

## Лабораторная работа №4

### "Программирование циклических алгоритмов"

Цель работы: ознакомиться с циклическими алгоритмами и операторами, реализующими эти алгоритмы. Освоить особенности применения каждого оператора. Составить программы с использованием всех операторов цикла.

#### Теоретические сведения

##### **Оператор цикла while**

Описание:

`while (выражение) оператор;`

Действие:

Выполняется оператор до тех пор, пока значение выражения в скобках истинно. Проверка значения выражения происходит перед каждым выполнением оператора. Когда значение выражения ложно, цикл `while` заканчивается. Если выражение ложно с самого начала, оператор не выполняется ни разу.

Комментарий:

Следует заметить, что после ключевого слова `while` и выражения, заключенного в круглые скобки, точка с запятой не ставится.

Оператор иногда называется телом цикла. В теле цикла должны выполняться действия, в результате которых меняется значение управляющего выражения. В противном случае можем получить бесконечный цикл.

Пример:

```
/*Демонстрация цикла while*/
#include <stdio.h>
main( )
{
    int i=1
    while (getchar()!='R') i++;
    /*оператор getchar() вводит любой символ с клавиатуры*/
    printf("Символ R %d-й", i);
}
```

Приведенная программа позволяет определить порядковый номер первой введенной буквы R в последовательности символов. Она показывает использование цикла `while`, в теле которого всего одна инструкция (`i++` - увеличение значения целого числа `i` на единицу). Если запустить эту программу на выполнение и ввести последовательность символов, например: `abFk!Rgm`, то на экране появится строка: Символ R 6-й.

##### **Оператор цикла do - while**

Описание: `do оператор while (выражение);`

Действие: В операторе `do-while` тело цикла выполняется по крайней мере один раз. Тело цикла будет выполняться до тех пор, пока выражение в скобках не примет ложное значение. Если оно ложно при входе в цикл, то его тело выполняется ровно один раз.

Комментарий: после слова `while` и выражения, заключенного в скобки, ставится точка с запятой. Если в теле цикла содержится более одной инструкции, то операторы цикла заключаются в фигурные скобки.

Пример:

```
/*Демонстрация цикла do-while */
#include <stdio.h>
main()
{
    int i=0; /*i=0, а не единице*/
    do i++; while (getchar()!='R');
    printf("Символ R %d-й",i);
}
```

Программа, представленная выше, теперь написана с циклом `do-while`. Результат программы будет таким же.

### Оператор цикла `for`

Описание: `for` (выражение 1; выражение 2; выражение 3) оператор;

Действие:

В круглых скобках содержится три выражения. Первое из них служит для инициализации счетчика. Она осуществляется только один раз – когда цикл `for` начинает выполняться. Второе выражение необходимо для проверки условия, которая осуществляется перед каждым возможным выполнением тела цикла. Когда выражение становится ложным, цикл завершается. Третье выражение вычисляется в конце каждого выполнения тела цикла, происходит приращение числа на шаг.

Комментарий: в операторе цикла `for` точка с запятой после закрывающейся круглой скобки не ставится. Любое из трех или все три выражения в операторе могут отсутствовать, однако разделяющие их точки с запятыми опускать нельзя. Если отсутствует выражение 2, имеем бесконечный цикл. Например: `for (scanf("%d",&p); ; p++) оператор;`

В языке СИ предусмотрены две нетрадиционные операции: `(++)` – для увеличения на единицу и `(--)` – для уменьшения на единицу значения операнда. Операции `++` и `--` можно записывать как перед операндом, так и после него. В первом случае (`++n` или `--n`) значение операнда (`n`) изменяется перед его использованием в соответствующем выражении, а во втором (`n++` или `n--`) – после его использования.

Если отсутствуют выражения 1 и 3, цикл становится эквивалентным `while`. Например: `for (; a<20;) оператор;`

Каждое из выражений может состоять из нескольких выражений, объединенных операцией "запятая". Например: `for(i=0, j=1; i<100; i++, j++) a[i]=b[j];`

Тело цикла заключается в фигурные скобки, если в нем более одного оператора.

Пример:

```
/*демонстрация цикла for*/
#include <stdio.h>
main()
{int i,j=1,k;
```

```

for (i=1;i<=3;i++)
printf("Минск\t");
/*В цикле for три раза выполняется функция вывода*/
/*Здесь i-управляющая переменная цикла*/
printf("\nУкажите число повторений цикла\n");
scanf("%d",&k);
for (i=1;i<=k;i++)
{
j*=i;
printf("%d",j);
}
/*Здесь две инструкции (более одной), поэтому они заключаются
в фигурные скобки*/
j=i;
printf("\n");
/*Переменной j присваивается значение 1 и осуществляется
перевод курсора*/
/*В следующем цикле for выполняются те же действия, что и в
предыдущем*/
for (i=1;i<=k;i++) printf("%d ", j*=i);
}

```

Результаты выполнения программы следующие:

Минск Минск Минск

Укажите число повторений цикла; 5

1 2 6 24 120

1 2 6 24 120

### **Оператор break**

Описание:

Break используется для прекращения выполнения цикла из-за обнаружения ошибки, для организации дополнения к условию в заголовке цикла, для прекращения бесконечного цикла.

