Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра И5 «Информационные системы и программная инженерия»

**Практическая работа №2**по дисциплине «Информатика: Основы программирования»  
на тему «Ветвления и циклы»

Выполнил:  
Студент Альков В.С.  
Группа И407Б  
  
Преподаватель:  
Першин Д.В.

Санкт-Петербург  
2020 г.

1. Вычислить значение функции используя условную операцию ?:

*Исходные данные:*

Аргументы функции a и b. Так как значения a и b могут быть любыми, объявим соответствующие переменные типа double.

*Результирующие данные:*

Значение функции f , соответствующая переменная тоже будет типа double.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| 1 формула: a=5.1, b=2.2 | 78.03 | 78.03 |
| 2 формула: a=4.4, b=2.2 | 2 | 2 |
| 3 формула: a=4.4, b=0 | 3.4 | 3.4 |
| 3 формула: a=-0.5, b=4 | 2.5 | 2.5 |

Схема программы

начало

конец

вывод f

Ввод a,b

b≠0 и a>0

a>5

f=a/b

f=3\*a\*a

f=b+a-1

да

да

нет

нет

Текст программы

#include <stdio.h>

#include <float.h>

#include <math.h>

int main()

{

/\* объявление переменных \*/

double f, a, b;

/\* ввод с клавиатуры вещественных чисел и запись их в переменные: a, b \*/

scanf("%lf%lf",&a,&b);

/\* вычисление значения с помощью тернарной условной операции и запись его

в переменную f. Если a>5, то f=3\*a\*a, если 0 <a<=5 и b!=0, то f=a/b, в

других случаях, f=b+a-1\*/

f = a > 5 ? 3\*a\*a : fabs(b)>=FLT\_EPSILON && a>0 ? a/b : b+a-1;

/\* вывод значения функии f\*/

printf("%lf",f);

return 0;

}

1. Вычислить значение функции

*Исходные данные:*

Аргументы функции a и b. Так как значения a и b могут быть любыми, объявим соответствующие переменные типа double.

*Результирующие данные:*

Значение выражения d , соответствующая переменная тоже будет типа double.

*Предварительные вычисления:*

Чтобы можно было вычислить значение функции, должны быть выполнены следующие условия: b≠0, ненулевой знаменатель дроби,, неотрицательное подкоренное выражение, tg определен, то есть cos(a/b) ≠ 0

*Вспомогательные переменные:*

v1 – подкоренное выражение, v2 – знаменатель дроби; обе переменные типа double.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| a=0,b=1 | 0 | 0 |
| a=3,b=0 | неопределен  неопределен  Результат неопределен | Программа возвращает -1, что значит, исходные данные не принадлежат ООФ |
| a=3,b=3 | Знаменатель равен 0  Результат неопределен | Программа возвращает -1, что значит, исходные данные не принадлежат ООФ |
| a=1.5708,b=1 | неопределен  Результат неопределен | Программа возвращает -1, что значит, исходные данные не принадлежат ООФ |

Схема программы:

начало

конец

вывод d

ввод a,b

b=0

да

нет

конец

v1 < 0

нет

да

конец

да

v2 =0

конец

да

нет

нет

конец

Текст программы

#include <stdio.h>

#include <math.h>

#include <float.h>

int main()

{

/\* объявление переменных \*/

double d,v1,v2,a,b;

/\* ввод с клавиатуры вещественных чисел и запись их в переменные: a, b \*/

scanf("%lf%lf",&a,&b);

/\* проверка b на ноль, если да, то выход\*/

if (fabs(b)<FLT\_EPSILON) return -1;

/\* подкоренное выражение\*/

v1 = sin(a)\*sin(a) + 12 \* fabs(log(fabs(b)));

/\* проверка подкоренного выражения на отрицательность, если да, то выход

\*/

if (v1 < 0) return -1;

/\* проверка cos(a/b) на ноль, если да, то выход\*/

if (cos(a/b) == 0) return -1;

/\* знаменатель дроби\*/

v2 = (b-a)\*(b-a)\*exp(tan(a/b));

/\* проверка знаменателя дроби на ноль, если да, то выход\*/

if (fabs(v2)< FLT\_EPSILON) return -1;

/\*вычисление значения выражения \*/

d = (-sin(a) + v1)/v2;

/\* вывод полученного значения \*/

printf("d=%lf\n", d);

return 0;

}

1. Даны два числа. Если они оба положительны, то большее из  
   них заменить их средним арифметическим; если оба отрицательны,  
   поменять знак у меньшего из них; если числа имеют разные знаки, то  
   каждое из них удвоить. Если хотя бы одно из чисел равно нулю, изменять их не требуется.

