

5. Что нужно изменить в программе 5.1, чтобы выдавался не последний из найденных элементов, а первый?
6. Измените условие цикла `while` примера 5.3 так, чтобы использовалась логическая операция `not`.
- 2 Рост учащихся класса представлен в виде массива. Определите количество учащихся, рост которых больше среднего роста по классу.
- 3 Заданы фамилии и рост учащихся 10-го класса. Вывести фамилии тех учащихся, рост которых меньше среднего роста по классу.
- 4 Известны данные о площади  $n$  стран (в млн кв. км) и численности населения (в млн жителей). Выведите номера тех стран, плотность населения которых больше  $x$ .
- 5 Для упражнения 4 добавьте возможность вводить и выводить названия стран.
- 6 Определите, есть ли в линейном массиве хотя бы один элемент, который является нечетным числом, кратным 7. Если да, то следует вывести его номер.
- 7\* В линейном массиве найдите и выведите все простые числа с нечетной суммой цифр. Укажите, сколько чисел вывели.
- 8\* В линейном массиве найдите и выведите все числа Армстронга. (Числом Армстронга называется такое число, которое равно сумме своих цифр, возведенных в степень, равную количеству его цифр. Например, числом Армстронга является число  $371 : 371 = 3^3 + 7^3 + 1^3 = 27 + 343 + 1$ .) Укажите, сколько чисел вывели.
- 9 Задан одномерный массив из  $n$  строк. Каждая строка является предложением из слов, разделенных пробелами. Найдите и выведите те предложения, в которых есть слова, начинающиеся на гласную (строчную или прописную).

## § 6. Максимальный и минимальный элементы массива

Пример 6.1.  
V. Программа:

```
var a: array[1..20] of integer;
n, max: integer;
begin
  write('Количество n = ');
  readln(n);
  writeln('Элементы массива');
  for var i := 1 to n do
    read(a[i]);
  max := a[1];
  for var i := 2 to n do
    if a[i] > max then
      max := a[i];
  writeln('Максимум = ', max);
end.
```

### 6.1. Поиск максимального (минимального) элемента в массиве

Очень часто для решения задачи требуется находить не заданный элемент массива, а максимальный (наибольший) или минимальный (наименьший).

Рассмотрим задачу нахождения максимального элемента. Если в массиве один-единственный элемент, то он и есть максимальный. Если элементов больше одного, то максимальным в массиве из  $i$  элементов является максимум из  $a[i]$  и максимального

среди первых  $i-1$  элементов. Находить максимум будем последовательно, сравнивая текущий элемент с максимумом, найденным на предыдущем шаге. Если текущий элемент больше, то значение максимума, найденное на предыдущем шаге, нужно обновить (пример 6.1).

Данный алгоритм находит значение максимального элемента, но не позволяет определить, на каком месте в массиве расположен этот максимальный элемент.

Будем использовать переменную `n_max` для хранения индекса максимального элемента. Значение переменной `n_max` будет изменяться тогда, когда изменяется значение максимального элемента (пример 6.2).

Если в массиве несколько элементов имеют максимальное значение, то значением переменной `n_max` будет индекс первого из них. Если использовать условие  $a[i] \geq \max$ , то переменная `n_max` будет хранить индекс последнего из максимальных элементов.

В случае, когда известен индекс  $i$  элемента массива, значение элемента можно получить, обратившись к элементу по индексу:  $a[i]$ . Поэтому при поиске максимального элемента достаточно хранить только его индекс `n_max`. Значение максимального элемента —  $a[n\_max]$  (пример 6.3).

Для поиска минимального элемента необходимо заменить знак  $>$  в условии оператора ветвления на знак  $<$  (пример 6.4).