Пример:

```

int st;
scanf("%d",&st);
while (st>0 && st<25)
{
if st==4||st==8||st==12)
break;
}

```

Работа цикла полностью прекращается, как только условие в операторе if становится истинным.

### **Оператор continue**

Действие: Этот оператор может использоваться во всех трех типах циклов. Как и в случае оператора break, он приводит к изменению характера выполнения программы. Однако вместо завершения работы цикла наличие оператора continue вызывает пропуск "оставшейся" части итерации и переход к началу следующей.

Пример. Заменим в предыдущей программе оператор break на continue.

```
int st;
scanf("%d", &st);
while (st>0 && st<25)
{
    if (st==4||st==8||st==12)
        continue;
    printf("%d", st);
}
```

При истинном условии в операторе if оператор continue вызывает пропуск идущих за ним операторов тела цикла и осуществляется переход к началу следующей итерации.

### Задание

1. Согласно варианту реализовать «Задание 1» в двух версиях: одна с применением цикла while, вторая – цикла for. Сделать вывод о том, какой вид цикла для данной задачи удобнее применять.
2. Согласно варианту реализовать «Задание 2» при помощи цикла do-while с условием выхода из цикла. Переписать программу при помощи бесконечного цикла do-while (условие которого всегда истина) с применением операторов break и continue. Сделать вывод, какой вариант программы лучше и почему?
3. Разработать блок-схемы алгоритмов (2 – по одной на каждую задачу).

**!!!! При написании программ выполнять проверку всех вводимых значений на возможность использования в программе (например: если вместо числа введена буква, если вместо положительного n введено отрицательное и т.п.)**

Вариант	Задание 1
1	Написать программу ввода произвольных чисел до тех пор, пока не будет введено число 0, вывести сообщение о количестве введенных чисел отрицательных и положительных
2	Написать программу ввода произвольных символов до тех пор, пока не будет введен символ q, вывести сообщение о том, сколько было введено символов a и символов f
3	Написать программу подсчета суммы n чисел, вводимых с клавиатуры, числа должны быть из диапазона от -20 до 20 (выполнять проверку).
4	Написать программу вычисления произведения n нечетных чисел, введенных с клавиатуры (числа можно вводить любые, необходимо выполнять проверку на нечетность).
5	Написать программу вычисления модулей введенных отрицательных чисел до тех пор, пока пользователь не введет 0 (для положительных чисел выводить просто сообщение).

6	Написать программу определения знака введенных чисел (числа должны проверяться на диапазон вхождения от -100 до 100) до тех пор, пока пользователь не введет 0
7	Написать программу определения минимального введенного числа из n чисел ( $n < 10$ – вводится с клавиатуры)
8	Даны два целых числа A и B ( $A < B$ ). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их убывания, а также количество N этих чисел.
9	Написать программу определения минимального среди положительных введенных n чисел ( $n < 15$ – вводится с клавиатуры), если введено отрицательное, то оно игнорируется
10	Определить, является ли заданное число совершенным, т.е. равным сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: $6 = 1 + 2 + 3$ ).
11	Написать программу, которая выводит на экран таблицу умножения
12	Написать программу определения максимального среди отрицательных введенных n чисел ( $n < 12$ – вводится с клавиатуры), если введено положительное, то оно игнорируется
13	Написать программу, которая рисует на экране елочку из звездочек :)
14	Написать программу определения суммы максимального и минимального числа из n введенных чисел ( $n < 14$ – вводится с клавиатуры).
15	Дано вещественное число A ( $> 1$ ). Вывести наибольшее из целых чисел N, для которых сумма $1 + 1/2 + \dots + 1/N$ будет меньше A, и саму эту сумму.

## Задание 2.

Вычислить значения функции  $f(x)$  на отрезке  $[a;b]$  с шагом  $h$ , кроме  $x = a + 2 \cdot h$ .

№	$f(x)$	$[a,b]$	$h$
1	2	3	4
1	$\ln(x)$	1;1.5	0.1
2	$1 + \ln^2(x)$	0.4;1	0.1
3	$1 + e^x$	0.5;0.6	0.01
4	$e^{x^2} / 2$	2;3	0.2
5	$\cos(x)e^{-x}$	1;2	0.2
6	$1/(1 + e^{-x})$	3;4	0.2
7	$\sin(x)\text{sh}(x)$	1;5	1
8	$0.5 + \text{sh}^2(x)$	2;3	0.2

9	$\sqrt{x}\text{ch}(x)$	3;4	0.2
10	$1/(1 + \text{ch}^2(x))$	2;4	0.5
11	$\sqrt{x}\text{sh}(x)$	1;5	1
12	$e^{-x}\text{ch}(x)$	1;4	1
13	$\ln(x^2)$	1;1.4	0.1
14	$x + \ln(x)$	1;5	1
15	$1/(1 + \sin x)$	$p/3;p/6$	$p/10$
16	$\sin x + \sqrt{x}$	$p/6;p/4$	$p/10$
17	$x(1 - \cos x)$	0.4;0.8	0.1
18	$e^{x+3}\sin x$	0;2	0.5
19	$\cos(x)\text{ch}(x)$	1;5	1
20	$e^{1+x}\text{sh}(x)$	1;4	1