**Условие задачи**

1.3.2 Составить циклический алгоритм и программу для вычисления результата по формуле (рис. 1). Для проверки программы задать X=0,5; n=20.

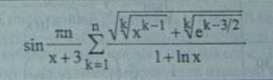


рис. 1

**Решении задачи**

Прежде всего необходимо учесть, чтобы наш алгоритм соответствовал нескольким важным факторам:

* Использование алгоритма было удобным;
* Алгоритм был простым и наглядным;
* Разработанная функция принимала все аргументы, включая эпсилон.

**Метод решения** В данной задачe необходимо вычислить произведение значения функции g(x) и суммы значений f(x) функции по задаваемому конечному значению n. Решение заключается в вычисление суммы значений некоторой функции fi(x) при целочисленном индексе i, меняющемся от начального значения n до конечного значения m и проведению получившейся суммы с синусом функции g(x);

**Состав данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| **Входные/Исходные данные** | | | |
| *n* | конечное значение для суммы | целочисленный тип данных | переменная |
| *x* | аргумент функции | вещественный тип данных | переменная |
| *e* | эпсилон | вещественный тип данных | переменная |
| *npi\** | число пи (не включено в текущую версию программы) | вещественный тип данных | переменная |
| **Выходные данные** | | | |
| *y* | результат функции | вещественный тип данных | переменная |
| **Промежуточные данные** | | | |
| *i,k* | переменная цикла, шаг суммы | целочисленный тип данных | переменная |
| *firstmultiplier* | результат первой функции | вещественный тип данных | переменная |
| *p* | число пи (берется из библиотек и модулей) | вещественный тип данных | переменная |
| *sum* | сумма функции | вещественный тип данных | переменная |

**Форма ввода**

x n

<*x> <n>*

e

<e>

**Форма вывода**

x y:

<x> <y>

**Листинг программы, написанной на языке object pascal и скомпилированной с помощью freepascal**

program lab\_1\_3\_2;

uses Math, sysutils;

function big\_brain\_formula(x, e, k: real): real;

begin

big\_brain\_formula := sqrt(Power(Power(x, (k - 1)), 1.0 / k) + Power(Power(e, (k - 3.0 / 2)), 1.0 / k)) / (1 + LogN(e, x))

end;

var

x, y, e, p, sum, firstmultiplier: real;

n, i, k: integer;

begin

k := 1;

p := Pi;

writeln('x n');

readln(x, n);

writeln('e');

readln(e);

firstmultiplier := sin((p \* n) / (x + 3));

for i := 0 to n do

begin

sum := sum + big\_brain\_formula(x, e, k);

k := k + 1;

end;

y := sum \* firstmultiplier;

writeln('x y');

writeln(x:2:2, ' ', y:2:2);

end.