

Практическая работа №11

Циклы **while** и **do**

1. Задание

Составить программу, которая:

- 1) вычисляет и распечатывает приближенное значение суммы, указанной в пункте «а» варианта задания, для произвольного значения x , которое должно вводиться с клавиатуры. Суммирование прекращается, если абсолютная величина очередного элемента суммы становится меньше 10^{-7} ;
- 2) вычисляет и распечатывает наименьший номер элемента последовательности, заданной рекуррентной формулой в пункте «б» варианта задания, для которого выполняется указанное в задании условие.

2. Требования к выполнению задания

При выполнении задания необходимо придерживаться следующих основных требований:

- 1) решение задач должно выполняться с использованием операторов присваивания, описания, циклов **while** и **do**, а также функций ввода и вывода;
- 2) при решении задачи, указанной в пункте «а», требуется использовать оператор цикла **while**, значения параметра x следует вводить с клавиатуры;
- 3) при организации цикла в пункте «б» следует вывести рекуррентную формулу, связывающую два последовательных элемента суммы, вывод формулы включается в текст отчета;
- 4) задача из пункта «в» решается с использованием оператор цикла **do**;
- 5) для проверки программы следует подготовить таблицу трассировки для каждого цикла;
- 6) обязательно наличие исходного текста программы, записанного в файле;
- 7) все операторы цикла должны оформляться с помощью отступов.

3. Примечания

Циклом называется управляющая конструкции языка программирования, предназначенная для организации многократного выполнения некоторого множества операторов. Множество операторов, которые должны многократно выполняться в цикле, принято называть телом цикла.

3.1 Оператор цикла **while**

Общая форма оператора цикла **while** имеет следующий вид:

<цикл while>: while (<условие>) [<оператор>];

Оператор, представляющий собой тело цикла, может быть пустым оператором, единственным оператором или составным оператором.

Условие (управляющее выражение) может быть любым допустимым выражением. Если значение выражения не равно нулю, то считается, что **условие** принимает значение ИСТИНА. В противном случае предполагается, что условие имеет значение ЛОЖЬ.

При выполнении цикла **while**:

- вычисляется значение управляющего выражения;
- если значение выражения не равно нулю, то выполняется оператор, представляющий собой тело цикла, если он существует;
- в противном случае выполнение цикла **while** прекращается и продолжается выполнение программы, начиная с оператора, следующего за **while**;
- если начальное значение управляющего выражения равно ЛОЖЬ (0), то тело цикла не выполняется ни разу.

Если в качестве тела цикла используется составной оператор, то оператор цикла необходимо оформлять следующим образом:

```
while (<условие>){  
    <оператор 1>  
    ...  
    <оператор n>  
}
```

Ключевое слово **while**, условие и открывающая фигурная скобка составного оператора располагаются в отдельной строке. Операторы тела цикла сдвигаются вправо на 2-3 позиции, а закрывающая скобка также располагается в отдельной строке, находясь в позиции, с которой начинается **while**.

Пример. Вычислить сумму целых чисел от 1 до 100.

```
int main() {  
    double x, s;  
    x=0;  
    s=0;  
    while (x<100){  
        x=x+1;  
        s=s + x;  
    }  
    printf("s= %f\n", s);  
}
```

Тело цикла **while** может быть пустым. Например, цикл

```
while ( getchar() != 'A' );
```

выполняется до тех пор, пока пользователь не введет символ 'A'.

3.2 Оператор цикла **do**

Общая форма оператора цикла **do** имеет следующий вид:

```
<оператор do>: do [<оператор>]; while (<условие>);
```

Оператор, представляющий собой тело цикла, может быть пустым оператором, единственным оператором или составным оператором. Если оператор опускается, то символ «;» необходимо сохранить.

Условие (управляющее выражение) может быть любым допустимым выражением. Если значение выражения не равно нулю, то считается, что **условие** принимает значение ИСТИНА. В противном случае предполагается, что условие имеет значение ЛОЖЬ.

При выполнении цикла **do**:

- выполняется оператор, представляющий собой тело цикла, если он существует;
- вычисляется значение управляющего выражения;
- если значение выражения не равно нулю, то снова выполняется тело цикла;
- в противном случае выполнение цикла **do** прекращается и продолжается выполнение программы, начиная с оператора, следующего за конструкцией **while** (<условие>);
- тело цикла всегда выполняется, по крайней мере, один раз.

