

АЛГОхитрости

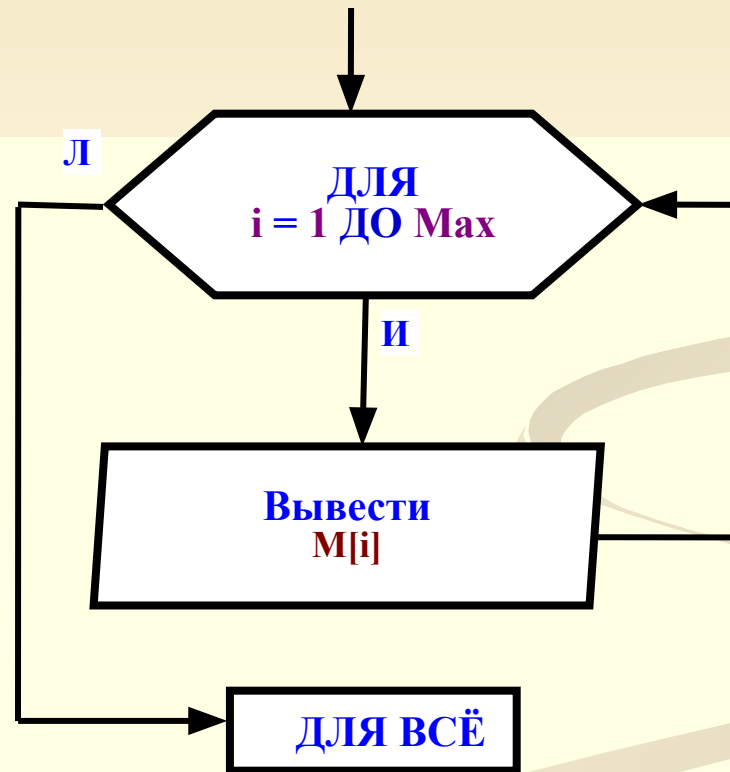
Типовые алгоритмические приёмы работы с массивами

1. Считывание содержимого одномерного массива.
2. Сумма элементов одномерного массива.
3. Среднее значение элементов одномерного массива.
4. Наибольший (наименьший) элемент одномерного массива.
5. Диапазон значений элементов одномерного массива.
6. Определение является ли число палиндромом.
7. Работа с приращениями нескольких массивов в одном цикле: $a[i++]$, $b[j++]$, $c[k++]$, $g[f--]$.
8. Ввод и вывод содержимого двумерного массива.
9. Сумма элементов двумерного массива.

Типовые алгоритмические приёмы с массивами

Одномерные массивы. Считывание содержимого массива

Считывание – вывод на экран, в файл, на печать содержимого массива – реализуется обычно с использованием простого цикла **ДЛЯ** начиная с первого элемента до конца массива. **Max** – размерность массива.

