**Семинар 3. Особенности обработки циклических процессов.**

**Циклическими называются процессы, при которых некоторая последовательность действий повторяется более одного раза.**

**В зависимости от способа организации самого процесса и выхода из него, циклы делятся на**

**- Счетные.**

**- Итерационные.**

**-Поисковые. (см. лекции презентацию 2).**

**Счетные циклы.**

Счетными называются циклические процессы, при которых количество повторений заранее известно или легко вычисляется.

**for (<Выражение 1>;<Выражение 2>;<Выражение 3>)<Оператор>;**

**Выражение 1 –** инициализирующее выражение; представляет собой последовательность описаний, определений и выражений, разделенных запятыми. Выполняется только один раз в начале цикла и задает начальные значения переменным цикла. Может отсутствовать, при этом точка с запятой остается.

**Выражение 2 –**выражение условия; определяет предельное значение параметра цикла. Может отсутствовать, при этом точка с запятой остается.

**Выражение 3 –** список выражений, которые выполняются на каждой итерации цикла после тела цикла, но до следующей проверки условия. Обычно определяют изменение параметра цикла. Может отсутствовать

**Оператор –** тело цикла. Может быть любым оператором С++, блоком операторов (тело цикла содержит более одного простого оператора) или может отсутствовать.

 **Пример полного оператора счетного цикла**

 **for(int i=0, s=0;i<k;i++)s+=i;**

На схеме алгоритма счетный цикл изображается так.

Счетный цикл дает возможность автоматически менять параметр цикла (например, i) и обеспечивать выход из цикла при достижении предельного значения параметра цикла (i=k).

Однако, для получения оптимальной программы, она должна обладать не очень высокой вычислительной сложностью. Например, так как операции умножения и деления являются самыми трудоемкими, целесообразно свести эти операции к сложению и вычитанию, возведение в степень – к умножению, используя рекуррентные соотношения (например, **x^k=x^(k-1)\*x**, или **n!=(n-1)!\*n**).

 При работе со сложными дробями, у которых числитель и знаменатель меняются по разным закономерностям, лучше разделить вычисления числителя и знаменателя.

При нахождении суммы n членов ряда использовать для вычисления очередного члена ряда рекуррентные соотношения.

Рассмотрим несколько примеров.

**Пример 1. Определить значение n членов дроби**

**D=(1/2)\*(3/4)\*(5/6)\*(7/8)……**

Как видим, в числителе значения меняются от 1 с шагом 2(нечетные числа), а в знаменателе тоже с шагом 2 от 2 (четные).

Есть 2 решения, числитель и знаменатель связать со значением индекса

При i=1 до n

Числитель – это 2\*i-1, а знаменатель 2\*i. Тогда очередной член дроби равен

(2\*i-1)/(2\*i).

Однако, нам нужно вычислять в цикле n раз два произведения. Это увеличивает вычислительную сложность, тем более, для проверки правильности решения в процессе хорошо бы выводить текущие значения участников вычисления. А это еще 2 умножения.

Поэтому, берем второй вариант.

Для числителя вводится своя переменная ch, которая вначале берется =-1.

Для знаменателя переменная zn=0. Для накопления значения дроби введем переменную p=1 (так как это произведение. После входа в цикл, значения числителя и знаменателя увеличиваются на 2, а потом p домножается на значение ch/zn и параллельно выводится на экран.

