**Задание:**

Разработать программу вычисления корня квадратного по итерационной формуле Герона Александрийского с точностью не хуже 0,05% (использовать FPU).

**Решение.**

Итерационная формула Герона имеет вид

,

где a — фиксированное положительное число, а x1 — любое положительное число.

Итерационная формула задаёт убывающую (начиная со 2-го элемента) последовательность, которая при любом выборе x1 быстро сходится к величине (квадратный корень из числа), то есть

Эту формулу можно получить, применяя метод Ньютона к решению уравнения a – x2=0.

Пример

Попробуем вычислить квадратный корень для 25, используя округления при вычислениях. Пусть нашим первым предположением для значения будет значение 3.

|  |  |  |  |
| --- | --- | --- | --- |
| n | xn |  | Приблизительное значение xn+1 |
| 1 | 3 |  |  |
| 2 | 5.67 |  |  |
| 3 | 5.04 |  |  |
| 4 | 5 |  |  |

Ограничением для данной формулы является x>0, a>0.

Напишем сначала программу, вычисляющую корень квадратный по итерационной формуле Герона Александрийского на языке С++, чтобы отработать алгоритм.

Текст программы приведен ниже.

#include <iostream>

using namespace std;

int main()

{

double a = 2.5;

double xp;

double e = 0.00005;

double x=a;

do

{

xp = x;

x = (x + a / x) / 2;

} while (fabs(x - xp)>=e);

cout << x << endl;

system("pause");

return 0;

}

Итерационный цикл завершается когда очередной член суммы по модулю будет меньше заданной точности. Точность 0,05% соответствует абсолютной величине 0,0005.

Теперь, когда алгоритм отлажен, и результат совпадает с точным значением (в пределах заданной точности), перепишем данную программу на ассемблере для компилятора FASM.

Для ввода исходных данных воспользуемся функциями стандартной библиотеки си:

char \*gets(char \*str)

Функция gets() считывает символы из stdin и помещает их в массив символов, на который ука­зывает str. Символы считываются до тех, пока не встретится новая строка или EOF. Символ «но­вая строка» не делается частью строки, а транслируется в нулевой символ, завершающий строку.

Чтобы преобразовать строку в действительное значение, применим функцию sscanf, которая распознает и считывает​​ данные​​ по​​ заданному​​ шаблону из строки.​​

int sscanf(char \*buf, const char \*format, arg-list);

Она​​ имеет​​ следующие​​ аргументы:

● buf ​​ —​​ указатель​​ на​​ символьный​​ буфер​​, подлежащий считыванию;

● format — указатель на C-строку, содержащую формат результата;

● остальные​​ аргументы​​ —​​ данные,​​ подлежащие​​ форматированию.

Данная функция возвращает целое значение, которое представляет собой количество корректно распознанных значений из format, что позволяет реализовать проверку корректности ввода.

Так как у данной функции количество аргументов не постоянно, то она использует соглашение вызова cdecl, которое отличается от stdcall тем, что очистку стека от аргументов выполняет вызывающая функция, а не  
вызываемая.

Для вывода результата воспользуемся функцией printf, которая форматирует​​ данные​​ по​​ заданному​​ шаблону и выводит их на консоль.

int printf (const char \* format, ... );

​​ Она​​ имеет​​ следующие​​ аргументы:

● format — указатель на C-строку, содержащую формат результата;  
● остальные​​ аргументы​​ —​​ данные,​​ подлежащие​​ форматированию.

Так как у данной функции количество аргументов не постоянно, то она использует соглашение вызова cdecl, которое отличается от stdcall тем, что очистку стека от аргументов выполняет вызывающая функция, а не  
вызываемая.

Для завершения работы программы выполним вызов функции ExitProcess,  
которая​​ имеет​​ один​​ аргумент​​ —​​ код​​ завершения​​ работы​​ программы.

VOID ExitProcess(

UINT uExitCode // код выхода для всех потоков

);

Входной параметр x считывается из консоли (клавиатуры). Результат работы выводится на консоль (экран). Программу можно разбить на следующие функции:

Главная функция программы. В ней выполняет ввод исходных данных с проверкой их корректности, вызывается функция mysqrt и выводятся на консоль результат ее выполнения и значение квадратного корня введенного числа, вычисленное встроенной функцией FPU.

**double mysqrt(double x,double eps);**

mysqrt – функция вычисления корня квадратного по итерационной формуле Герона Александрийского с заданной точностью eps. Вызывается по соглашению cdecl. Имеет локальную переменную xp - значение х на предыдущей итерации.

Результат возвращается в ST(0).

Текст программы приведен ниже:

format PE console

entry start

include 'include\win32ax.inc'

;Секция данных содержит строковые переменные и константы, доступна только для чтения

section '.data' data readable writeable

x1 dq ? ;введенное значение

eps1 dd 0.0005 ;точность 0.05%

;константы

c2 dd 2

msg1 db 'Enter x (x>0): ',0

msg2 db 'Wrong number.',13,10,0

fmt1 db '%lf',0

msg3 db "Geron's formula = %lg",13,10,0

msg4 db 'Sqrt = %lg',13,10,0

buf db 256 dup(0)