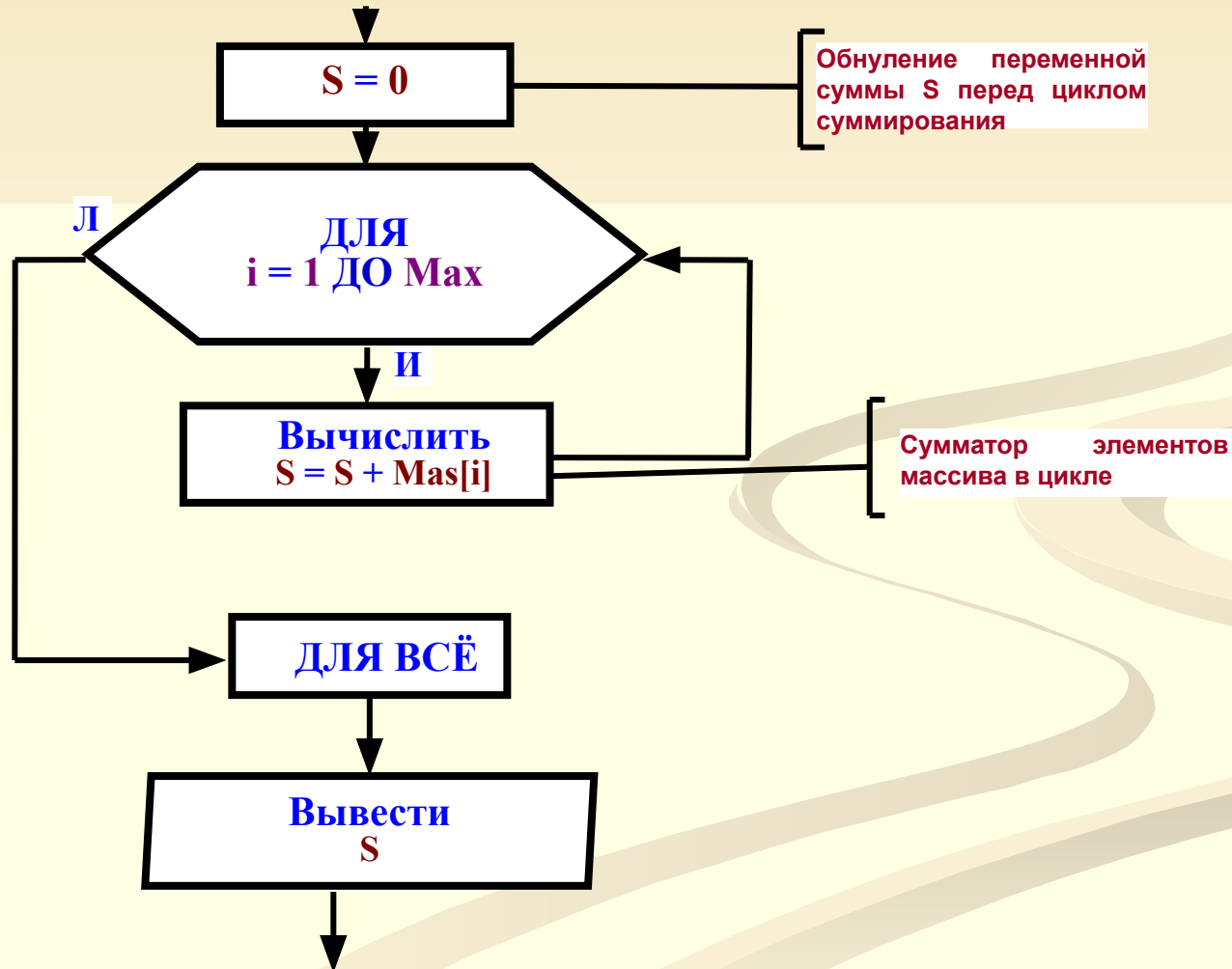


АЛГОхитрости

Типовые алгоритмические приёмы с массивами

Одномерные массивы. Сумма элементов массива

Все элементы массива целых чисел **Mas** суммируются в переменную **S**, которая затем выводится на экран. **Max** – размерность массива.