*Исходные данные:*

Целые a и b, тип int.

*Результирующие данные:*

Целые a и b, тип int.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| a=0,b=0 | a=0,b=0 | a=0,b=0 |
| a=1,b=0 | a=1,b=0 | a=1,b=0 |
| a=0,b=1 | a=0,b=1 | a=0,b=1 |
| a=-1,b=0 | a=-1,b=0 | a=-1,b=0 |
| a=0,b=-1 | a=0,b=-1 | a=0,b=-1 |
| a=2,b=4 | a=2,b=3 | a=2,b=3 |
| a=4,b=2 | a=3,b=2 | a=3,b=2 |
| a=-2,b=4 | a=-4,b=8 | a=-4,b=8 |
| a=4,b=-2 | a=8,b=-4 | a=8,b=-4 |
| a=-2,b=-4 | a=-2,b=4 | a=-2,b=4 |
| a=-4,b=-2 | a=4,b=-2 | a=4,b=-2 |

Схема программы:

нет

начало

вывод a, b

ввод a,b

a≠0 и b≠0

a = a\*2

b = b\*2

да

нет

a > 0

нет

да

b > 0

да

a > b

конец

да

нет

нет

a < b

b > 0

a = -a

a=a\*2

b=b\*2

b = -b

нет

да

да

Текст программы

#include <stdio.h>

int main()

{

/\* объявление переменных \*/

int a,b;

/\* ввод с клавиатуры вещественных чисел и запись их в переменные: a, b \*/

scanf("%d%d", &a, &b);

/\* проверка a и b на не ноль, если нет, то изменять не требуется\*/

if (a!=0 && b !=0)

/\* проверка a на положительность\*/

if (a>0)

/\* a>0, проверка b на положительность\*/

if (b>0)

/\* b>0, определяем большее из a и b\*/

if (a>b)

/\* a>b, заменяем a среднеарифметическим\*/

a = (a + b) / 2;

else

/\* b>a, заменяем b среднеарифметическим\*/

b = (a + b) / 2;

/\* b<0, a>0, умножаем оба на 2\*/

else

{

a\*=2;

b\*=2;

}

else

if (b>0)

/\* b>0, a<0, умножаем оба на 2\*/

{

a\*=2;

b\*=2;

}

else

if (a<b)

/\* a<b<0, меняем знак a\*/

a=-a;

else

/\* b<a<0, меняем знак b\*/

b=-b;

/\* вывод a, b\*/

printf("%d %d", a, b);

return 0;

}

1. Дано натуральное число n. Вычислить n сомножителей произведения

. Использовать управляющую инструкцию *for*.

*Исходные данные:*

n – кол-во сомножителей, целое, тип int .

*Результирующие данные:*

res - результат умножения, тип double.

*Вспомогательные переменные:*

*i – счетчик цикла, тип int*

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *n=2* | 1.3333 | 1.333333 |
| *n=*7 | 1.6718 | 1.671837 |
| *n=*8 | 1.4860 | 1.486077 |

Схема программы:

начало

вывод res

ввод n

res = res\*i/(i+1)

res =res\*( i+1)/i

нет

да

i дел на 2

конец

Цикл 1

i=2,n ш. 1

Цикл 1

res = 2

Текст программы

#include <stdio.h>

int main()

{

/\* объявление переменных \*/

int n,i;

double res=2.;

/\* ввод с клавиатуры вещественного числа и запись его переменную: n \*/

scanf("%d",&n);

/\* цикл for с шагом в 1, от 2 до n, кол-во повторов n-2+1\*/

for (i=2;i<=n;i++)

{

/\* проверка i на четность\*/

if (i%2 == 1)

{

/\* i нечетное, выражаем число, приводим к double, умножаем res на число\*/

res \*= (double)(i+1)/i;

}

else

{

/\* i четное, выражаем число, приводим к double, умножаем res на число\*/

res \*= (double)i/(i+1);

};

};

/\*выводим результат\*/

printf("%lf",res);

return 0;

}

1. Представить натуральное число N в виде произведения простых сомножителей. Простыми называются сомножители, которые  
   нельзя в свою очередь разложить на сомножители.

*Исходные данные:*

a – целое, тип int .