**Пример 6.1.** Продолжение.

VI. Тестирование.

```
Окно вывода
Количество n = 5
Элементы массива
1 2 6 3 2
Максимум = 6
```

**Пример 6.2.**

V. Программа:

```
var a: array[1..20] of integer;
    n, max, n_max: integer;
begin
  write('Количество n =');
  readln(n);
  writeln('Элементы массива');
  for var i := 1 to n do
    read(a[i]);
  max := a[1]; n_max := 1;
  for var i := 2 to n do
    if a[i] > max then
      begin
        max := a[i]; n_max := i;
      end;
  writeln('Максимум = ', max);
  writeln('Его место ', n_max);
end.
```

VI. Тестирование.

```
Окно вывода
Количество n = 5
Элементы массива
2 5 3 5 1
Максимум = 5
Его место 2
```

**Пример 6.3.** Фрагмент программы:

```
n_max := 1;
for var i := 2 to n do
  if a[i] > a[n_max] then
    n_max := i;
writeln('Максимум =', a[n_max]);
writeln('Его место ', n_max);
```

**Пример 6.4.** Фрагмент программы:

```
n_min := 1;
for var i := 2 to n do
  if a[i] < a[n_min] then
    n_min := i;
```

**Пример 6.5.**

V. Программа:

```

var a: array [1..20] of real;
    n, n_min: integer;
begin
  writeln('Количество
           спортсменов');
  readln(n); writeln('Время');
  for var i := 1 to n do
    read(a[i]);
  //поиск минимального элемента
  n_min := 1;
  for var i := 2 to n do
    if a[i] < a[n_min] then
      n_min := i;
  writeln('Победитель - лыжник
           номер ', n_min);
  writeln('Его время - ', a[n_min]);
end.

```

VI. Тестирование.

Окно вывода

```

Количество спортсменов
5
Время
6.31 6.17 7.32 6.54 7.03
Победитель - лыжник номер 2
Его время - 6.17

```

**Пример 6.6.**

V. Программа:

```

var a: array [1..20] of integer;
    n, min, k: integer;
begin
  write('Количество n =');
  readln(n); writeln('Числа');
  for var i := 1 to n do
    read(a[i]);
  //поиск минимального элемента
  min := a[1];
  for var i := 2 to n do
    if a[i] < min then
      min := a[i];
  //подсчет количества
  k := 0;
  for var i := 1 to n do
    if a[i] = min then
      k := k + 1;
  writeln('Минимальный ', min);
  writeln('Встретился ', k,
          ' раз(-a)');
end.

```

**6.2. Решение задач с использованием алгоритма поиска максимального (минимального) элемента**

**Пример 6.5.** В массиве хранится информация о результатах спортсменов, участвующих в лыжной гонке. Определить результат победителя и его номер.

I. Исходные данные: массив  $a$  — числа, являющиеся временем прохождения трассы, количество спортсменов —  $n$ .

II. Результат:  $a[n\_min]$  — минимальное время,  $n\_min$  — номер победителя.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Для решения задачи воспользуемся алгоритмом поиска минимального элемента в массиве и его номера (пример 6.4).

3. Вывод результата.

IV. Описание переменных:  $a$  — `array[1..20] of real`;  $n, n\_min$  — `integer`.

**Пример 6.6.** Определить, сколько раз в линейном массиве встречается элемент, равный минимальному.

I. Исходные данные: массив  $a$ , количество чисел  $n$ .

II. Результат:  $min$  — минимальный элемент,  $k$  — количество минимальных.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Поиск минимального элемента.

3. Линейный поиск элементов, равных минимальному.

4. Вывод результата.

IV. Описание переменных: `a` — `array[1..20] of integer`; `n`, `min`, `k` — `integer`.

**Пример 6.7.** Задан массив из слов различной длины. Найти в нем самое длинное и самое короткое слово.

I. Исходные данные: массив `a`, количество слов `n`.

II. Результат: `min_s` — самое короткое слово, `max_s` — самое длинное слово.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Поиск самого короткого слова. Самое короткое слово — слово, в котором минимальное количество символов. Для его поиска можно воспользоваться алгоритмом поиска минимального элемента в массиве. Однако, если сравнивать сами элементы массива, то сравнение будет происходить не по длине<sup>1</sup>. Для сравнения строк по длине нужно использовать функцию вычисления длины строки `length`.

3. Для поиска самого длинного слова можно использовать алгоритм поиска максимального элемента и сравнивать элементы с использованием функции вычисления длины строки `length`.

4. Вывод результата.

