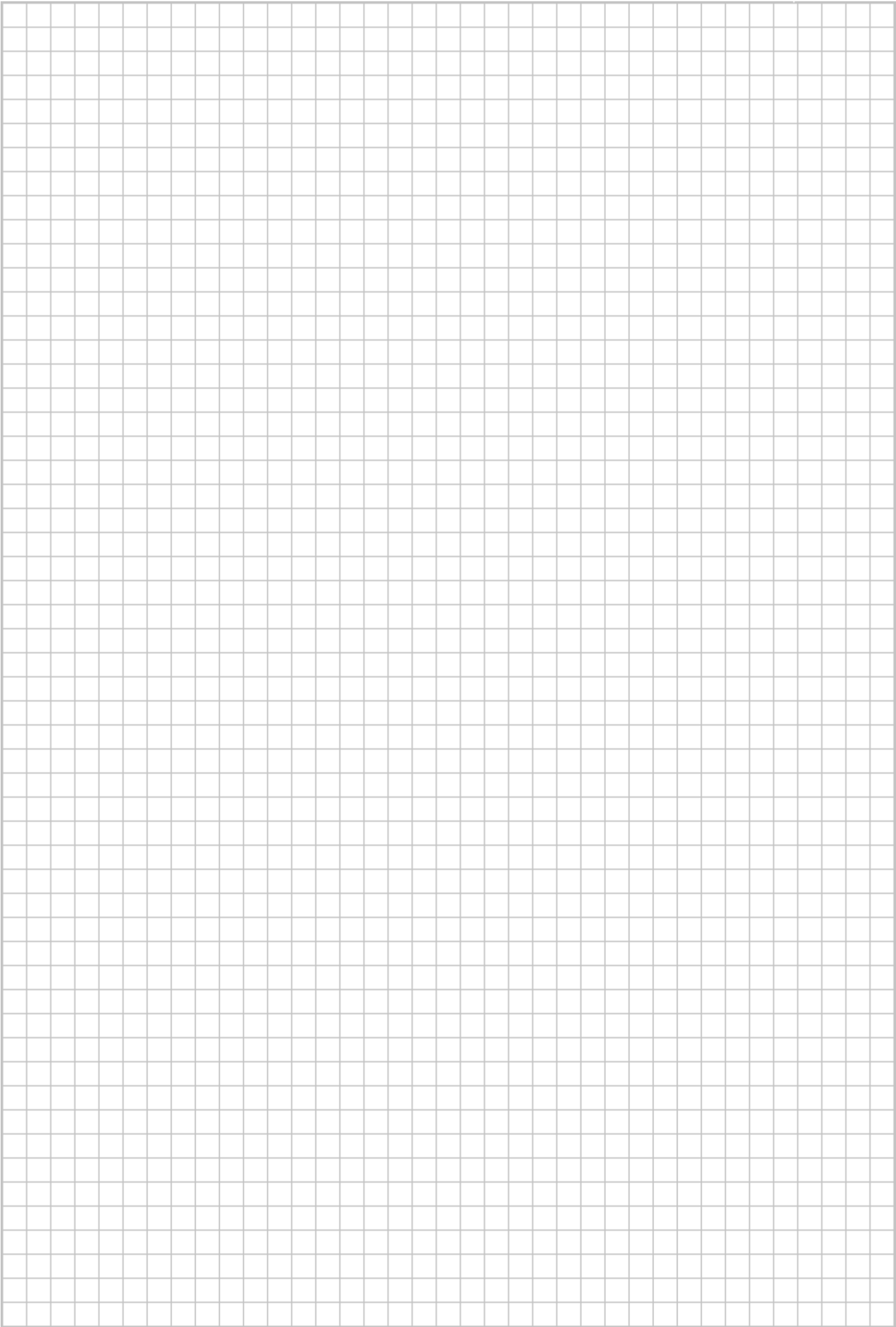
ТРЕБОВАНИЯ

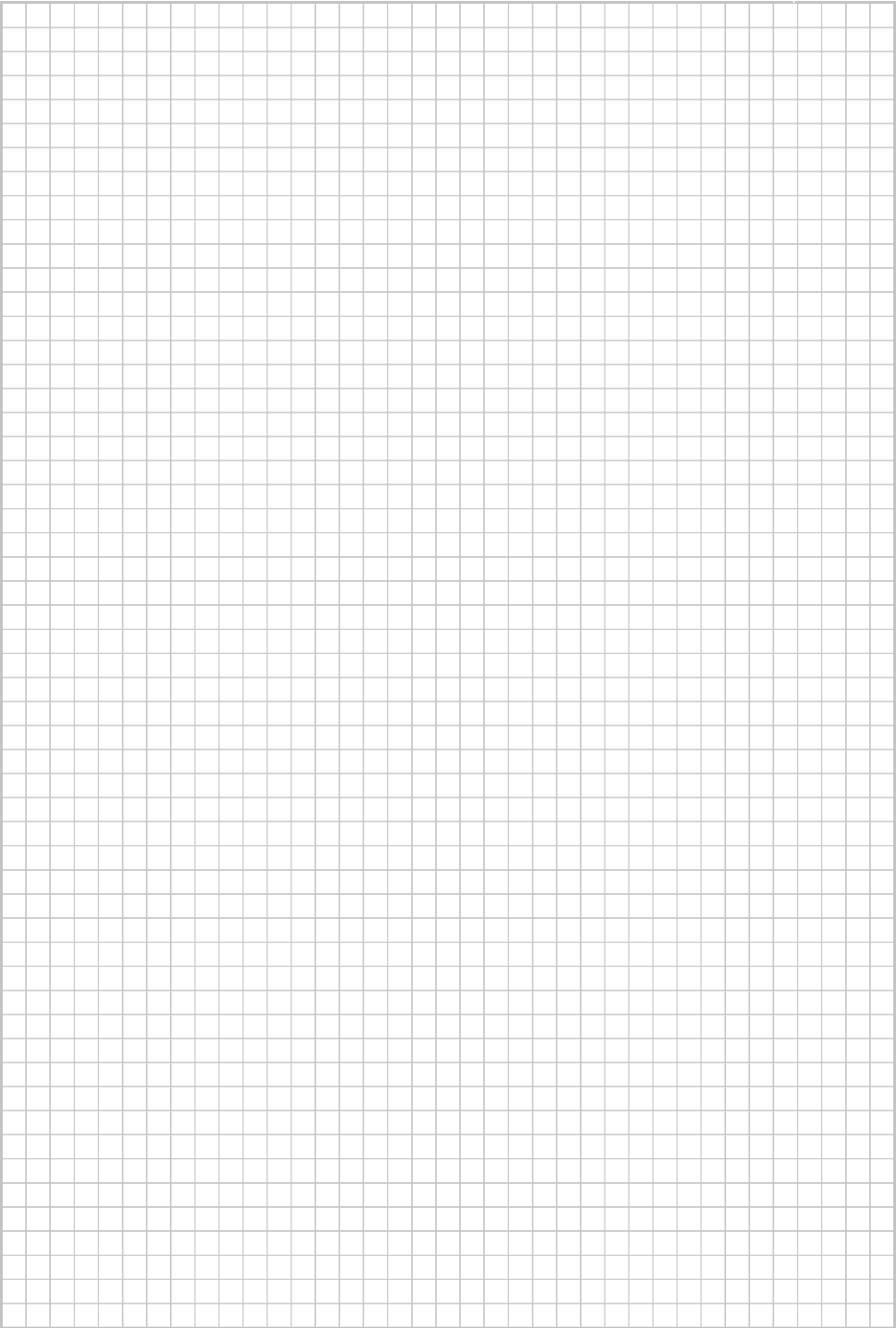
Программа должна:

- выполнять алгоритм лабораторной работы № 1;
- размер массива (количество строк и столбцов) должен вводиться с помощью клавиатуры пользователем на этапе выполнения программы;
- создавать массив динамически, используя оператор **new** и удалять его в конце работы, используя оператор **delete**;
- заполнять массив случайными числами, используя функции из стандартной библиотеки **rand**, **srand**;
- перед началом работы вывести на экран значения исходного массива, а после завершения вывести значения изменённого массива;
- состоять из подпрограмм;
- содержать в тексте определения подпрограмм до главной функции **main()**.

