

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. Всё реализовать в основной функции `main()`.

Ввод	5
Вывод	1 4 6 9 25

2. Дано целое положительное число n от 1 до 100. Далее n целых чисел от -1000 до 1000 . Добавить чётные числа в динамический расширяющийся массив. Вывести количество чётных чисел и сами числа. Всё реализовать в основной функции `main()`.

Ввод	5 1 -2 8 2 4
Вывод	4 4 2 8 -2

3. Дано целое положительное число n от 1 до 100. Далее n целых чисел от -1000 до 1000 . Считать данные числа в динамический массив и вывести их.

```
void scan(int *size_ptr, int **array_ptr);  
void print(int size, int *array);
```

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

4. Дано целое положительное число n от 1 до 100. Далее n целых чисел от -1000 до 1000 . Поменять первый и последний элемент массива местами.

```
void scan(int *size_ptr, int **array_ptr);  
void exchange(int size, int *array);  
void print(int size, int *array);
```

Ввод	6 11 12 13 14 15 16	3 11 12 13
Вывод	16 12 13 14 15 11	13 12 11

5. Дано целое положительное число n от 1 до 100. Далее n целых чисел от -1000 до 1000 . Считать данные числа в динамический массив. Скопировать чётные числа в другой массив. Вывести количество чётных чисел и сами числа. Описать функции:

```
void scan(int *size_ptr, int **array_ptr);  
void filter(int src_size, int *src_array,  
            int *dst_size_ptr, int **dst_array_ptr);  
void print(int size, int *array);
```

Ввод	5 1 -2 8 2 4
Вывод	4 4 2 8 -2

6. Дана последовательность целых чисел от -1000 до 1000 , которая оканчивается нулем. Считать ее в расширяющийся буфер. Описать функцию поиска суммы всех чисел.

```
void scan(int *size_ptr, int **array_ptr);  
int sum(int size, int *array);
```

Ввод	5 6 7 3 4 8 0
Вывод	33

7. Дано целое положительное число n от 1 до 1000000 . Сохранить в динамический массив все делители числа n (по возрастанию). Распечатать массив.

```
void filter(int n,
           int *dividers_counter_ptr,
           int **dividers_array_ptr);
void print(int size, int *array);
```

Ввод	12
Вывод	1 2 3 4 6 12

8. Дана последовательность слов (заранее неизвестной длины), разделенных пробелом, которая заканчивается переносом строки. Распечатать каждое слово задом наперед (каждое слово считывать в динамический массив).

Ввод	Are you ready kids Aye Aye Captain
Вывод	erA erA ydaer sdik eyA eyA niatpaC

```
char *get_word(int *word_size_ptr);
void print_reversed_word(int word_size, char *word);
```

9. 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.

```
void scan(int **string_ptr);
char *prefix(char *string1, char *string2);
```

Ввод	waterfall watermelon
Вывод	water

10. Дано время начало 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 отсечки 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 отсечками, возвращая через указатель `pair_count_ptr` количество отсечек.

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