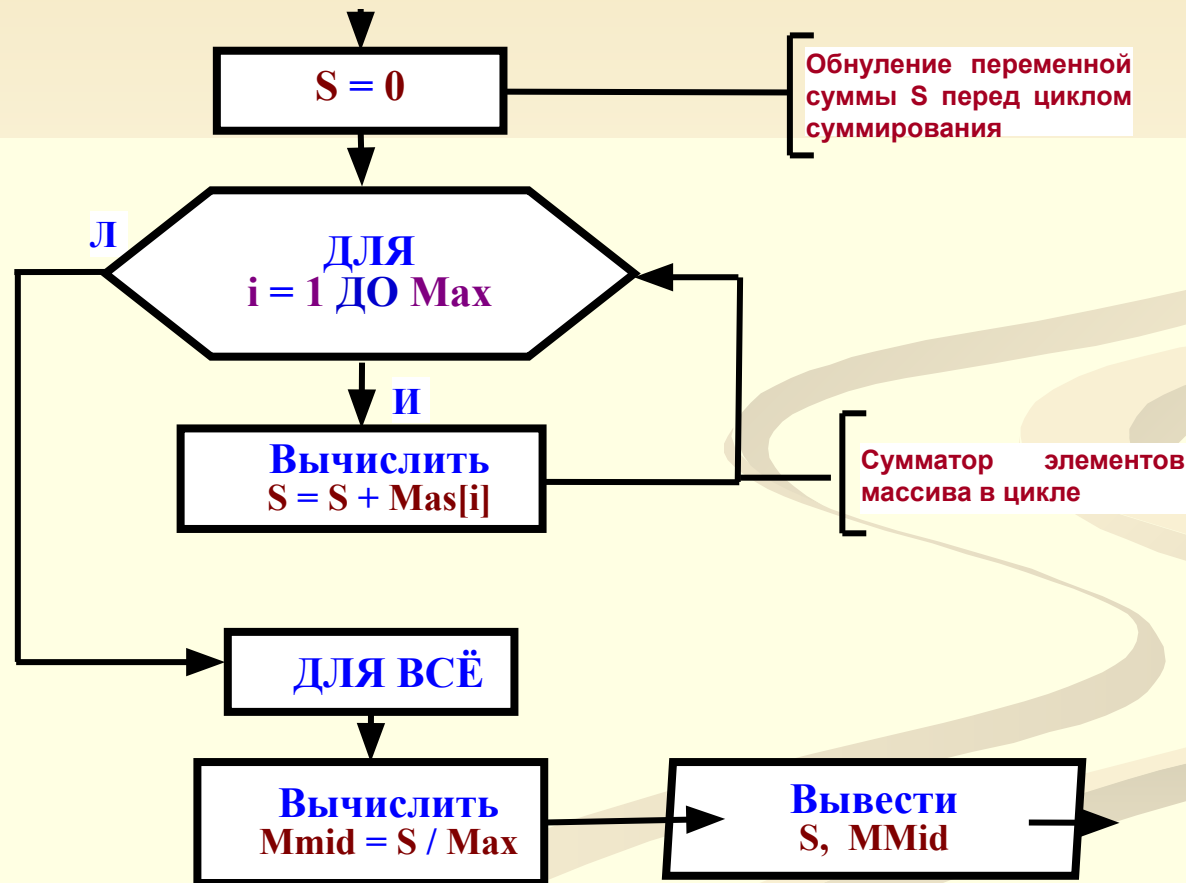


АЛГОхитрости

Типовые алгоритмические приёмы с массивами

Одномерные массивы. Среднее значение элементов массива

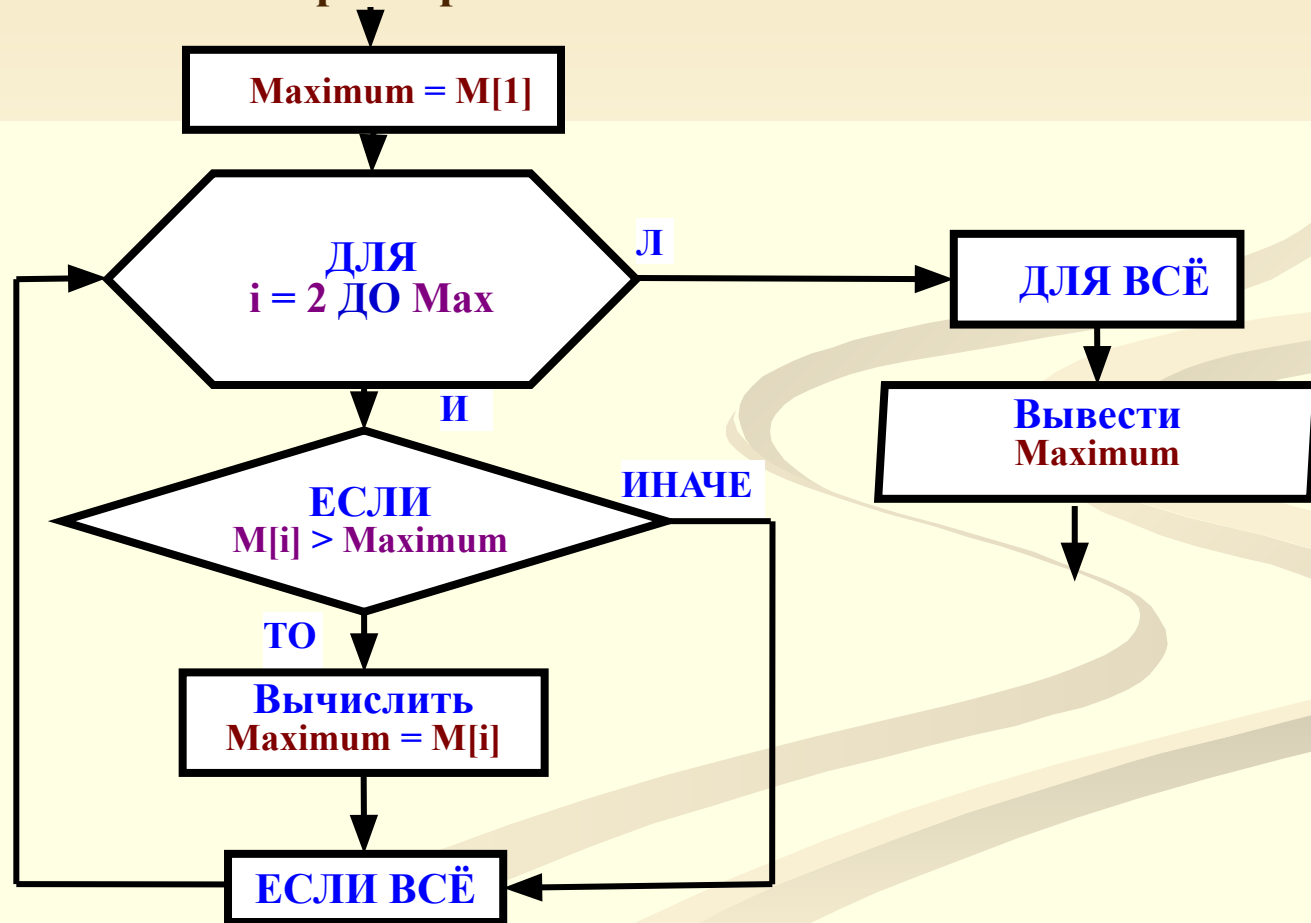
Для определения среднего значения элементов массива сначала вычисляется их сумма и потом делится на количество элементов. **Max** – размерность массива. **MMid** – среднее значение.