Если в качестве тела цикла используется составной оператор, то оператор цикла необходимо оформлять следующим образом:

```
do {  
    <оператор 1>  
    ...  
    <оператор n>}  
while (<условие>)
```

Ключевое слово **do** и открывающая фигурная скобка составного оператора располагаются в отдельной строке. Операторы тела цикла сдвигаются вправо на 2-3 позиции, а закрывающая скобка и **while** (<условие>) располагается в отдельных строках без сдвига.

Пример. Методом последовательных приближений вычислить значение корня уравнения $x = \cos(x)$, используя рекуррентную формулу $x_{n+1} = \cos(x_n)$ с начальным приближением $x_0 = 1$. Вычисления прекратить при выполнении условия $|x_{n+1} - x_n| < 10^{-4}$.

```
int main() {
    double x0,x1;
    x1=1;
    do {
        x0=x1;
        x1=cos(x0); }
    while (abs(x0-x1)>=1e-4);
    printf("x= %f\n",x1);
    system ("pause");
}
```

4. Операторы break и continue

Иногда бывает удобно выйти из цикла не по результату проверки, осуществляемой в начале или в конце цикла, а каким-то другим способом. Такую возможность для циклов **for**, **while** и **do-while**, а также для переключателя **switch** предоставляет оператор **break**, который вызывает немедленный выход из самого внутреннего из объемлющих его циклов или переключателей.

Оператор **continue** вынуждает ближайший объемлющий его цикл начать следующий шаг итерации. Для **while** и **do-while** это означает немедленный переход к проверке условия, а для **for** – к приращению шага. Оператор **continue** можно применять только к циклам, но не к оператору выбора **switch**.

5. Пример выполнения задания

Задача «а»

Вычислить сумму $s = \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i+1}}{2i(2i+1)!}$, для $x=2$ значение суммы $s = -0.8647$.

Для вычисления очередного элемента суммы выведем рекуррентную формулу, связывающую два элемента с последовательными номерами a_i и a_{i-1} . Запишем общую формулу для i -го элемента

$$a_i = \frac{(-1)^i x^{2i+1}}{2i(2i+1)!}.$$

Аналогично для $i-1$ -го элемента

$$a_{i-1} = \frac{(-1)^{i-1} x^{2(i-1)+1}}{2(i-1)(2(i-1)+1)!} = \frac{(-1)^{i-1} x^{2i-1}}{2(i-1)(2i-1)!},$$

т.е. общая форма элемента a_{i-1} получается путем подстановки значения $i-1$ вместо i формулу для a_i . Вычислим отношение

$$\frac{a_i}{a_{i-1}} = \frac{(-1)^i x^{2i+1}}{2i(2i+1)!} \cdot \frac{2(i-1)(2i-1)!}{(-1)^{i-1} x^{2i-1}} = -\frac{x^2(i-1)}{i(2i+1)2i} = -\frac{x^2(i-1)}{2i^2(2i+1)}.$$

При вычислении отношения учитывалось, что факториалы чисел определяются следующим образом: $i! = i(i-1)(i-2)\dots 2 \cdot 1$, а $(2i)! = (2i)(2i-1)(2i-2)\dots 2 \cdot 1$, $(2i+1)! = (2i+1)(2i)(2i-1)(2i-2)\dots 2 \cdot 1$ и $(2i-1)! = (2i-1)(2i-2)\dots 2 \cdot 1$.

Таким образом, получим, что следующий элемент суммы можно вычислить по известному значению предыдущего элемента с помощью формулы:

$$a_i = -a_{i-1} \frac{x^2(i-1)}{2i^2(2i+1)}.$$

В качестве начального значения, необходимого для вычисления по рекуррентной формуле, возьмем a_1 , выражение для которого легко получить, подставив 1 вместо i в общую форму i -го элемента:

$$a_1 = -\frac{x^3}{12}.$$

Для проверки правильности вывода формулы вычислим a_2 по рекуррентной формуле.

$$a_2 = -a_1 \frac{x^2(2-1)}{2 \cdot 2^2(2 \cdot 2 + 1)} = -\frac{x^3}{12} \frac{x^2}{40} = \frac{x^5}{480}.$$

$$\text{И на основе общей формы } a_2 = \frac{(-1)^2 x^{2 \cdot 2 + 1}}{2 \cdot 2(2 \cdot 2 + 1)!} = \frac{x^5}{4 \cdot 5!} = \frac{x^5}{480}.$$