IV. Описание переменных: `a` — `array[1..20] of string`; `n` — `integer`; `min_s`, `max_s`: `string`;

**Пример 6.6.** Продолжение.

VI. Тестирование.

```
Окно вывода
Количество n = 5
Числа
3 1 2 1 1
Минимальный 1
Встретился 3 раза (-a)
```

**Пример 6.7.**

V. Программа:

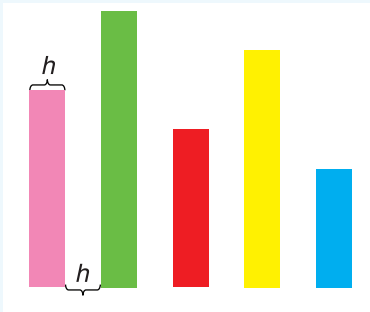
```
var a: array [1..20] of string;
    n: integer;
    min_s, max_s: string;
begin
  write('Количество n =');
  readln(n); writeln('Слова');
  for var i := 1 to n do
    readln(a[i]);
    //поиск короткого слова
    min_s := a[1];
  for var i := 2 to n do
    if length(a[i]) < length(min_s)
    then
      min_s := a[i];
    //поиск длинного слова
    max_s := a[1];
  for var i := 2 to n do
    if length(a[i]) > length(max_s)
    then
      max_s := a[i];
  writeln('Короткое - ',min_s);
  writeln('Длинное - ',max_s);
end.
```

VI. Тестирование.

```
Окно вывода
Количество n = 5
Слова
Все
дороги
ведут
в
Рим
Короткое - в
Длинное - дороги
```

<sup>1</sup> Сравнение строк осуществляется лексикографически:  $s_1 < s_2$ , если для первого несовпадающего символа с номером  $i$  верно неравенство  $s_1[i] < s_2[i]$  или все символы строк совпадают, но  $s_1$  короче  $s_2$ .

## Пример 6.8.



## V. Программа:

```

uses graphABC;
var a: array[1..20] of integer;
    n, max, h, x, y1, y2: integer;
    m: real;
begin
  write('Количество n =');
  readln(n);
  writeln(n);
  writeln('Элементы массива');
  for var i := 1 to n do
  begin
    read(a[i]);
    write(a[i], ' ');
  end;
  max := a[1];
  for var i := 2 to n do
  if a[i] > max then
    max := a[i];
  h := trunc(WindowWidth/(2*n+1));
  m := WindowHeight/max;
  x := h;
  for var i := 1 to n do
  begin
    SetBrushColor(clrandom);
    y1 := WindowHeight;
    y2 := y1 - trunc(a[i]*m);
    Rectangle(x, y1, x+h, y2);
    x := x + 2*h;
  end;
end.

```

## 6.3. Построение гистограммы (столбчатой диаграммы)

**Пример 6.8.** Дан одномерный массив из целых чисел. Построить гистограмму по числовым данным, хранящимся в массиве.

I. Исходные данные: массив  $a$ , количество чисел  $n$ .

II. Результат: построенная диаграмма.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Гистограмма состоит из  $n$  прямоугольников одинаковой ширины. Элементы массива определяют высоту соответствующего прямоугольника. Максимальное значение элементов массива (переменная  $max$ ) должно по высоте поместиться в окне ( $WindowHeight$ ).

Обозначим  $m = \frac{WindowHeight}{max}$  (масштабный коэффициент). Тогда

значению элемента массива  $a[i]$  будет соответствовать целая часть от величины  $a[i] * m$ .

3. Находим максимальный элемент массива.

4. В цикле строим прямоугольники. Все прямоугольники имеют одинаковую ширину ( $h$ ), расстояние между ними можно определить равным ширине прямоугольника. Тогда ширина прямоугольника — целая часть от деления ширины окна на  $(2n + 1)$ :

$$h = \frac{WindowWidth}{2n + 1}.$$

5. При вычислении высоты прямоугольника нужно учесть то, что ось Y направлена сверху вниз.

6. Цвет прямоугольника будем задавать случайным образом.

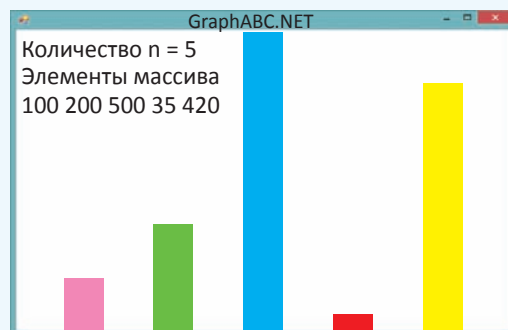
7. Местоположение прямоугольника определяется переменной  $x$ . Начальное значение  $x = h$ . Новое значение получается из предыдущего увеличением на  $2 \cdot h$ .