Типовые алгоритмические приёмы с массивами

Одномерные массивы. Наибольший (наименьший) элемент массива

Определение наибольшего элемента массива начинается с помещения значения первого элемента массива в переменную **Maximum (Minimum)**, затем значение этой переменной поочерёдно сравнивается со всеми элементами массива и если находится большее (меньшее) значение в элементе массива, то оно заменяет значение переменной. **Max** – размерность массива.



C / C++

Элементы ЯПВУ

МАССИВЫ

Практическое занятие

Задание: Введите с клавиатуры одномерный массив из 6 элементов, найдите и выведите на печать минимальный элемент массива

// Поиск минимального элемента массива

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

```
#define MS 6      //размер массива
```

```
void main ()
```

```
{
```

```
    int a[MS]; // одномерный массив
```

```
    int min, i; // min – минимальный элем. массива
```

```
    printf ("\n Введите элементы массива – ");
```

```
    printf ("%i", MS);
```

```
    printf (" целых чисел в одной строке \n через пробел и нажмите Enter -> ");
```

```
        for (i = 0; i < MS; i++)
```

```
            scanf ("%i", &a[i]);
```

```
    min = a[0]; // назнач. мин. первый элем. массива
```

```
    // Сравниваем остальные элементы с мин.
```

```
        for (i = 0; i < MS; i++)
```

```
            if (a[i] < min) min=a[i];
```

```
    printf ("Минимальный элемент массива");
```

```
    printf ("= %i ", min);
```