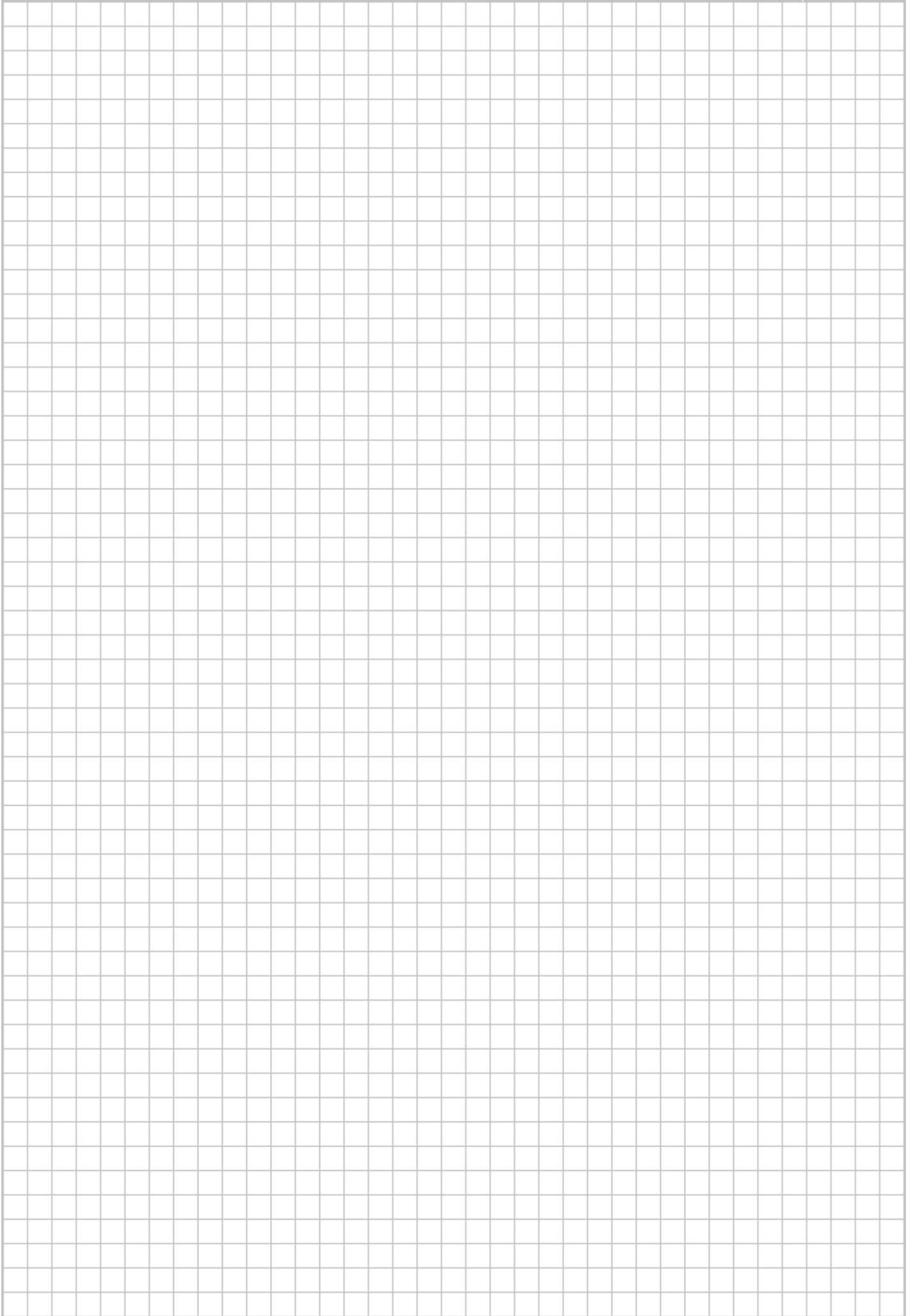
Определения подпрограмм