Тесты

Таблица трассировки для $x=2$

i	a	s
1	-0,666667	-0,666667
2	0,066667	-0,600000
3	-0,004233	-0,604233
4	0,000176	-0,604056
5	-0,000005	-0,604062
6	0,000000	-0,604061
7	0,000000	-0,604061

```
#include "stdafx.h"
#include <iostream>
int main() {
    double s,a,x;
    int i;
    printf("Input x=");
    scanf("%lf",&x);
    s=0;
    a=-x*x*x/12;
    i=1;
    while (abs(a)>1e-7) {
        s=s+a;
        i=i+1;
        a=-a*x*x*(i-1)/(2.0*i*i*(2*i+1));
    }
    printf("Sum= %f\n",s);
    system ("pause");
}
```

Задача «б»

Определить наименьший номер элемента последовательности, заданной рекуррентной формулой:

$$A_{n+1} = A_n + A_{n-1}, \quad A_0 = 1, \quad A_1 = 1,$$

для которого выполняется условие $A_n > 100$. Значение номера $n=11$.

Тесты

Таблица трассировки

n	a2	a1	a0
2	2,0000000	1,0000000	1,0000000
3	3,0000000	2,0000000	1,0000000
4	5,0000000	3,0000000	2,0000000
5	8,0000000	5,0000000	3,0000000
6	13,0000000	8,0000000	5,0000000
7	21,0000000	13,0000000	8,0000000
8	34,0000000	21,0000000	13,0000000
9	55,0000000	34,0000000	21,0000000
10	89,0000000	55,0000000	34,0000000
11	144,0000000	89,0000000	55,0000000

Программа

```

int main() {
    double a0,a1,a2;
    int n;
    a1=1;
    a2=1;
    n=1;
    do {
        n=n+1;
        a0=a1;
        a1=a2;
        a2=a1+a0;}
    while (a2<=100);
    printf("n= %i\n",n);
    system ("pause");
}

```

6. Варианты заданий

Вариант 1.

а) $\sum_{i=1}^{\infty} \frac{(-1)^i x^i}{2i(2i)!}$, $s=-0.4601$

б) $A_{n+1} = \frac{A_n + A_{n-1}}{2}$, $A_1=1$, $A_2=10$, $|A_n - 7| < 10^{-4}$, $n=16$

Вариант 2.

а) $\sum_{i=1}^{\infty} \frac{(-1)^i x^{2i}}{3i(2i)!}$, $s=-0.5649$

б) $A_{n+1} = 1.99A_n - A_{n-1}$, $A_1=1$, $A_2=1$, $A_n > A_{n-1}$, $n=33$

Вариант 3.

а) $\sum_{i=0}^{\infty} \frac{(-1)^i x^{3i+1}}{(3i+1)!}$, $s=1.3584$

б) $A_{n+1} = \frac{A_n - A_{n-1}}{2}$, $A_1=1$, $A_2=2$, $|A_n| < 10^{-3}$, $n=18$

Вариант 4.

а) б) $\sum_{i=0}^{\infty} \frac{(-1)^i x^i}{(2i)!}$, $s=0.1559$

б) $A_{n+1} = 2.01A_n - A_{n-1}$, $A_1=A_2=1$, $|A_n| > 2$, $n=14$

Вариант 5.

$$a) \sum_{i=1}^{\infty} \frac{x^i}{i!i}, s=3.6839$$

$$б) A_{n+1}=nA_n+A_{n-1}, A_1=0, A_2=0.1, |A_n|>100, n=7$$

Вариант 6.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{i!(2i+1)}, s=0.8821$$

$$б) A_{n+1}=A_n+1.1A_{n-1}, A_0=0.1, A_1=0.2, A_n>100, n=14$$

Вариант 7.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)!(2i+1)}, s=1.6054$$

$$б) A_{n+1}=\frac{A_n}{n}+A_{n-1}, A_0=1, A_1=1.1, A_n>5, n=14$$

Вариант 8.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!2i}, s=-0.8474$$

$$б) A_{n+1}=\frac{A_n+A_{n-1}}{n}, A_0=A_1=1, A_n<10^{-3}, n=13$$

Вариант 9.