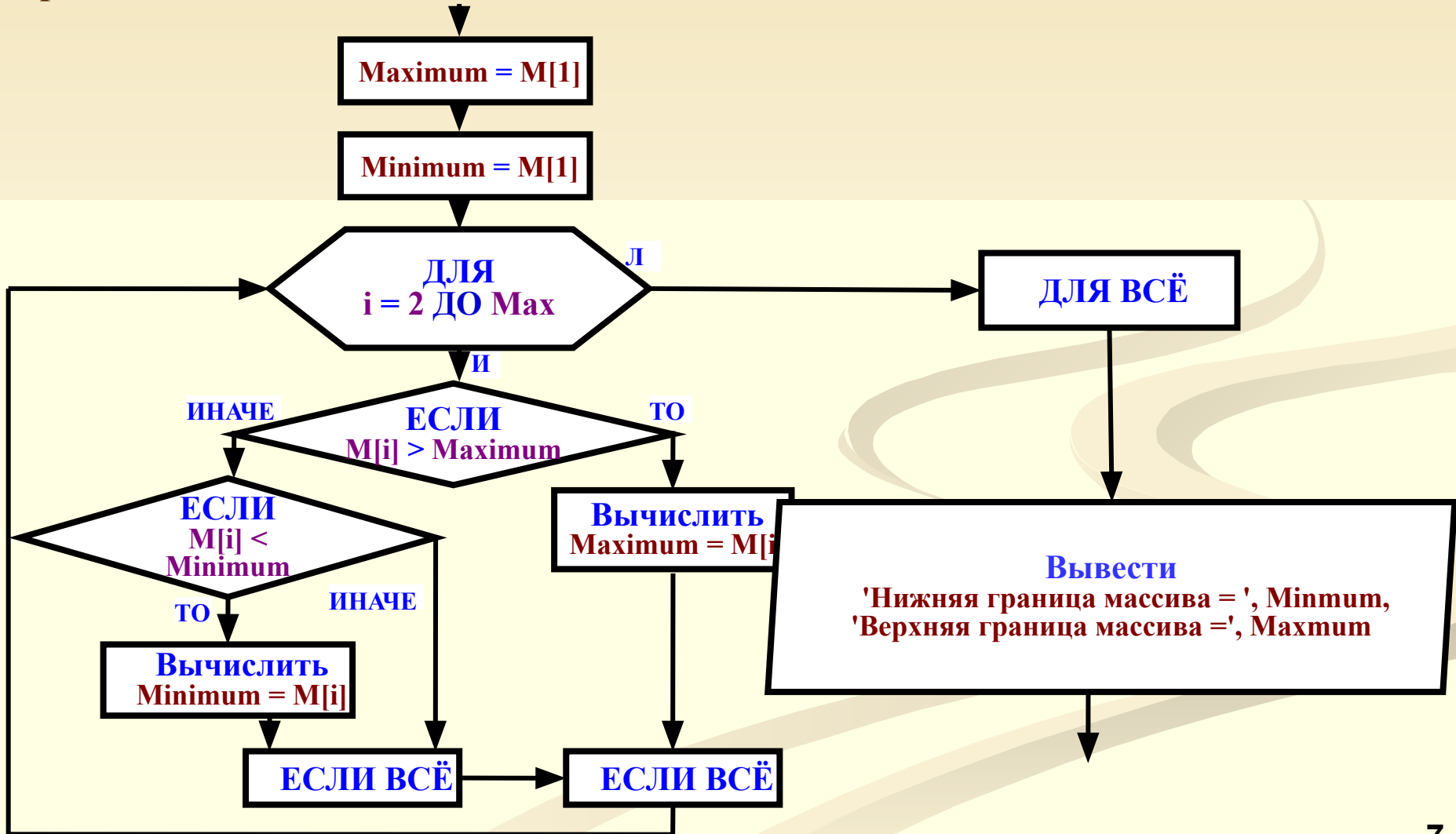
```
}
```

АЛГОхитрости

Типовые алгоритмические приёмы с массивами

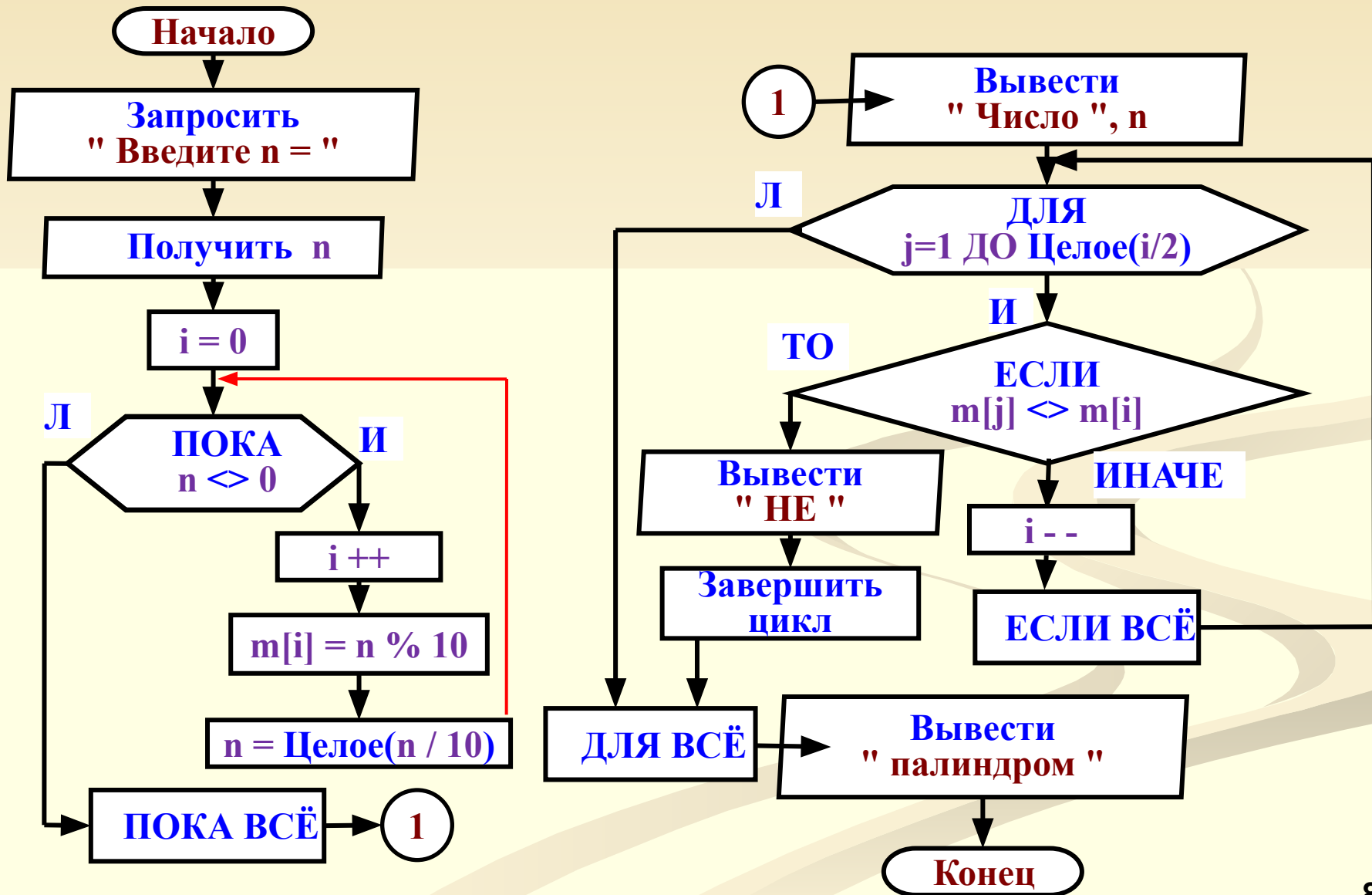
Одномерные массивы. Диапазон значений элементов массива