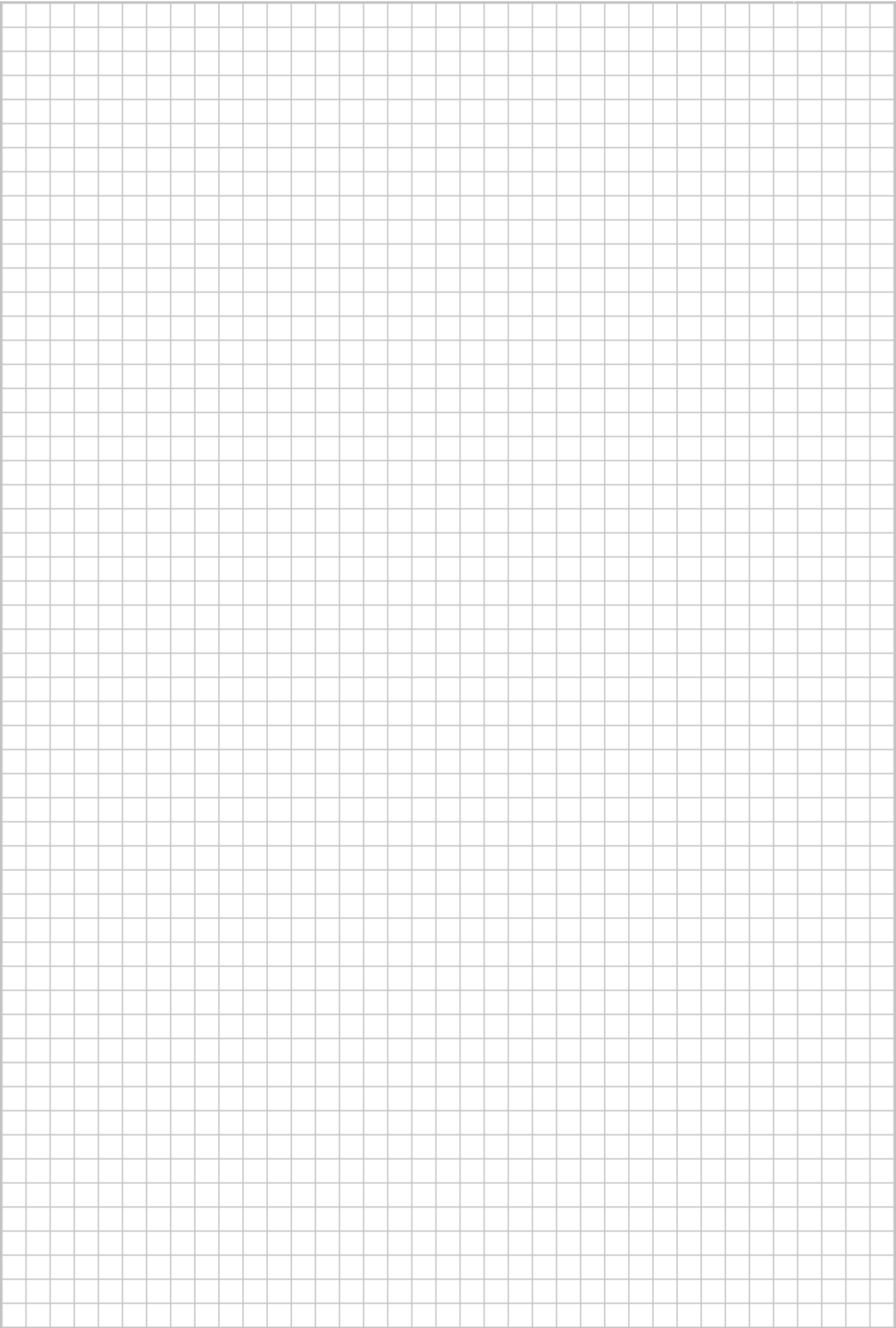




