

0

1 курс

ПЛАН – КОНСПЕКТ
Проведения практического занятия № 25
по дисциплине «Информатика»

Раздел 4. «Основы алгоритмизации и программирования.»

Тема 4.2.: «Структурированные типы данных.»

к частям 7-8

Подготовил: преподаватель
В.Н. Борисов

Рязань 2023

Вопросы занятия:

1. Составление программ, использующих процедуры ввода-вывода и обработки массивов (практическое занятие № 25, теоретическая часть, выполнение практического задания).

Время проведения занятия – 2 часа.

Первый вопрос: Составление программ, использующих процедуры ввода-вывода и обработки массивов (практическое занятие № 25, теоретическая часть, выполнение практического задания).

6.5. Примеры типовых задач на тему «Подпрограммы»

1. Задан одномерный массив вещественных чисел. Найти максимальный элемент массива и определить его номер.

Составим программу с использованием двух процедур:

1) процедура ввода элементов массива;

2) процедура поиска максимального элемента и определение его номера.

Для передачи массива в процедуру введем пользовательский тип mas.

Программа	Пояснения
<pre> program PP; type mas=array[1..100] of integer; var a:mas; max:real; n,max,kmax:integer; procedure input_mas(n:integer ; var a:mas); var i:integer; begin for i:=1 to n do readln(a[i]); end; procedure max_m(n:integer; a:mas; var max,kmax:integer); var i:integer; begin max:=a[1];kmax:=1; for i:=1 to n do if a[i]>max then begin max:=a[i]; kmax:=i; end; end; begin writeln(' Введите n<=100') ; readln(n); writeln (' Введите элементы массива '); input_mas(n,a); max_m(n,a,max,kmax); writeln(' max=',max,' kmax=',kmax); readln; end. </pre>	<p>Имя программы – PP {Описательная часть программы} Пользовательский тип данных</p> <p>Глобальные переменные Текст первой процедуры {Описательная часть процедуры} {Исполнительная часть процедуры} Ввод элементов массива</p> <p>Текст второй процедуры {Описательная часть процедуры} {Исполнительная часть процедуры}</p> <p>Поиск максимального элемента</p> <p>{Исполнительная часть программы} Вывод сообщения на экран Ввод размера массива Вывод сообщения на экран Обращение к первой процедуре Обращение ко второй процедуре Вывод результатов на экран</p>

2. Составить программу вычисления $n!$ с использованием функции, в которую вынесем подсчет факториала.

$$n! = 1 * 2 * 3 * \dots * n$$

Введем обозначения:

n, k – размер и индекс массива, глобальные переменные;

p, i – значение факториала и индекс массива в функции, локальные переменные;

fup – имя функции, возвращает результат в основную программу.

Программа	Пояснения
<pre> program factorial; var n,k:integer; function fun(k:integer):integer; var p,i:integer; begin p:=1; for i:=1 to k do p:=p*i; fun:=p; end; begin writeln('введите n'); readln(n); for k:=1 to n do writeln(fun(k)); readln; end. </pre>	<p>Имя программы – factorial {Описательная часть программы} Глобальные переменные Текст функции {Описательная часть функции} {Исполнительная часть функции}</p> <p>{Описательная часть процедуры} {Исполнительная часть процедуры}</p> <p>{Исполнительная часть программы} Вывод сообщения на экран Ввод размера массива</p> <p>Обращение к функции</p>

Имя функции в теле функции в операторах присваивания может стоять только слева от знака присваивания «:=», поэтому использована дополнительная переменная *p* для подсчета произведения.

3. Задан двумерный массив целых чисел. Определить количество четных чисел в массиве.

Составим программу с использованием двух процедур и функции:

- процедура ввода элементов массива *input_mas*;
- процедура вывода двумерного массива в виде матрицы *print_mas*;
- функция подсчета количества четных чисел *kol_mas*.

Программа	Пояснения
<pre> program pp_1; type mas=array[1..10,1..10] of integer; var a:mas; n,m,i,j:integer ; procedure input_mas(n,m:integer; var a:mas); begin for i:=1 to n do for j:=1 to m do readln(a[i,j]); end; function kol_mas(n,m:integer; a:mas):integer; </pre>	<p>Имя программы – pp_1 {Описательная часть программы} Глобальные переменные</p> <p>Процедура ввода матрицы {Исполнительная часть процедуры}</p> <p>Функция подсчета количества четных чисел</p>

<pre> var k:integer ; begin kol:=0; for i:=1 to n do for j:=1 to n do if not odd(a[i,j]) then k:=k+1; kol:=k; end; procedure print_mas(n,m:integer; a:mas); begin for i:=1 to n do begin for j:=1 to n do write(a[i,j], ' '); writeln; end; end; begin writeln('Введите n<=10 , m<=10'); readln (n,m); writeln('Введите элементы массива по строкам'); input_mas(n,m,a); writeln('Исходная матрица'); print_mas(n,m,a); writeln('Количество нечетных чисел =',kol_mas(n,m,a)); readln; end. </pre>	<pre> {Описательная часть функции} {Исполнительная часть функции} Процедура вывода матрицы {Исполнительная часть процедуры} {Исполнительная часть программы} Вывод сообщения на экран Ввод размера массива Вывод сообщения на экран Обращение к процедуре ввода массива Обращение к процедуре вывода массива Обращение к функции подсчета количества четных чисел в массиве </pre>
---	--

Выражение `not odd(a[i,j])` истинно, если элемент массива — четное число.