;секция кода

section '.code' code readable executable

start: ; точка входа в программу

ccall [printf],msg1 ;вывести сообщение

ccall [gets],buf ;ввод строки с консоли

ccall [sscanf],buf,fmt1,x1 ;преобразовываем введенную строку в число

cmp eax,1 ;если преобразование удалось,

jz m1 ;продолжить

ccall [printf],msg2 ;вывести сообщение об ошибке

jmp start ;начать заново

m1: fld [x1] ;x

fldz ;0

fcompp ;сравнить 0 с веденным числом

fstsw ax ;записать флаги сопроцессора в ах

sahf ;перенести их в флаги процессора

jae start ;0>=x, начать заново

fld [eps1] ;точность вычисления

sub esp,8 ;выделить в стеке место под double

fstp qword [esp] ;записать в стек double число

fld qword [x1] ;введенное значение

sub esp,8 ;выделить в стеке место под double

fstp qword [esp] ;записать в стек double число

call mysqrt ;Вычислить mysqrt(x,eps)

add esp,16 ;удалить переданные параметры

sub esp,8 ;передать вычисленное по формуле Герона значение

fstp qword [esp] ;функции через стек

push msg3 ;формат сообщения

call [printf] ;сформировать результат

add esp,12 ;коррекция стека

fld qword [x1] ;введенное значение

fsqrt ;вычислить квадратный корень из него

sub esp,8 ;передать точное значение квадратного корня

fstp qword [esp] ;функции через стек

push msg4 ;формат сообщения

call [printf] ;сформировать результат

add esp,12 ;коррекция стека

ccall [\_getch] ;ожидание нажатия любой клавиши

ex: stdcall [ExitProcess], 0;выход

;double mysqrt(double a,double eps)

;вычисление sqrt(a) с точностью eps

;соглашение вызова cdecl

mysqrt:

push ebp ;создать кадр стека

mov ebp,esp

sub esp,08h ;создание локальных переменных

;локальные переменные

xp equ ebp-8h ;значение х на предыдущем шаге

fld qword [ebp+8] ;x=a

lp: fst qword [xp] ;xp=x

fld qword [ebp+8] ;a

fld st1 ;x

fdivp st1,st ;a/x

faddp st1,st ;x+a/x

fidiv [c2] ;x=(x+a/x)/2

fld st ;x

fsub qword [xp] ;x-xp

fabs ;|x-xp|

fcomp qword [ebp+16] ;сравнить |x-xp| с eps

fstsw ax ;перенести флаги сравнения в ах

sahf ;занести ah в флаги процессора

jae lp ;если |x-xp| >= xp, продолжаем цикл

leave ;эпилог функции

ret

section '.idata' import data readable ; секция импортируемых функций

library kernel,'kernel32.dll',\

user,'user32.dll',\

msvcrt,'msvcrt.dll'

import kernel,\

ExitProcess,'ExitProcess'

import msvcrt,\

sscanf,'sscanf',\

gets,'gets',\

\_getch,'\_getch',\

printf,'printf'

Результат выполнения программы приведен на рисунке 1.

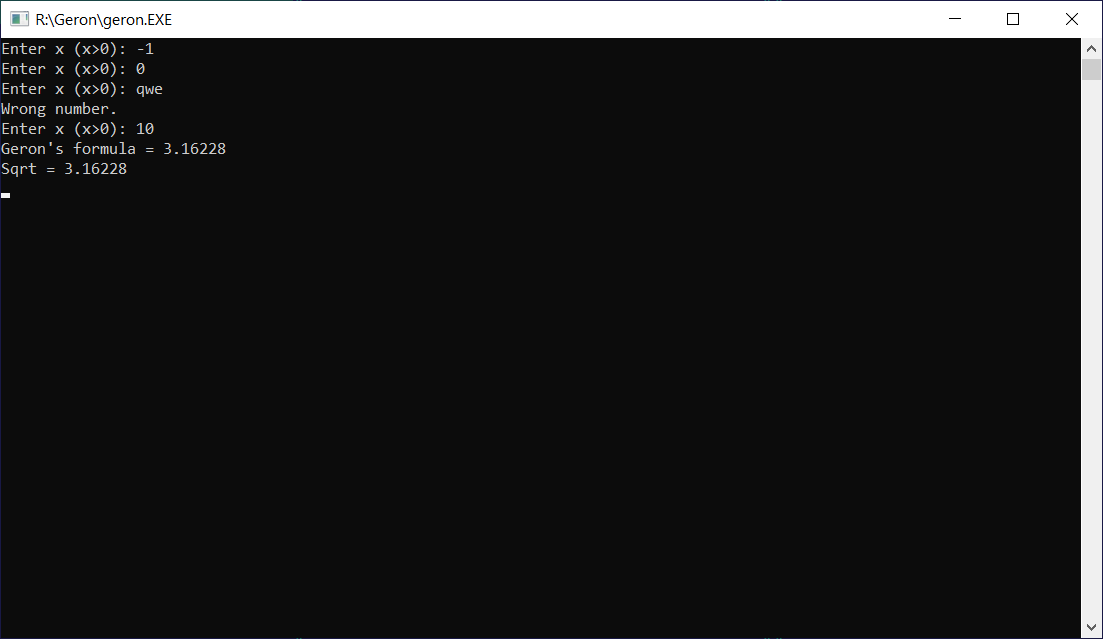


Рисунок 1 ­– Результат выполнения программы

Как видно из рисунка 1, программа обрабатывает некорректный ввод, погрешность вычисления соответствует заданной.

**Список использованных источников**

1 <https://ru.wikipedia.org/wiki/Итерационная_формула_Герона>

2 Юров В., Хорошенко С. Assembler: учебный курс – СПб: Питер Ком, 1999. – 672с.: ил.

3 Кип Р.И. Язык ассемблера для процессоров Intel. М.: Издательский дом “Вильямс”, 2005. – 912с.