$$a) \sum_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)(i+1)!}, s=6.1062$$

$$б) A_{n+1}=e^{-(A_n+A_{n-1})}, A_0=0, A_1=1, |A_n-A_{n-1}|<10^{-4}, n=18$$

Вариант 10.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)(i+2)!}, s=0.7231$$

$$б) A_{n+1}=A_n+e^{-A_{n-1}}, A_0=10, A_1=1, |A_n-3|<0.01, n=16$$

Вариант 11.

$$a) \sum_{i=0}^{\infty} \frac{x^{2i}}{(2i)!}, s=3.7622$$

$$б) A_{n+1}=\operatorname{tg} \frac{A_n+A_{n-1}}{2}, A_0=A_1=1, |A_n|<0.2, n=10$$

Вариант 12.

$$a) \sum_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)!}, s=3.6269$$

$$б) A_{n+1}=A_n+nA_{n-1}, A_0=0, A_1=1, A_n>6000, n=10$$

Вариант 13.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^i}{(2i+3)i!}, s=0.1157$$

$$б) A_{n+1}=\sin A_n+A_{n-1}, A_0=0, A_1=1, A_n>5, n=11$$

Вариант 14.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i}}{(2i)!}, s=-0.4161$$

$$б) A_{n+1}=\lg |A_n-A_{n-1}|, A_0=1, A_1=100, |A_n+1.42|<10^{-4}, n=13$$

Вариант 15.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{2i+1}}{(2i+1)!}, s=0.9093$$

$$б) A_{n+1}=A_n+e^{-A_{n-1}}, A_0=10, A_1=1, A_n>3, n=16$$

Вариант 16.

$$a) \sum_{i=0}^{\infty} \frac{x^{3i+2}}{(3i)!}, s=9.6946$$

$$б) A_{n+1} = \sin A_n + A_{n-1}, A_0=1, A_1=2, A_n < 0, n=10$$

Вариант 17.

$$a) \sum_{i=0}^{\infty} \frac{x^{3i+3}}{(2i+2)!}, s=7.4890$$

$$б) A_{n+1} = nA_n - \frac{1}{n}A_{n-1}, A_0=0, A_1=1, A_n > 10^4, n=9$$

Вариант 18.

$$a) \sum_{i=1}^{\infty} \frac{x^{2i-1}}{(2i+1)!}, s=0.4067$$

$$б) A_{n+1} = \frac{\cos A_n}{n} + A_{n-1}, A_0=1, A_1=0.3, A_n \leq -0.5, n=15$$

Вариант 19.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{i+1}}{i!(i+1)}, s=-1.1353$$

$$б) A_{n+1} = n(n+1)A_n + n^2 A_{n-1}, A_0=1, A_1=2, A_n > 10^8, n=8$$

Вариант 20.

$$a) \sum_{i=0}^{\infty} \frac{x^{2i+2}}{i!(2i+1)}, s=32.9053$$

$$б) A_{n+1} = \frac{n}{n+1}A_n - A_{n-1}, A_0=1, A_1=10, |A_n| > 10, n=9$$

Вариант 21.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i-2}}{(2i)!(2i+2)}, s=-0.0998$$

$$б) A_{n+1} = \frac{n}{n+1}A_n + \frac{1}{n}A_{n-1}, A_0=0.1, A_1=1, A_n > 1, n=10$$

Вариант 22.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{i-1}}{(2i)!(i+1)}, s=-0.2236$$

$$б) A_{n+1} = \frac{1}{n} \operatorname{tg} A_n - A_{n-1}, A_0=0, A_1=0.1, |A_n| < 10^{-3}, n=7$$

Вариант 23.

$$a) \sum_{i=0}^{\infty} \frac{(-1)^i x^{3i+3}}{(2i+1)!}, s=0.8714$$

$$б) A_{n+1} = e^{-n}A_n + e^n A_{n-1}, A_0=1, A_1=1, A_n > 10^4, n=7$$

Вариант 24.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i+3}}{i(i+1)!}, s=-9.7017$$

$$б) A_{n+1} = A_n - \frac{e^{-n}}{n}A_{n-1}, A_0=2, A_1=2.9887, |A_n - 2| < 10^{-5}, n=11$$

Вариант 25.

$$a) \sum_{i=1}^{\infty} \frac{(-1)^i x^{2i-2}}{(2i)!(2i+2)}, s=-0.0998$$

$$б) A_{n+1} = \frac{n+2}{n}A_n - \operatorname{tg} A_{n-1}, A_0=0, A_1=1, |A_n| < 0.2, n=8$$