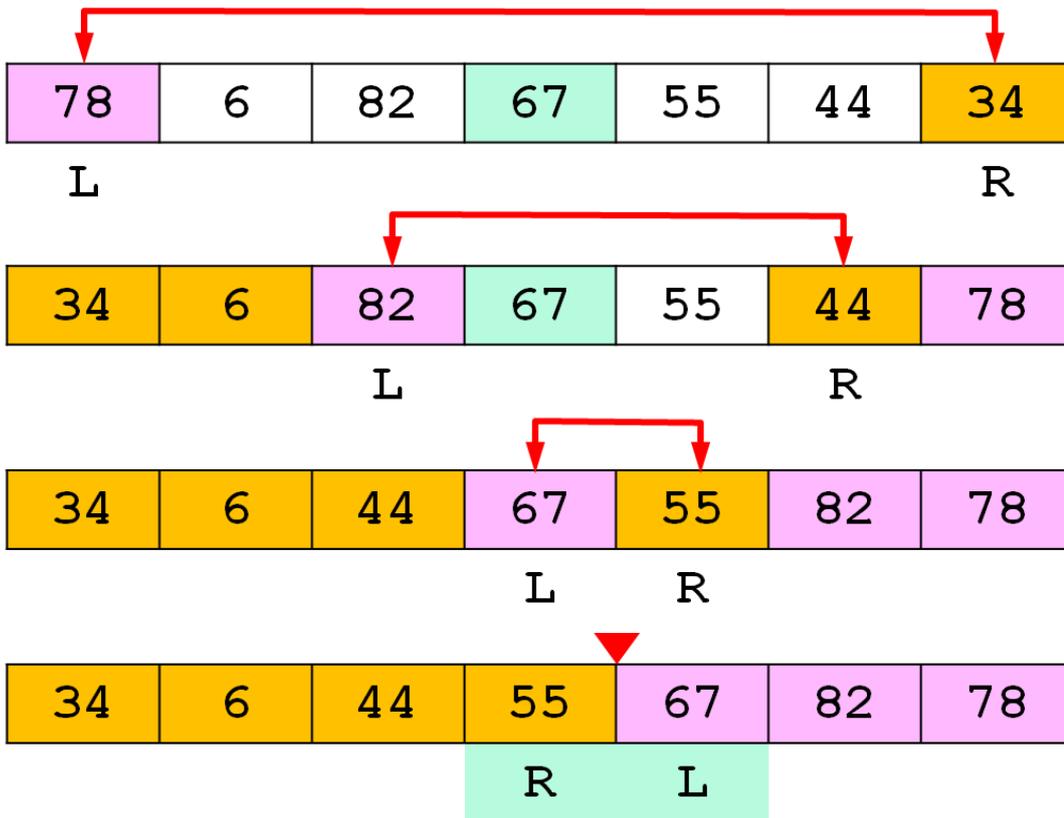
Быстрая сортировка или quick sort.

Алгоритм:

1. Выбирается и запоминается средний элемент массива (присвоим X):

78	6	82	67	55	44	34
----	---	----	----	----	----	----

2. Инициализируем две переменные (будущие индексы массива): $L:=1$, $R:=N$ (N — количество элементов).
3. Увеличиваем L и ищем первый элемент $A[L]$, который больше либо равен X (в итоге он должен находиться справа).
4. Уменьшаем R и ищем элемент $A[R]$, который меньше либо равен X (в итоге он должен находиться слева).
5. Смотрим, если $L \leq R$, то меняем местами $A[L]$ и $A[R]$, возвращаемся к пункту 3.



L стало больше R

```
procedure QSort ( first, last: integer);
```

```
var L, R, c, X: integer;
```

```
begin
```

```
  if first < last then begin
```

```
    X:=A[(first+last) div 2];
```

```
    L:=first; R:=last;
```

```
    while L <= R do begin
```

```
      while A[L] < X do L:=L+1;
```

```
      while A[R] > X do R:=R-1;
```

```
      if L<=R then begin
```

```
        c:=A[L]; A[L]:=A[R]; A[R]:=c;
```

```
        L:=L+1; R:=R-1;
```

```
      end;
```

```
    end;
```

```
    QSort(first, R);  QSort(L, last);
```

```
  end;
```

```
end.
```

для рекурсии

Делаем обмен

идем дальше

сортировка двух частей