Определяются наименьший и наибольший элементы массива, они являются границами диапазона значений элементов массива.



АЛГОхитрости

Типовые алгоритмические приёмы с массивами
 Определить является ли введённое число палиндромом



АЛГОхитрости

Типовые алгоритмические приёмы с массивами
Определить является ли введённое число палиндромом

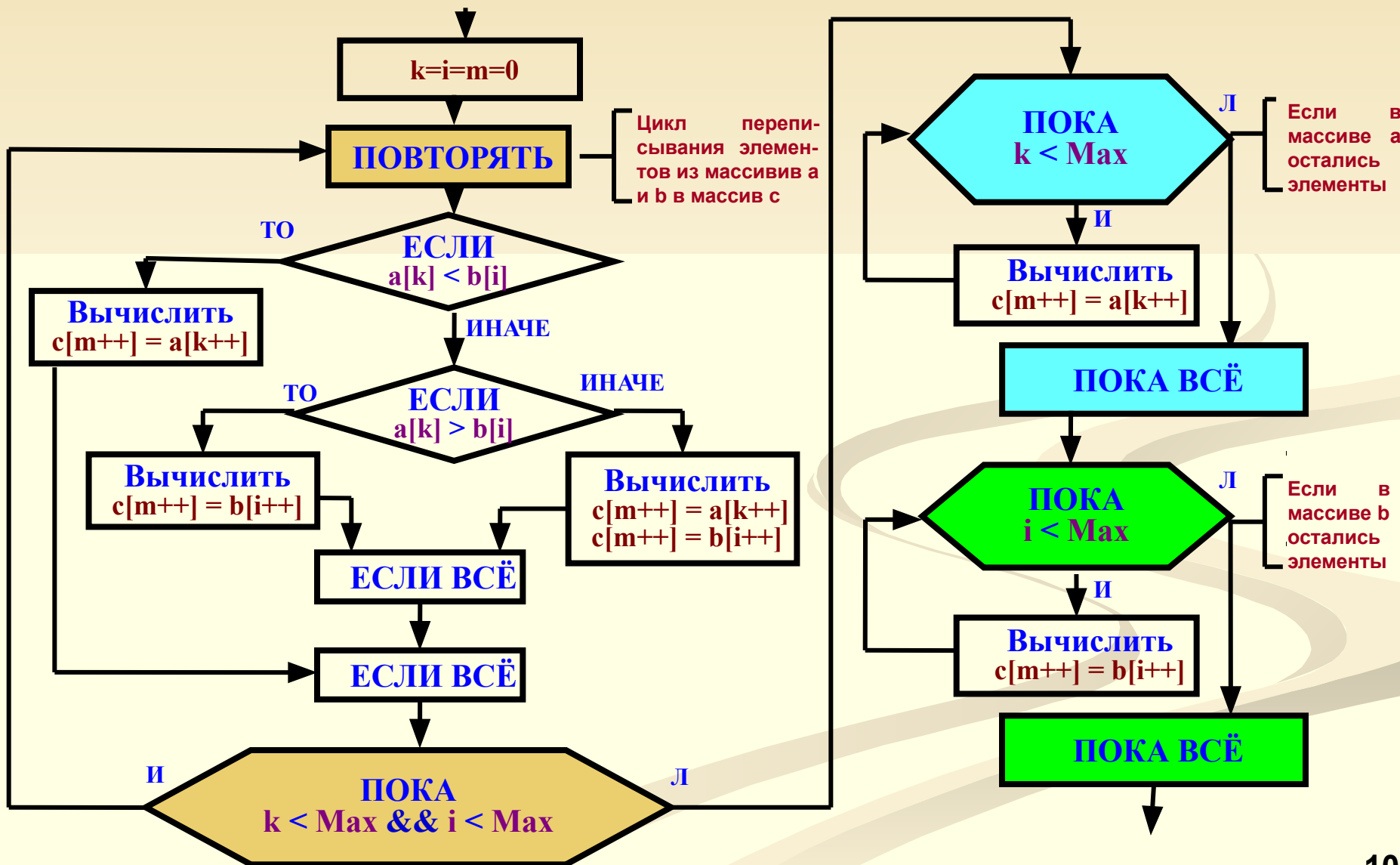
// Определить палиндром ли введенное число

```
#include <stdio.h>
#define SZ 5
main() {
int n, m, i=0, j, P[SZ];
printf("\nВведите целое число (до 32 767) n=");
scanf("%D", &n);
m=n;
while(m!=0)
{ P[i] = m % 10; m = (int) m / 10; i++; }
printf("Число %d", n);
for (j=0; j <= (int)i/2; j++)
if (P[j] != P[i-1])
{
printf(" НЕ ");
break;
}
else i--;
printf(" палиндром");
return 0;
}
```

```
// Вариант цикла проверки на палиндром
for (j=0; j <= (int)i/2; j++, i--)
if (P[j] != P[i-1])
{
printf(" НЕ ");
break;
}
else continue;
```