**Схема алгоритма**

**Текст программы**

 **#include <stdio.h>**

**#include <conio.h>**

**// 1/2\*3/4\*5/6\*7/8......**

**void main()**

**{float ch,zn,p;**

 **int i,n;**

 **puts("input n>1");**

 **scanf("%d",&n);**

 **ch=-1;**

 **zn=0;**

 **p=1.0;**

 **for(i=1;i<=n;i++)**

 **{ ch=ch+2;**

 **zn=zn+2;**

 **p=p\*ch/zn;**

 **printf("%2.0f/%2.0f \* ",ch,zn);**

 **}**

 **printf("\nZnachenie druby iz %5d elementov=%8.5f\n",n,p);**

 **getch();**

**}**

**Пример работы программы**

**Пример 2.** **Определить значение n членов дроби**

**D=(1/2)\*(3/2)\*(5/2)\*(7/4)\*(9/4)\*(11/4)\*(13/6)……**

Как видим, в числителе значения меняются от 1 с шагом 2(нечетные числа), а в знаменателе тоже происходит изменение с шагом 2 от 2 (четные), однако, эти изменения происходят не на каждой итерации, а только при ш=1, 4, 7 и т.д.

В этом случае лучше сразу выбрать решение с раздельным изменением числителя и знаменателя без использования индекса цикла.

Для числителя вводится своя переменная ch, которая вначале берется =1, а далее на каждой итнрации увеличивается на 2.

Для знаменателя переменная zn=2.

Для накопления значения дроби введем переменную d=1 (так как это произведение). После входа в цикл, вычисляется значениe деления числителя и знаменателя и на эту величину домножается дробь d. Однако, перед этим, необходимо проверять, не пора ли увеличить знаменатель на 2. Если условие выполнено, а это (i-1)%3=0 и это не первый член (i!=1) увеличиваются на 2, а потом p домножается на значение ch/zn и параллельно выводится на экран. Затем числитель ch увеличивается на 2.

Кроме того, на каждой итерации выводятся значения числителя и знаменателя.

Схема алгоритма похожа на предыдущую.

**Текст программы приведен ниже.**

**#include <stdio.h>**

**#include <conio.h>**

**// D=(1/2)\*(3/2)\*(5/2)\*(7/4)\*(9/4)\*(11/4)\*(13/6)……**

**int i,n;**

**float d,ch,zn;**

**int main(int argc, char\* argv[])**

**{**

 **puts("Input n");**

 **scanf("%d",&n);**

 **for(i=1,d=1,ch=1,zn=2;i<=n;i=i+1)**

 **{ if(((i-1)%3==0)&&(i!=1))**

 **zn=zn+2;**

 **d\*=ch/zn;**

 **printf("%3.0f/%3.0f \*",ch,zn);**

 **ch=ch+2;**

 **}**

 **printf("\n\ Pri n=%3d drob= %8.3f \n",n,d);**

 **getch();**

 **return 0;**

**}**

**Пример работы программы**

**Пример 3.Определить сумму последовательности**

**s= ∑ (2k/(1-ak2+bk2) для k=1 до n**

**n задается с клавиатуры**

При этом a,b меняются так:

**a1=1;b1=1; при к=1**

**ak=2ak-1+3bk-1 ;при к>1**

**bk=2ak-1+bk-1 ;при к>1**

Для решения этой задачи используется несколько приемов.

1. Чтобы не возводить на каждой итерации 2 в k степень, воспользуемся рекуррентной формулой. Мы знаем, что 20=1, а 2k=2k-1\*2. Это позволяет нам, сохранив предыдущую степень числа 2 просто домножить ее на 2. Для этого введем переменную st2 степень двойки. Начальное st2=1.
2. Очередные значения a,b находим по приведенным рекуррентным формулам. Для этого нам понадобятся 4 переменных a,b текущие значения, a1,b1 предыдущие значения.
3. Для возведения чисел a1и b1в квадрат использовать операцию умножения, так как стандартной функции возведения в квадрат нет, а использование функции pow, увеличивает время выполнения программы.

