**Условие задачи**

Составить программу вычисления N значений функции У (рис. 1) для Х, изменяющегося от |Х1, с шагом dХ. Для проверки правильности программы задать значения для А, Х1, и dХ из второй таблицы (рис. 2).

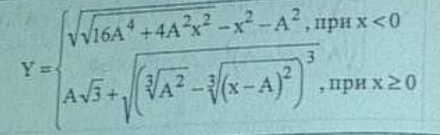


рис. 1

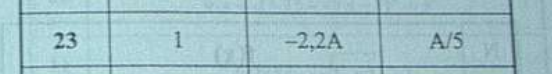


рис. 2

**Решении задачи**

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

* Использование алгоритма было удобным;
* Алгоритм был простым и наглядным;
* Алгоритм самого решения должен быть однопроходным;

**Метод решения** задачи состоит в нахождении элементов массива Y – значений функции при заданном множестве X, но по определенным правила изображенным на рисунке 1. Также для данной задачи необходимо задавать количество элементов массива и значения иниализирующие элементы этого массива.

**Состав данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя** | **Смысл** | **Тип** | **Структура** |
| **Входные/Исходные данные** | | | |
| *x* | входной параметры функции | вещественный тип данных | переменная |
| *dx* | приращение аргумента x | вещественный тип данных | переменная |
| *a* | входной параметр функции | вещественный тип данных | переменная |
| *n* | количество вычислений значений функции | целочисленный тип данных | переменная |
| **Выходные данные** | | | |
| *y* | результата функции | вещественный тип данных | переменная |
| **Промежуточные данные** | | | |
| *i* | переменная цикла | целочисленный тип данных | переменная |

**Форма ввода**

A X DX

<*a*> <x> <dx>

n  
<n>

**Форма вывода**

n x y

<i> <x> <y>

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

**program** lab\_1\_6;

**uses** Math;

**function** big\_brain\_formula\_2(x, a: real): real;

**begin**

big\_brain\_formula\_2 := a \* sqrt(3) + sqrt(Power((Power(sqr(a), 1.0 / 3) - Power(sqr(x - a), 1.0 / 3)), 3));

**end**;

**function** big\_brain\_formula\_1(x, a: real): real;

**begin**

big\_brain\_formula\_1 := sqrt(sqrt(16 \* Power(a, 4) + 4 \* sqr(a) \* sqr(x)) - sqr(x) - sqr(a));

**end**;

**var**

x, dx, a, y: real;

n, i: integer;

**begin**

writeln('A X DX');

readln(a, x, dx);

writeln('n');

readln(n);

writeln('n x y');

**for** i := 0 **to** n **do**

**begin**

**if** x < 0 **then**

**begin**

writeln(i, ' ', x:2:2, ' ', big\_brain\_formula\_1(x, a):2:2);

**end**

**else begin**

writeln(i, ' ', x:2:2, ' ', big\_brain\_formula\_2(x, a):2:2);

**end**;

x := x + dx;

**end**;

**end**.