**УО «Белорусский государственный университет информатики и**

**радиоэлектроники»**

**Кафедра ПОИТ**

**Отчет по лабораторной работе №1.3**

**по предмету**

**Основы Алгоритмизации и Программирования**

**Вариант 1**

**Выполнил**

**Андросов И.С.**

**Проверила**

**Данилова Г.В.**

Группа:

**8**51001

**Минск 2018**

**Задание**

Вычислите значение квадратного корня y = с точностью EPS с использованием итерационной формулы Ньютона:

Y0 = 1

Yi = 1/2 (Yi-1 + X/Yi-1) (i = 1, 2, 3...)

Вычисления производить пока |Yi - Yi-1| не станет меньше EPS. Подсчитайте количество итераций, за которое достигается эта точность.

**Код программы**

**(Delphi)**

**program** laba\_1\_3;

{$APPTYPE CONSOLE}

**uses**

SysUtils;

**var**

Count, X: Integer;

Y: Real;

IsCorrect: Boolean;

**begin**

WriteLn('This program calculates the square root of X. Enter X [0;2147483647]:');

IsCorrect := false;

**repeat**

**try**

ReadLn(X);

**if** (X > -1) **and** (X < 2147483648) **then**

IsCorrect := true

**else**

WriteLn('Enter number from interval [0;2147483647]:');

**except**

WriteLn('Check entered data (enter number from interval [0;2147483647]):');

**end**;

**until** IsCorrect;

Y := 1;

Count := 0;

**while** abs((X + sqr(Y)) / (2 \* Y) - Y) > 0.00001 **do**

**begin**

Y := (X + sqr(Y)) / (2 \* Y);

Inc(Count);

**end**;

WriteLn(Y:0:5, #10+ 'Number of iterations: ',Count);

ReadLn;

**end**.

**Код программы**

**(С)**

#include <stdio.h>

int main()

{

    int Count = 0, X;

    float Y = 1;

    printf("This program calculates the square root of X. Enter X [1;2147483647]:**\n**");

    scanf("%d", &X);

    if ( X < 0 || X > 2147483647)

        printf("Enter number from interval [1;2147483647].**\n**");

    else

        while (fabs((X + (Y \* Y)) / (2 \* Y) - Y) > 0.00001)

        {

            Y = (X + (Y \* Y)) / (2 \* Y);

            Count++;

        }

    printf("%0.5f **\n**Number of iterations: %d.",Y , Count);

    return 0;

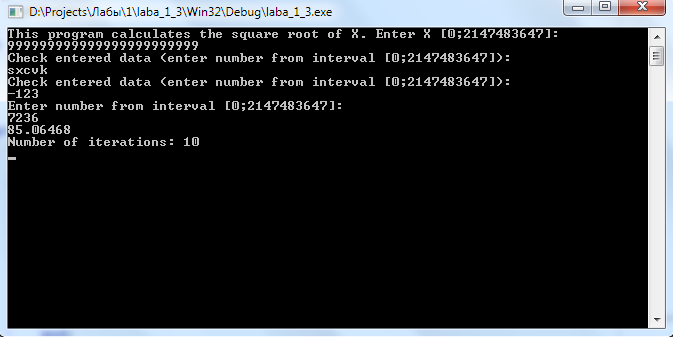
}

**Схема алгоритма**



**Скриншоты**

**(Delphi)**

****

**(С)**