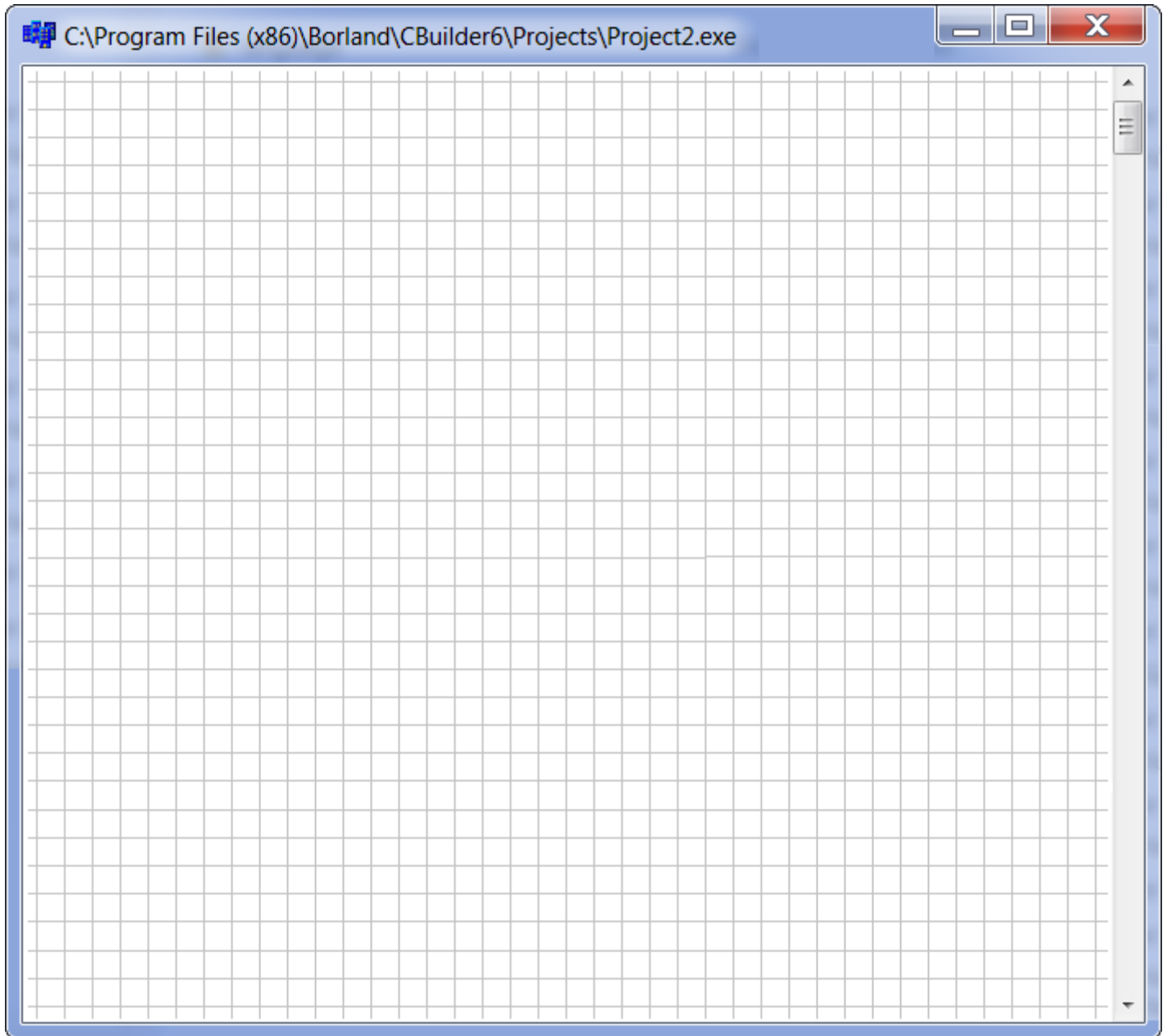


Главная функция main()





