Лабораторная работа №4

Итерационные алгоритмы. Флаги. Накопление суммы.

Поиск максимума

**Флаги**

В условных циклах в качестве условия входа в цикл или выхода из цикла часто удобно использовать так называемые *Boolean-флаги*.

**Флаг** – это логическая переменная типа Boolean, которая принимает значение TRUE или FALSE и указывает, произошло событие или нет. Значение флага первоначально устанавливается в FALSE, т.к. событие еще не произошло. Когда событие совершается, значение переменной изменяется на TRUE. Или наоборот, (см. Пример 1) предполагается, что событие (свойство) имеет место, поэтому флаг «поднят» (flag=true), и как только событие (свойство) нарушается, флаг «опускается» (flag=false).

Пример 1

Пример программы, которая определяет, является ли натуральное число N простым.

Первоначально предполагается, что число N простое (флаг «поднимается» Fl=true). Затем в цикле проверяются все предполагаемые делители, начиная с 2 и до числа N div 2. Если среди них найдется хотя бы одно число, на которое N делится нацело, флаг «опускается» (fl=false) и выполняется выход из цикла. Но, если введенное число окажется простым, флаг никогда не «опустится», поэтому на этот случай добавляется второе условие выхода из цикла: del=N div 2.

program Prost\_chislo;

Uses CRT;

Var N,del:integer;

**Fl:Boolean;**

Begin

write('Введите число N: ');

readln(N);

**fl:=true;** *{число простое}*

**del:=2;** *{первый делитель на проверку}*

**Repeat**

If N mod del =0 then **fl:=false** *{ число не простое}*

else del:=del+1; *{получаем следующий делитель}*

**Until (fl=false)or (del=N div 2);** *{число не простое }*

if **fl** then writeln('Число простое!')

else Writeln('Число не простое!');

end.

В последнем операторе IF условие задано логической переменной fl, а не логическим выражением (fl=true или fl=false), что возможно, так как сама переменная fl имеет логический тип, т.е. принимает значение true или false.

Накопление суммы

При вычислении суммы последовательности из N чисел используется стандартный алгоритм, который накапливает сумму в переменной-накопителе. Так как чисел много, то необходимо использовать цикл.

**

Аналогично можно посчитать произведение N чисел, вводимых с клавиатуры. Только при этом первоначальное значение переменной-накопителя изменится.

*Пример 2*

**P:=1;**

**For i:=1 to N do**

**Begin**

Readln(a);

**P:=P\*a;**

**End;**

Пример 3

Вычислить S*=sin x+sin2 x+…+sinn x*, где *n*-натуральное число, *x* – вещественное.

Разработаем алгоритм решения задачи: каждое слагаемое суммы представляет собой степень синуса. Поэтому в цикле помимо суммы необходимо накапливать и степень синуса. Степень синуса представляет собой накопление произведения (*sin2 x=sin x\*sin x, sin3 x=sin x\*sin x\*sin x*, и т.д. ).

Program Summa\_radya;

Var n,i: Integer; S,slag:real;

Begin

Writeln(‘Введите n: ’); Readln(N);

Write(‘Введите число x-’); Readln(x);

**S:=0; slag:=1;**

**For i:=1 to N do**

**Begin**

**Slag:=slag\*sin(x);**

S:=S+slag;

**End;**

Writeln (‘Summa =’,S:7:2);

End.

**Поиск максимального/минимального значения**

На практике часто приходится сталкиваться с такой задачей, как поиск максимального или минимального значения среди множества значений. Например, поиск максимума функции, поиск максимальной прибыли, поиск кратчайшего расстояния между двумя пунктами, поиск максимальной пропускной способности канала и т.д. Алгоритмы решения всех этих задач опираются на стандартный алгоритм поиска максимума, с которым мы сейчас и познакомимся.

Пример 4

С клавиатуры вводится последовательность из *N* целых чисел *a1, a2, … an*. Необходимо определить, какое значение являлось максимальным.

Program Maximum;

**Алгоритм поиска:**

*Шаг 1.* Переменной Max присвоить значение первого числа а1.

*Шаг 2.* i=2.

*Шаг 3.* Если аi больше Max, то Max присвоить аi.

*Шаг 4*. i=i+1.

*Шаг 5.* Если i<N, то перейти на шаг 3, иначе на шаг 6.

*Шаг 6.* Конец.

