Лабораторная работа №3.  
Организация циклов в C.

Задание в аудитории:

#include <stdio.h>  
#include <math.h>  
  
**int** main(**int** argc, **char const** \*argv[]) {  
 **double** x;  
 x=1./20\*log(7./15);  
 printf("%f\n",x);  
 **return** 0;  
}

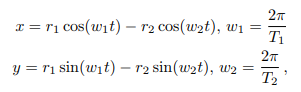
**Результат:**



Задание №1

**Постановка задачи:** Вычислить используя цикл for координаты планеты Марс относительно Земли с течением времени t. Распечатать на экране координаты для каждой итерации по t.

**Математическая модель:** Координаты планеты Марс для каждой итерации задаются заданы формулами:



где r1 – радиус орбиты Марса, r2 – радиус орбиты Земли, T1 и T2 – периоды обращения, указанных планет соответственно, t – каждый заданный момент времени внутри цикла по времени.

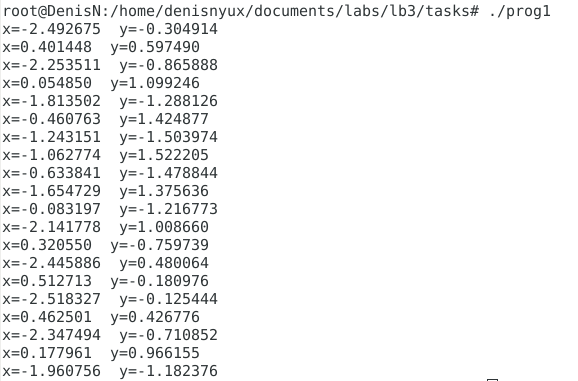
**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| r1 | радиус орбиты марса | float |
| r2 | радиус орбиты земли | float |
| t1 | период обращения марса вокруг солнца | float |
| t2 | период обращения земли вокруг солнца | float |
| x | координата по оси x | float |
| y | координата по оси y | float |
| w1, w2 | угловые скорость | float |

**Код программы:**

#include <stdio.h>  
#include <math.h>  
  
const float r1=1.5235; //радиус орбиты марса в а.е.  
const float r2=1.0; //радиус орбиты земли в а.е.  
const float t1=686./365; //период обращения марса вокруг солнца (год)  
const float t2=1.0; //период обращения земли вокруг солнца (год)  
  
int main(){  
 float w1=2\*M\_PI/t1,w2=2\*M\_PI/t2,x,y; //t (год)  
 for (float t=1;t<=20;t++){  
 x=r1\*cosf(w1\*t)-r2\*cos(w2\*t);  
 y=r1\*sinf(w1\*t)-r2\*sin(w2\*t);  
 printf("x=%f y=%f\n",x,y);  
 }  
 return 0;  
}

**Результат:**



Задание №2

**Постановка задачи:** Вычислить определённый интеграл от заданной функции:



**Математическая модель:** Формула вычисления интеграла методом левых частей:

, где

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| a,b | границы интегрирования | float |
| h | шаг | float |
| n | количество итераций | float |
| fx | значение интеграла | float |

**Код программы:**

#include <stdio.h>  
#include <math.h>  
  
int main(int argc, char const \*argv[]) {  
 float a,b,h,fx,n=100000;  
 printf("Enter a,b:");  
 scanf("%f %f",&a,&b);  
 h=(b-a)/n;  
 for (float x=a;x<=b-h;x=x+h){  
 fx=fx+expf(x+2);  
 }  
 printf("%f\n",fx );  
 return 0;  
}

**Результат:**

****

Задание №3

**Постановка задачи:** Организовать и распечатать последовательность чисел Фибоначчи, не превосходящих m, введенную с клавиатуры.

**Математическая модель:** Числа Фибоначчи - каждое число этой последовательности равно сумме двух предыдущих; например: 1, 1, 2, 3, 5, 8, 13,..

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| m | ограничение для чисел | int |
| x1 | 2ое предыдущее число в последовательности | int |
| x2 | 1ое предыдущее число в последовательности | int |
| t | временная переменная для того чтобы можно было сложить сумму 2 предыдущих | int |

**Код программы:**

#include <stdio.h>  
  
int main(int argc, char const \*argv[]) {  
 int x1=1,x2,t,m ;  
 printf("Enter m:" );  
 scanf("%d",&m );  
 printf("1 ");  
 for (x2=1;x2<m;) {  
 printf("%d ",x2);  
 t=x2;  
 x2=x2+x1;  
 x1=t;  
 }  
 printf("\n" );  
 return 0;  
}

**Результат:**

****

Задание №4

**Постановка задачи:** Дано натуральное число. Вывести на экран все натуральные числа до заданного включительно.

**Математическая модель:** Числа выводятся в цикле от 0 до заданного.

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| x | заданное число | int |
| i | счетчик в цикле перечисляющий все числа до заданного | int |