*Результирующие данные:*

Последовательный вывод множителей через printf

*Вспомогательные переменные:*

i – множитель числа, тип int

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *36* | 36 = 1 \* 2 \* 2 \* 3 \* 3 | 36 = 1 \* 2 \* 2 \* 3 \* 3 |
| 245 | 245 = 1 \* 5 \* 7 \* 7 | 245 = 1 \* 5 \* 7 \* 7 |
| 624 | 624 = 1 \* 2 \* 2 \* 2 \* 2 \* 3 \* 13 | 624 = 1 \* 2 \* 2 \* 2 \* 2 \* 3 \* 13 |

Схема программы:

начало

ввод a

a = a/i

i = i +1

нет

да

a дел на i

конец

пока

a≠1

a

i = 2

вывод a = 1

вывод \* i

i = i - 1

Текст программы

#include <stdio.h>

int main()

{

/\* объявление переменных \*/

int i=2,a;

/\* ввод с клавиатуры вещественного числа и запись его переменную: a \*/

scanf("%d",&a);

/\* вывод a в формате: число = 1\*/

printf("%d = 1",a);

/\* цикл while с предусловием, повторять пока a≠1, потому что числа делим

без остатка, в результате a станет = 1, тогда цикл завершиться\*/

while (a != 1)

{

/\* делится ли a на i без остатка? \*/

if (a%i==0)

{

/\* да, делим a на i, выводим множитель, вычитаем из i один, чтобы

не потерять следующий возможный множитель\*/

a /= i;

printf(" \* %d",i);

i--;

};

/\* увеличивем множитель на 1 \*/

i++;

};

return 0;

}

1. Поменять местами цифры старшего и младшего разрядов  
   данного натурального числа (например, из числа 3872 получится  
   2873).

*Исходные данные:*

n – целое, тип int .

*Результирующие данные:*

res - целое, тип int .

*Вспомогательные переменные:*

i – разряд цифры, b – последняя цифра n, c – текущая последняя цифра n в цикле, у всех тип int.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *123* | 321 | 321 |
| 56785 | 56785 | 56785 |
| 235150 | 35152 | 35152 |

Схема программы:

начало

ввод n

i=i\*10

n=n/10

c=n%10

res = res + c\*i

конец

пока

n/10≠0

n

i = 1

res = 0

вывод res

res=res + b\*i-c\*(i-1)

b = n%10

Текст программы

#include <stdio.h>

int main()

{

/\* объявление переменных \*/

int i=1,n,b,c,res=0;

/\* ввод с клавиатуры вещественного числа и запись его переменную: n \*/

scanf("%d",&n);

/\*b=последняя цифра n\*/

b=n%10;

/\* цикл while с предусловием, повторять пока n/10≠0, для того чтобы

не потерять первую цифру числа n \*/

while(n/10 != 0)

{

/\*задаем разряд текущей цифры\*/

i\*=10;

/\*обрезаем n, берем от него последнюю цифру \*/

c=(n/=10)%10;

/\*прибавляем цирфу \* разряд к результату \*/

res+=c\*i;

};

/\* прибавляем к результату b\*i, то есть последнюю цифру исходного числа \*

высший порядок, полученный в результате работы цикла, вычитаем c\*(i-1),

то есть первую цифру исходного числа \* (высший порядок – 1), - 1, для

того чтобы цифра перешла в разряд единиц \*/

res+=b\*i-c\*(i-1);

/\*выводим результат\*/

printf("%d",res);

return 0;

}

1. Вычислить значение суммы бесконечного ряда

с точностью до члена ряда, меньшего ε=10-4 и значение функции (для проверки) , учесть, что 0,1≤ x ≤1.

*Исходные данные:*

a,x – аргументы функции, тип double.

*Результирующие данные:*

s - значение суммы, , тип double.

*Вспомогательные переменные:*

n – индекс слагаемого, тип int, f – значение текущего слагаемого, тип double, v1 – значение числителя, тип double, v2 – значение знаменателя, тип int.

*Предварительные вычисления:*

n-ное слагаемое , можно рассчитывать отдельно числитель и знаменатель для каждого , знаменатель – n!, будем умножать v2 на n в каждом проходе, чтобы получить знаменатель текущего , числитель - , будем умножать v1 на в каждом проходе, чтобы получить значение числителя текущего . Для вычисления итоговой суммы будем прибавлять значение текущего к s в цикле.

Схема программы:

начало

ввод x,a

конец

x и a

пока

x < 0.1 или x > 1 или a=0

s=1

v1=1

n=1

v2=1

вывод s

n=1

пока

f

v1 = v1\*x\*ln(a)

v2 = v2 \* n

f = v1/v2

s = s + f

n = n + 1

Текст программы

#include <stdio.h>

#include <math.h>

int main()

{

/\* объявление переменных \*/

double f, s=1., v1=1., x, a;

int n=1,v2=1;

/\*цикл while с постусловием, контролирует ввод, чтобы 0.1≤x≤1 и a≠0\*/

do

/\*ввод с клавиатуры вещественных чисел и запись их в переменные:x,a\*/

scanf("%lf%lf", &x, &a);

while (x < 0.1 || x > 1.0 || a==0);

