

Выделите память для массива чисел и заполнить его любым способом. Создайте функцию, выполняющую задачу, указанную в варианте. Вызовите её из основной функции, получите её результат и выведите его.

Вариант	Функция
1.	Получить индексы всех вхождений максимума в массив
2.	Во входном массиве – целые числа. Поместить в массив на выходе только простые числа из него
3.	Получить все индексы вхождения некоторой переменной <i>element</i>
4.	Входной массив содержит 4-значные целые числа вида <i>abcd</i> . Поместить в массив-результат только те, у которых $ab=cd$
5.	Выделить элементы <i>z</i> , подчиняющиеся выражению: $x^2 + y^2 = z^2$, <i>x</i> , <i>y</i> - целые
6.	Выделить все отсортированные по возрастанию участки входного массива и записать в результирующий массив пары (<i>i</i> , <i>len</i>), где <i>i</i> – индекс начала участка, <i>len</i> – его длина
7.	Получить количество элементов, совпадающих со средним арифметическим в массиве
8.	Считать входной массив компонентами векторов размерности <i>n</i> (размер массива кратен <i>n</i>). Получить среднюю норму вектора
9.	Во входном массиве – целые числа. Произвести арифметический сдвиг на <i>n</i> позиций влево (вправо) каждого элемента
10.	Удалить все нечётные числа в массиве
11.	Входной массив содержит 4-значные целые числа <i>abcd</i> . Поместить в выходной массив только те, у которых $a+b=c+d$
12.	Во входном массиве находятся коэффициенты поверхности второго порядка. Вывести тип поверхности.
13.	Вывести все элементы массива, удовлетворяющие выражению $3k+1$, где <i>k</i> – целое.
14.	Считать входной массив компонентами векторов размерности <i>n</i> (размер массива кратен <i>n</i>). Получить среднее значение угла между соседними парами векторов.
15.	Выделить все элементы, отличающиеся от среднего арифметического в массиве не более, чем на переменную <i>diff</i>
16.	Входной массив содержит 4-значные целые числа <i>abcd</i> . Поместить в выходной массив только те, у которых <i>a+c</i> делится на 3, <i>b+d</i> делится на 7
17.	Входной массив содержит 5-значные целые числа <i>abcde</i> . Поместить в выходной массив только те, у которых $a+b=d+e=c$
18.	Получить количество элементов, больших, чем среднее геометрическое в массиве
19.	Входной массив содержит 5-значные целые числа <i>abcde</i> . Поместить в выходной массив только те, у которых $a*b=d*e=c$
20.	Получить количество элементов, больших, чем среднее геометрическое в массиве
21.	Получить количество элементов, меньших, чем среднее гармоническое в массиве
22.	Получить количество элементов, расположенных между первым максимальным и последним минимальным элементами массива. Если таких элементов нет, то вывести -1.
23.	Получить количество элементов, расположенных между последним максимальным и первым минимальным по модулю элементами массива. Если

	таких элементов нет, то вывести -1.
24.	Получить сумму элементов, расположенных между первым и последним нулевыми элементами (если, конечно, массив содержит не менее 2-х нулей), и значение -1 (минус 1), если в массиве нулей меньше 2
25.	Получить индексы всех максимальных элементов массива
26.	Вывести все элементы, которые встречаются больше двух раз
27.	Все повторяющиеся элементы (дубли) заменить средним арифметическим исходного массива
28.	Заменить все элементы массива средним арифметическим их соседей, крайние элементы оставить нетронутыми
29.	Переставить элементы массива, которые меньше входного параметра n в начало массива, остальные разместить в конце (порядок не важен)
30.	Отразить зеркально элементы массива относительно середины, но только для тех симметричных элементов, у которых совпадает четность
31.	Дан массив, где каждая пара элементов - это координаты точки (x,y) на плоскости. Найти пары точек, которые симметричны относительно оси OY.
32.	Дан массив целых чисел. Получить сумму индексов элементов, кратных трем.
33.	Сдвинуть элементы массива вправо на величину, равную количеству отрицательных элементов массива. "Пустые" элементы слева заполнить нулями
34.	Нормализовать элементы массива по следующей формуле: $\text{new} = (\text{old} - \text{min}) / (\text{max} - \text{min})$ <p>max - значение максимального элемента</p> <p>min – значение минимального элемента</p>
35.	Определить максимальную длину последовательности упорядоченных элементов (последовательность может возрастать или убывать, учесть оба варианта)
36.	Отразить зеркально участки, находящиеся между отрицательными элементами

Пример кода (MS Visual Studio 2010):

```
#include <stdio.h>
#include <stdlib.h>

int get_min(int* arr, int len)
{
    //на вход функция получает массив и его длину
    //переменной min надо дать какое-то стартовое значение: первый элемент
    //массива не с чем сравнить
    int min = arr[0];
    int i = 0;
```

```

        //обработку начинаем со второго элемента
        for(i=1;i<len;i++)
        {
            if(arr[i]<min)
                min = arr[i];
        }
        return min;
    }

int my_sum(int* a, int len)
{
    //функция определяет сумму нечётных чисел массива
    int sum = 0;
    int i=0;
    for(i=0;i<len;i++)
    {
        if(a[i]%2!=0)
            sum = sum + a[i];
    }
    return sum;
}

int* invert_arr(int* a, int len)
{
    //Функция меняет значения элементов на противоположные и возвращает новый
    массив
    //выделяем память для нового массива - размер равен входному
    int* inv_a = (int*)malloc(sizeof(int)*len);
    int i=0;
    for(i=0;i<len;i++)
    {
        inv_a[i]=-a[i];
    }
    return inv_a;
}

int _tmain(int argc, _TCHAR* argv[])
{
    //выделение памяти под массив из 10 элементов типа int
    int* a = (int*)malloc(sizeof(int)*10);
    printf("\n");
    int i=0;
    //заполнение массива
    for(;i<10;i++)
    {
        a[i] = i*i - 2*i;
        printf("%d ", a[i]);
    }
    int m = get_min(a, 10);
    printf("\nMin = %d\n", m);
    //сумма нечётных чисел
    int sum = 0;
    sum = my_sum(a, 10);
    printf("\nSum = %d", sum);

    //Получаем инвертированный массив
    int* ia = invert_arr(a, 10);

    //Выводим его
    printf("\n");
    for(i=0;i<10;i++)
    {
        printf("%d ", ia[i]);
    }
}

```

```
        //освобождение памяти
        free(a);
        free(ia);

        return 0;
    }
```

Пример функции main в других средах разработки:

```
int main()
{
    //ваш код
    return 0;
}
```