

1. Дано целое число  $n$  от 1 до 1000. Сгенерировать динамический массив из квадратов натуральных чисел  $1^2, 2^2, 3^2, \dots, n^2$ . Вывести данный массив.

Ввод	5
Вывод	1 4 9 16 25

2. Дано целое положительное число  $n$  от 1 до 100. Далее  $n$  целых чисел от  $-1000$  до  $1000$ . Считать данные числа в динамический массив, посчитать количество четных чисел и записать четные числа в другой динамический массив с сохранением порядка.

Ввод	5 1 -2 3 2 4
Вывод	3 -2 2 4

3. Дано целое положительное число  $n$  от 1 до 100. Далее  $n$  целых чисел от  $-1000$  до  $1000$ . Считать числа в динамический массив. Поменять первый и последний элемент массива местами. Вывести результат.

Ввод	6 11 12 13 14 15 16	1 11
Вывод	16 12 13 14 15 11	11

4. Дана последовательность целых чисел от  $-1000$  до  $1000$ , которая оканчивается нулем. Считать ее в расширяющийся массив (функция `realloc`). Описать функцию поиска суммы всех чисел.

```
int sum(int size, int *array);
```

Ввод	5 6 7 3 4 8 0
Вывод	33

5. Дано целое положительное число  $n$  от 1 до  $1000000$ . Сохранить в динамический массив (с динамическим расширением `realloc`) все числа взаимно простые с  $n$  (по возрастанию). Распечатать массив.

Ввод	12
Вывод	4 1 5 7 11

6. Даны две строки  $s_1$  и  $s_2$  заранее неизвестной длины. Считать их в динамические массивы. Сохранить их максимальный общий префикс в новый динамический массив (функция `realloc`).

```
char *prefix(char *string1, char *string2);
```

Ввод	waterfall watermelon
Вывод	water

7. Дано время начала занятий в формате чч:мм и целое положительное число  $x$  от 1 до  $24*60$  — длительность одной пары вместе с переменной в минутах. Необходимо посчитать количество пар и вывести время начала пар, если известно, что последняя пара начинается не позже 17:00. Все отрезки сохранить в динамический массив и вывести.

```
struct Time {
    short hour, minute, second;
};
void print(struct Time current);
struct Time add_time(struct Time current, int pair_time);
struct Time *generate_time(struct Time start,
                           int pair_time,
                           int *pair_count_ptr);
```

Функция `add_time(...)` добавляет к текущему времени `current` интервал в `pair_time` минут. Функция `generate_time(...)` генерирует динамический массив с отрезками, возвращая через указатель `pair_count_ptr` количество отрезков.

Ввод	09:00 90
Вывод	6 09:00 10:30 12:00 13:30 15:00 16:30

8. Дано целое положительное число  $n$ . Далее матрица целых чисел  $n \times n$ . Считать элементы в динамическую матрицу. Найти сумму элементов каждой строки матрицы и произведение элементов каждого столбца матрицы.

Ввод	3 1 2 3 4 5 6 7 8 9
Вывод	6 15 24 28 80 162

9. Дано целое положительное число  $n$ . Далее матрица целых чисел  $n \times n$ . Считать элементы в динамическую матрицу. Переставить строки матрицы в обратном порядке (без какого либо копирования элементов, только изменив адреса строк).

Ввод	4 1 2 3 4 2 3 4 5 3 4 5 6 4 5 6 7
Вывод	4 5 6 7 3 4 5 6 2 3 4 5 1 2 3 4

10. Дано целые положительные числа  $n$  и  $m$ . Далее матрица целых чисел  $n \times m$ . Считать элементы в динамическую матрицу. Вывести транспонированную матрицу. Описать вспомогательную функцию, которая генерируют транспонированную матрицу:

```
int **transpose(int n, int m, int **matrix);
```

Ввод	3 4 1 2 3 4 5 6 7 8 9 10 11 12
Вывод	1 5 9 2 6 10 3 7 11 4 8 12