/\*цикл for работает пока >=1e-4\*/

for (n=1,f=x\*log(a)/n; fabs(f)>=1e-4; n++)

{

/\*расчет числителя текущего fn\*/

v1 \*= x\*log(a);

/\*расчет знаменателя текущего fn\*/

v2 \*=n;

/\*расчет текущего fn\*/

f = v1/v2;

/\*расчет суммы слагаемых\*/

s += f;

};

/\*выводим результат суммы слагаемых\*/

printf("s=%lf\n", s);

return 0;

}

*Результаты тестирования*

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| x = 0.5, a = 10 | 3.162278 | s=3.162267 |
| x = 0.2, a = 0.7 | 0.931150 | s=0.931149 |

1. *Вычислить значение суммы бесконечного ряд с точностью до члена ряда, меньшего ε=10-5 и значение функции (для проверки) , учесть,что x > 1.*

*Исходные данные:*

x – аргумент функции, тип double.

*Результирующие данные:*

s - значение суммы, , тип double.

*Вспомогательные переменные:*

n – индекс слагаемого, тип int, f – значение текущего слагаемого, тип double, v1 – значение числителя, тип double, v2 – значение знаменателя, тип double.

*Предварительные вычисления:*

n-ное слагаемое , можно рассчитывать отдельно числитель и знаменатель для каждого , знаменатель – , будем умножать v2 на n в каждом проходе, чтобы получить , текущего , получается, знаменатель = n\*v1, числитель - , будем умножать v1 на в каждом проходе, чтобы получить значение числителя текущего . Для вычисления итоговой суммы будем прибавлять значение текущего к s в цикле.

Схема программы:

*.*

начало

ввод x

конец

x

пока

x ≤ 1

s=0

v1=1

n=1

v2=1

вывод s

n=1

пока

f

v1 = v1\*(x-1)

v2 = v2 \* x

f = v1/(n\*v2)

s = s + f

n = n + 1

Текст программы

#include <stdio.h>

#include <math.h>

int main()

{

/\* объявление переменных \*/

double f, s=0., v1=1., x, v2=1.;

int n=1;

/\*цикл while с постусловием, контролирует ввод, чтобы x>1\*/

do

/\*ввод с клавиатуры вещественного чисела и запись в переменную:x\*/

scanf("%lf", &x);

while (x<=1.);

/\*цикл for работает пока >=1e-5 \*/

for (n=1,f=(x-1)/x; fabs(f)>=1e-5; n++)

{

/\*расчет числителя текущего fn\*/

v1 \*= x-1;

/\*расчет xn текущего fn\*/

v2 \*=x;

/\*расчет текущего fn\*/

f = v1/(n\*v2);

/\*расчет суммы слагаемых\*/

s += f;

};

/\*выводим результат суммы слагаемых\*/

printf("s=%lf\n", s);

return 0;

}

*Результаты тестирования*

|  |  |  |
| --- | --- | --- |
| **Исходные данные** | **Ожидаемый результат** | **Результат работы программы** |
| 2.5 | 0.916291 | s=0.916278 |
| 1.6 | 0.470004 | s=0.470001 |
| 15.1 | 2.714695 | s=2.714573 |

1. Дано натуральное число N. Вычислить . Функцию возведения в степень не использовать.

*Исходные данные:*

*n – целое, тип int.*

*Результирующие данные:*

*s – сумма слагаемых, тип double.*

*Вспомогательные переменные:*

f – значение знаменателя, тип double, i – счетчик первого цикла, цикла для прохода по дробям, тип int, j - счетчик второго цикла, цикла для возведения в степень.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *3* | 1.28703 | s=1.287037 |
| 4 | 1.29094328703 | s=1.290943 |
| 5 | 1.29126 | s=1.291263 |

Схема программы:

Текст программы

начало

ввод n

конец

Цикл 1

*i=2,n ш.1*

Цикл 2

s=1

вывод s

f = i

Цикл 2

j=1,i-1 ш.1

Цикл 1

f = f \* i

s = s + 1/f

#include <stdio.h>

int main()

{

/\* объявление переменных \*/

double f, s=1.;

int j,i,n;

/\*ввод с клавиатуры вещественного числа и запись в переменную:n\*/

scanf("%d",&n);

/\* 1 цикл for работает пока i<=n, проходит по слагаемым \*/

for (i=2; i<=n; i++)

{

f=i;

/\* 2 цикл for работает пока j<=i-1, возводит f в степень i\*/

for (j=1; j<=i-1; j++)

f\*=i;

/\*вычисление суммы слагаемых\*/

s+=1/f;

};

/\*вывод суммы\*/

printf("\ns=%lf", s);

return 0;

}