Var a, i, Max, N: Integer;

Begin

Writeln(‘Введите кол-во чисел: ’);

Readln(N);

Write(‘Введите число-’);

Readln(a);

**Max:=a;**

**For i:=2 to N do**

**Begin**

Write(‘Введите число-’);

Readln(a);

**If a>max Then max:=a;**

**End;**

Writeln (‘Max =’,Max);

End.

**Задания для самостоятельной работы**

Задание 4.1

Вводится последовательность из N целых чисел. Выяснить, является ли последовательность монотонно возрастающей (убывающей).

Задание 4.2

Вводится последовательность N целых чисел. Вычислить сумму четных элементов последовательности.

Задание 4.3

Вводится последовательность N целых чисел. Вычислить среднее арифметическое элементов последовательности, заканчивающихся цифрой 6.

Задание 4.4

Вводится последовательность целых чисел до первого нуля. Определить разность между максимальным и минимальным элементами последовательности.

Задание 4.5

Вводится последовательность целых чисел до первого нуля. Определить порядковый номер максимального элемента последовательности.

Задание 4.6

Разработать алгоритм определения двух наименьших элементов из последовательности N целых чисел и вывести эти элементы в порядке возрастания их значений.

Задание 4.7

С клавиатуры вводится  последовательность из *N*  (> 3) целых чисел. Найти три наибольших элемента из данного набора и вывести эти элементы в порядке убывания их значений.

Задание 4.9

С клавиатуры вводится  *N* целых чисел. Найти максимальную сумму двух соседних чисел из данного набора.

Задание 4.10

Вводится натуральное число *N*. Определить максимальную цифру в записи числа.

Задание 4.11

Составить программу, которая определяет минимальное значение функции *y= x - sin x* при заданном интервале [*a,b*] и шаге *h*.

Задание 4.12

Дано целое число *N* (> 1). Вывести наименьшее из целых чисел *K*, для которых сумма 1 + 2 + … + *K* ≥ *N*.

Задание 4.13

Напишите программу, которая вычисляет степень числа *an*, где *n*-целое число (воспользуйтесь определением степени - ).

Задание 4.14

Выяснить, является ли функция *y=–x3+ax2+bx+c* монотонно убывающей на заданном интервале [*x1,x2*] с шагом h.

Задание 4.15

Составить программу, которая считает факториал заданного числа n (n!=1\*2\*3\*…\*n) .

Задание 4.16

Вычислить сумму ряда:

1. Y=x10+2 x9+3 x8+…+10 x+11;
2. Y=11 x10+10 x9+9 x8+…+2 x+1.

Задание 4.17

С клавиатуры вводится  последовательность из *N*  (> 3) целых чисел. Найти номера двух соседних чисел из данного набора, произведение которых является минимальным, и вывести вначале меньший, а затем больший номер.

Задание 4.18

Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между первым и последним нулем (если первый и последний нули идут подряд, то вывести 0).

Задание 4.19

Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между последними двумя нулями (если последние нули идут подряд, то вывести 0).

Задание 4.20

Разработать алгоритм, который определяет квадратный корень из числа А, учитывая, что сумма первых *n* нечетных чисел равна *n2*:

1=12

1+3=4=22

1+3+5=9=32

1+3+5+7=16=42.

Задание 4.21

Разработать программу вычисления: .

Задание 4.22

Составить программу возведения заданного натурального числа N в третью степень, используя следующую закономерность:

13=1

23=3+5

33=7+9+11

43=13+15+17+19

53=21+23+25+27+29

Задание 4.23

С клавиатуры вводится  *N* целых чисел. Найти максимальное количество четных чисел в наборе, идущих подряд. Если четные числа в наборе отсутствуют, то вывести 0.

Задание 4.24

Дано целое число *N* и набор из *N* целых чисел, содержащий только нули и единицы. Найти номер элемента, с которого начинается самая длинная последовательность одинаковых чисел, и количество элементов в этой последовательности. Если таких последовательностей несколько, то вывести номер первой из них.

Задание 4.25

Дано целое число *N* (> 1). Найти наибольшее целое число *K*, при котором выполняется неравенство *3k < N*.

Задание 4.26

Для заданного положительного вещественного числа A необходимо найти максимальное целое число K, при котором выполняется неравенство .

Задание 4.27

С клавиатуры вводится  последовательность из *N*  (> 3) целых чисел. Найти три наибольших элемента из данного набора и вывести эти элементы в порядке убывания их значений.

***ИСТ: 4.1, 4.3, 4.10, 4.13***

***4.25, 4.19, 4.26, 4.27***