АЛГОхитрости

Типовые алгоритмические приёмы работы с массивами
Одномерные массивы. Слияние двух упорядоченных массивов в
третий упорядоченный. Max – размерность массива.



АЛГОхитрости

МАССИВЫ

Практическое занятие

С

Слить два упорядоченных по возрастанию массива целых чисел введённых с клавиатуры в один упорядоченный по возрастанию и вывести новый массив на печать.

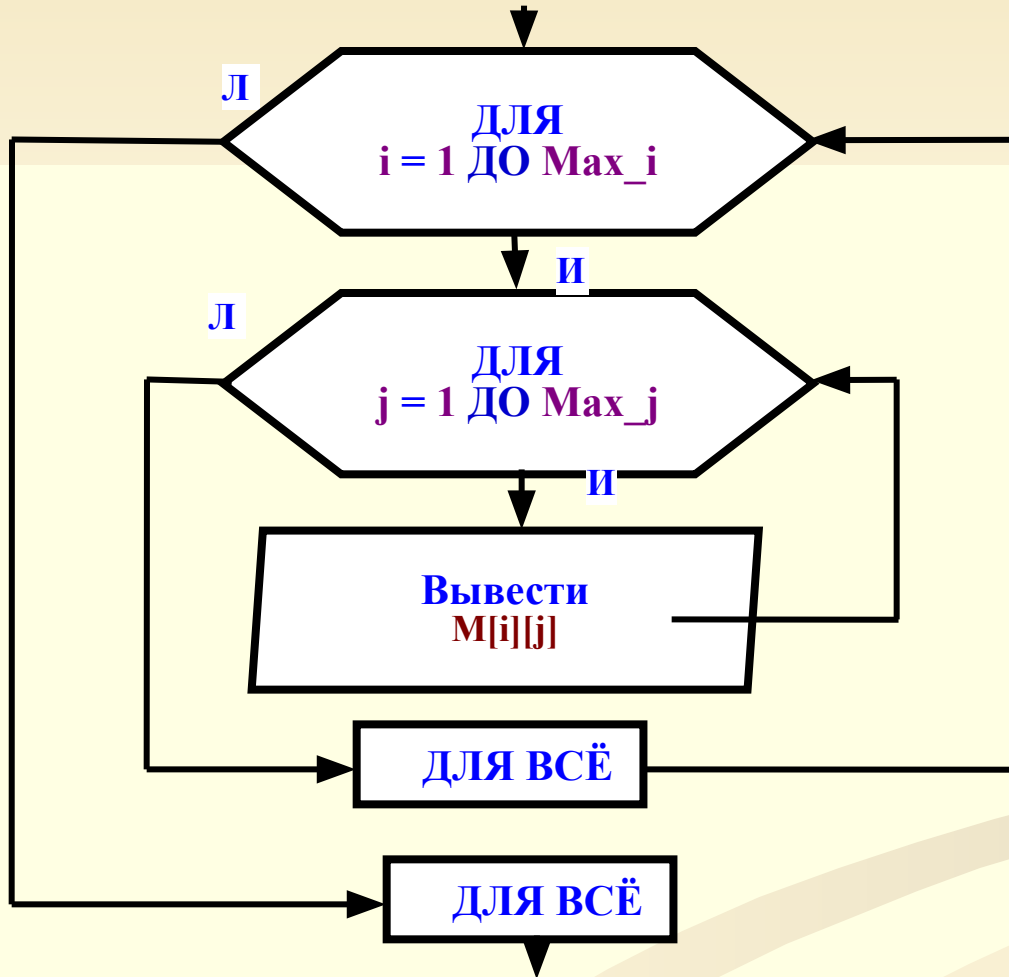
```
#include <stdio.h>
#include <conio.h>
#define SZ 5          //Размер исходных массивов
void main()   {
    int a[SZ], b[SZ];    // исходные массивы
    int c[SZ*2];        // массив-результат
    int k,i,m;          // индексы массивов a, b и c
    printf ("Слияние двух упорядоченных массивов в один,\n");
    printf ("ввод элементов через пробел, завершение: Enter\n");
    printf ("\nВведите первый массив %i -> ", SZ);
    for (k=0; k<SZ; k++)    scanf ("%i", &a[k]);
    printf ("Введите второй массив %i -> ", SZ);
    for (i=0; i<SZ; i++)    scanf ("%i", &b[i]);
    k=i=m=0;
    do { if (a[k] < b[i])      c[m++] = a[k++];
        else if (a[k] > b[i])  c[m++] = b[i++];
        else { c[m++] = a[k++]; c[m++] = b[i++]; } }
    while ((k < SZ) && (i < SZ));
    while (k < SZ) /*есть элем. массива А не переписанные в С*/
        c[m++] = a[k++];
    while (i < SZ) /*есть элементы В не переписанные в С*/
        c[m++] = b[i++];
    printf("Массив-результат: \n");
    for (i=0; i<2*SZ; i++)    printf ("%i ", c[i]);
    printf("\nДля завершения работы нажмите Enter\n");
    getch(); }