Результаты работы программы (скриншот)



Выводы по лабораторной работе № 6

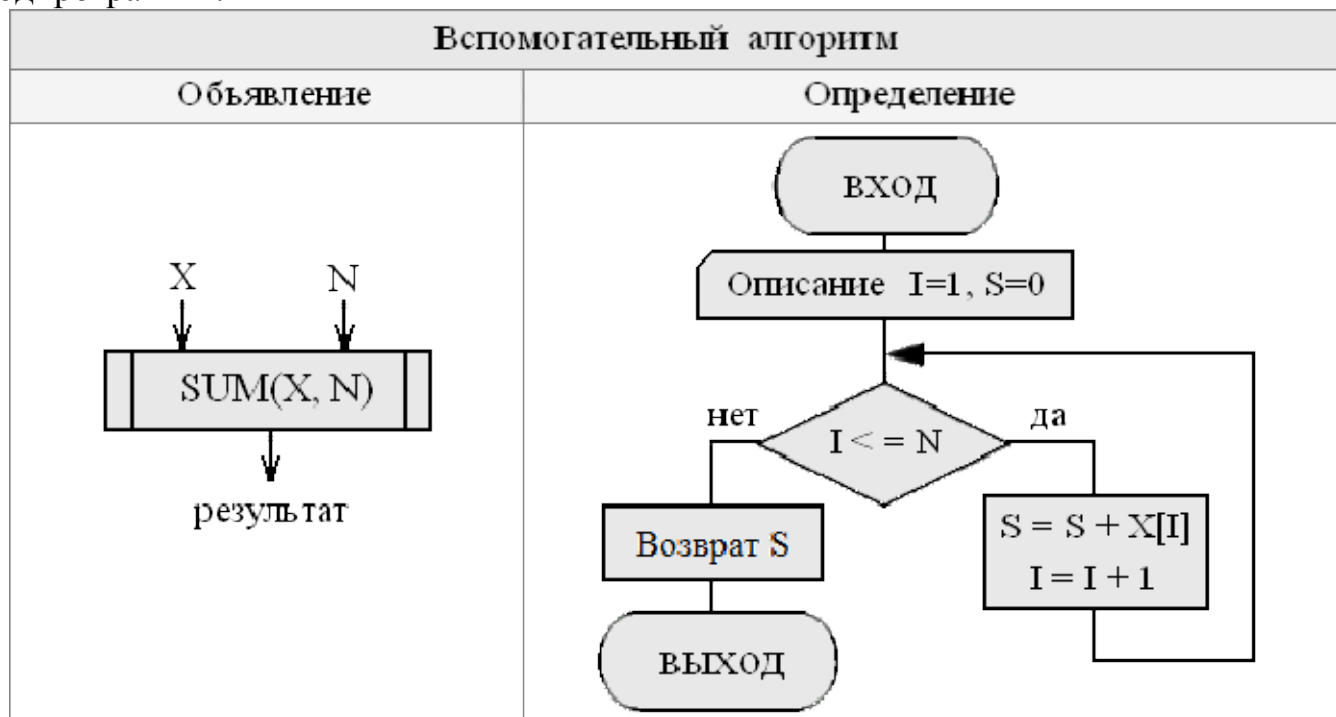
Функция возвращает значение с помощью оператора _____

Ключевое слово `const` в параметрах подпрограммы используется для _____

Прототип подпрограммы используется:

1. *компилятором* для _____,
2. *программистом* для _____.

ПРАКТИКУМ: запрограммируйте вспомогательный алгоритм на языке Си в виде подпрограммы:



Объявите подпрограмму:

Объявление	
------------	--

Определите подпрограмму:

Определение	
-------------	--

Вызовите подпрограмму из функции **main**:

Использование	
---------------	--