Текст программы приведен ниже

**// Определить сумму проследовательности , члены которой определяются**

**// так**

**// s= ∑ (2^k/(1-(ak)^2+(bk)^2) для k=1 до n**

**// a1=1;b1=1; при к=1**

**// ak=2ak-1+3bk-1 ; при к>1**

**// bk=2ak-1+bk-1 ;при к>1**

**#include <stdio.h>**

**int main()**

**{float a1,a=1,b1,b=1,sum=0,st2=1,znam;**

 **int n;**

 **puts("Input integer n>1");**

 **scanf("%d",&n);**

 **for(int i=1;i<=n;i++)**

 **{ st2=st2\*2;**

 **a1=a;**

 **b1=b;**

 **znam=(1+a1\*a1+b1\*b1);**

 **sum=sum+st2/znam;**

 **printf("(%3.0f/%7.0f )+",st2,znam);**

 **a=3\*b1+2\*a1;**

 **b=2\*a1+b1;**

 **}**

 **printf("\n sum=%10.8f\n",sum);**

 **return 0;**

**}**

**Пример выполнения программы**

**Пример 4. Опрределить значение n членов последовательночти**

**s=sin(x)+sin(x^2)+sin(x^3)........**

**n и x задаются пользователем и вводятся с клавиатуры.**

Для решения этой задачи для вычисления очередного членана последовательности на каждой итерации необходимо знать очередную степень. x.
Мы знаем, x0=1;

 xn=xn-1\*x

Для хранения очередной степени x используем переменную xi, начальное значение которой xi=1. Далее в цикле на каждой итерации определяем очередной член последовательности

r=sin(xi) и добавляем его к начальной сумме s=0.

Параллельно выводим все промежуточные данные.

**Текст программы.**

**#include <stdio.h>**

**#include <conio.h>**

**#include <math.h>**

**//s=sin(x)+sin(x^2)+sin(x^3)........**

**int main()**

**{**

 **float x,r,xi=1,s;**

 **int i,n;**

 **puts("input n");**

 **scanf("%d", &n);**

 **puts("input x");**

 **scanf("%f", &x);**

 **s=0;**

 **puts(" N X^i r\n----------------------------------------------");**

 **for(i=1;i<=n;i++)**

 **{xi=xi\*x;**

 **r=sin(xi);**

 **printf("%4d %14.2f %12.6f \n",i,xi,r);**

 **s=s+r;**

 **}**

 **puts("------------------------------------------");**

 **printf ("s=%17.7f" , s);**

**printf(" i=%3d" , i-1);**

**getch();**

** return 0;**

 }

**Примеры работы программы**



**Пример 5. Определить сумму n членов ряда**

 **s= ∑ (1/(i2) для i=1 до n , n вводится с клавиатуры**

Решение этой задачи на каждой итерации цикла требует вычисления очередного члена ряда r=1/i/i;

Так как выражение простое, использовать рекуррентные соотношения не целесообразно. Нам нужны три переменные i,r,s=0.

**Текст программы**

**#include <stdio.h>**

**#include <math.h>**

**void main(int argc, char\* argv[])**

**{ float s,r;int n,i;**

**puts("Input n");**

**scanf("%d",&n);**

**printf("n=%4d\n",n);**

**s=0;**

**puts(" i r s");**

**puts("-----------------------------------------------");**

 **for(i=1;i<=n;i++)**

 **{**

 **r=1.0/i/i;**

 **s+=r;**

 **printf(" %4d %16.15f %16.15f\n",i,r,s);**

 **}**

 **puts("-----------------------------------------------");**

 **printf("Result= %16.15f\n n=%12d.\n", s,n);**

 **}**

**Пример работы программы**

****

**Примеры задач для домашнего задания**

**1. Определить значение дроби d= (1/2)\*(3/2)\*(5/4)\*(7/4)\*(9/6)……**

**2. Вычислить S=∑(ai-bi)2 ;**

**Где: ai=i, если i - нечетное**

 **ai=i/2, если i – четное**

 **bi=i2, если i - нечетное**

 **bi=i3 , если i - четное**

1. **Вычислить сумму n членов ряда**

 **S=∑(-2)i/ i! ;**

**при i=1 до n, n вводится с клавиатуры**