8. Вывод результата.

IV. Описание переменных:  $a$  — `array[1..20] of integer`;  $n$ ,  $max$ ,  $h$ ,  $x$ ,  $y1$ ,  $y2$  — `integer`;  $m$ : `real`.

**Пример 6.8. Продолжение.**

VI. Тестирование.



VII. Постройте по этим данным диаграмму в Excel и сравните.



1. Какой элемент массива является максимальным? Какой — минимальным?
2. Как найти максимальный элемент в массиве?
3. Как найти минимальный элемент?
4. Каким образом определить номер первого элемента, равного максимальному?
5. Как определить номер последнего элемента, равного минимальному?



### Упражнения

- 1 Измените программы примеров 6.1 и 6.2 так, чтобы находился минимальный элемент в массиве.
- 2 Для примера 6.5 выполните перечисленные задания.
  1. Найдите номер спортсмена, пришедшего на финиш последним.
  2. Определите, был ли победитель единственным или есть еще лыжник, прошедший трассу с таким же результатом (см. пример 6.6).
  3. Добавьте еще один массив. Введите в него фамилии спортсменов. Реализуйте пункты 1—3 так, чтобы выводилась фамилия, а не номер (см. пример 5.10).
- 3 В массиве хранится информация о стоимости автомобилей. Определите стоимость самого дорогого автомобиля и его номер в массиве. Если есть несколько таких автомобилей, то выведите все номера.
- 4 В массиве хранится информация о среднесуточной температуре декабря. Определите, сколько в декабре было дней с самой низкой и с самой высокой температурой.
- 5 Задан массив из слов. Найдите в нем самое длинное слово, заканчивающееся буквой  $a$ .
- 6\* Задан массив из слов. Найдите в нем самое короткое слово, начинающееся с прописной буквы.



- 7 Для примера 6.8 выполните перечисленные задания.
1. Измените программу так, чтобы при построении диаграммы использовалась не вся высота окна, а оставались поля сверху и снизу.
  2. Измените программу так, чтобы столбики строились без промежутков между ними.
  3. Создайте массив цветowych констант и используйте эти цвета для закрашивания столбиков.
  4. Постройте линейчатую диаграмму.
- 8 По данным массива постройте диаграмму в виде ломаной линии. Соответствуют ли ординаты вершин ломаной значениям массива?

## § 7. Преобразование элементов массива

### Пример 7.1.

V. Программа:

```
var a: array[1..20] of integer;
    n: integer;
begin
  write('Количество n =');
  readln(n);
  writeln('Элементы массива');
  for var i := 1 to n do
    read(a[i]);
  for var i := 1 to n do
  begin
    if a[i] > 0 then
      a[i] := a[i] * 2;
    if a[i] < 0 then
      a[i] := a[i] + 5;
  end;
  writeln('Преобразованный массив');
  for var i := 1 to n do
    write(a[i], ' ');
end.
```

VI. Тестирование.

#### Окно вывода

```
Количество n = 5
Элементы массива
3 -2 0 -1 5
Преобразованный массив
6 3 0 4 10
```

VII. Анализ результатов. Элементы 3 и 5 увеличены в 2 раза, элементы -2 и -1 увеличены на 5, элемент 0 остался неизменным.

### 7.1. Основные задачи

Среди задач преобразования элементов массива можно выделить задачи следующих типов:

1. Изменение элементов массива в зависимости от условий.
  2. Обмен местами элементов массива.
  3. Удаление элемента из массива.
  4. Вставка элемента в массив.
- Рассмотрим каждую из задач.

### 7.2. Изменение элементов массива в зависимости от выполнения некоторых условий

**Пример 7.1.** Задан одномерный массив целых чисел. Преобразовать его элементы по следующему правилу: положительные элементы увеличить в 2 раза, а отрицательные — увеличить на 5.

- I. Исходные данные: одномерный массив  $a$ , количество элементов  $n$ .
- II. Результат: преобразованный массив  $a$ .

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. В цикле проверяем текущий элемент. Если он положительный,