**Код программы:**

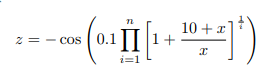
#include <stdio.h>  
  
int main(int argc, char const \*argv[]) {  
 int x;  
 printf("Enter a number: ");  
 scanf("%d",&x);  
 for (int i=0;i<=x;i++){  
 printf("%d ",i);  
 }  
 printf("\n" );  
 return 0;  
}

**Результат:**

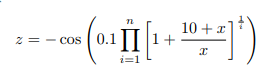
****

Задание №5

**Постановка задачи:** Вычислить значение выражения:



**Математическая модель:**



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| z | значение выражения | float |
| x | переменная, от которой зависит выражение | float |
| i | счетчик в цикле, который считает произведение | float |
| n | граница произведения | float |
| t | переменная, в которой сохраняется произведение | float |

**Код программы:**

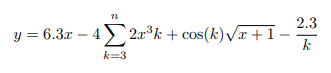
#include <stdio.h>  
#include <math.h>  
  
int main() {  
 float x,z,t=1,n;  
 printf("Enter x and n:" );  
 scanf("%f %f",&x,&n);  
 for (float i=1; i<n; i++){  
 t=t\*powf(1+(10+x)/x,1/i);  
 }  
 z=cos(0.1\*t);  
 printf("Z=%f\n",z);  
 return 0;  
}

**Результат:**

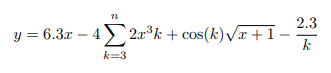


Задание №6

**Постановка задачи:** Вычислить значение выражения:



**Математическая модель:**



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| k | счетчик в цикле, который считает сумму | float |
| x | переменная в выражении | float |
| n | граница суммы | float |
| t | переменная, в которой накапливается сумма | float |

**Код программы:**

#include <stdio.h>  
#include <math.h>  
  
int main(int argc, char const \*argv[]) {  
 float x,n,y,t=0;  
 printf("Enter n,x:");  
 scanf("%f %f",&n,&x);  
 for (float k=3;k<=n;k++){  
 t=t+2\*x\*x\*x\*k+cosf(k)\*sqrtf(x+1)-2.3/k;  
 }  
 y=6.3\*x-4\*t;  
 printf("y=%f\n",y);  
 return 0;  
}

**Результат:**

****

Задание №7

**Постановка задачи:** С клавиатуры вводится трёхзначное число, считается сумма его цифр. Если сумма цифр числа больше 10, то вводится следующее трехзначное число, если сумма меньше либо равна 10 – программа завершается.

**Математическая модель:** В программе будет 2 цикла. Во внешнем число будет вводится до тех пор пока сумма его цифр не будет > 10. Во внутреннем сумма цифр будет считаться посредством суммирования остатка от целочисленного деления числа на 10.

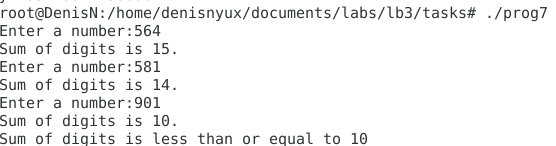
**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| x | число | int |
| sum | сумма цифр числа | int |

**Код программы:**

#include <stdio.h>  
#include <math.h>  
  
int main(int argc, char const \*argv[]) {  
 int x,sum=11;  
 while (sum>10){  
 printf("Enter a number:");  
 scanf("%d",&x );  
 sum=0;  
 while (x>0){  
 sum=sum+x%10;  
 x=x/10;  
 }  
 printf("Sum of digits is %d.\n" ,sum);  
 }  
 printf("Sum of digits is less than or equal to 10\n");  
 return 0;  
}

**Результат:**

****

Задание №8

**Постановка задачи:** Вычислить методом Ньютона:



с точностью abs=10e−6

**Математическая модель:** Метод ньютона заключается в том что мы строим касательные, постепенно приближаясь к точкам пересечения графика с осью абсцисс. Графики касательных строятся исходя из зависимости:

**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| fn(x) | функция по которой вычисляется каждый последующий x | float |
| acc | точность | float |
| x | предыдущее значение x | float |
| x1 | последующее значение x | float |

**Код программы:**

#include <stdio.h>  
#include <math.h>  
  
const float acc=10e-6;  
  
float fn(float x){  
 return x-(powf(x,4)-18\*x\*x+6)/(4\*powf(x,3)-36\*x);  
}  
  
int main(int argc, char const \*argv[]) {  
 float x,x1=10 ;  
 do {  
 x=x1;  
 x1=fn(x);  
 } while(fabs(x1-x)>acc);  
 printf("x=%f\n",x);  
 return 0;  
}

**Результат:**

****