```

Типовые алгоритмические приёмы с массивами

Двумерные массивы. Считывание содержимого массива

Считывание – вывод на экран, в файл, на печать содержимого массива – реализуется обычно с использованием циклов **ДЛЯ** начиная с первого элемента до конца массива. Первый индекс (**i**) определяет номер строки, второй индекс (**j**) – номер столбца. **Max_i** – количество строк массива, **Max_j** – количество столбцов.



Для смещения по каждому индексу массива выделяется отдельный цикл. Циклы вложены один в другой.

Внешний цикл (по индексу - **i**) смещается по строкам (выбирает строку) и передает управление внутреннему циклу.

Внутренний цикл (по индексу - **j**) смещается по столбцам – выбирает в текущей строке ячейку, соответствующую столбцу массива (таблицы).

Затем управление возвращается внешнему циклу, происходит переход к следующей строке массива.

И так пока не прочитается весь массив.

Типовые алгоритмические приёмы с массивами

C / C++

Массивы

Двумерные массивы. Загрузка содержимого массива с клавиатуры и вывод его на экран (пример вывода двумерного массива на экран)

// Заполнение двумерного массива

```
#include<stdio.h>
// Количество строк массива
#define SIZE_i 5
// Количество столбцов массива
#define SIZE_j 5

void main ()
{
int a[SIZE_i][SIZE_j]; // Двумерный массив
int i, j; // индексы массива
printf("\nВведите элементы массива\n");
```

см. продолжение

Продолжение

// Ввод элементов массива с клавиатуры

```
printf("После ввода числа - Enter\n");
for (i=0; i<SIZE_i; i++)
for (j=0; j<SIZE_j; j++)
{
printf ("a[%i][%i] = ",i,j);
scanf ("%i",&a[i][j]);
}
```

// Вывод массива на экран

```
for (i=0; i<SIZE_i; i++)
{
for (j=0; j<SIZE_j; j++)
printf("%4i \n",a[i][j]);
printf("\n");
}
}
```

АЛГОхитрости

Типовые алгоритмические приёмы с массивами

Двумерные массивы. Сумма элементов массива

Все элементы массива целых чисел **Mas** суммируются в переменную **S**.
Max_i и **Max_j** – размерность векторов массива.

