Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ МОРСКОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(СПБГМТУ)

Факультет Морского Приборостроения

Кафедра систем автоматического управления и бортовой вычислительной техники

**Пояснительная записка**

К курсовой работе

«Программа табулирования функции»

Вариант 16

По дисциплине «Языки ассемблера»

Специальность: Вычислительные машины, комплексы, системы и сети

Выполнил:

Студент 2 курса группы № 3270

Сергиенко Д.К.

Подпись:

Проверил:

Доцент

Сакович С. Ю.

Подпись:

Дата выполнения отчёта:

Дата сдачи отчёта:

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc106727584)

[1 Анализ решаемой задачи 4](#_Toc106727585)

[ЗАКЛЮЧЕНИЕ 9](#_Toc106727586)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc106727587)

[ПРИЛОЖЕНИЕ А 11](#_Toc106727588)

# ВВЕДЕНИЕ

Задача

Составить программу табулирования заданной функции в заданном интервале с заданным шагом:

Используя парадигму структурного программирования в терминах языка ассемблера FASM с функцией записи в файл.

# Анализ решаемой задачи

Анализ функции

Функция 16 варианта:

Область Допустимых Значений функции:

ОДЗ функции обусловлена присутствием аргумента функции *x*:

* под знаком корня
* под знаком корня в знаменателе дроби в функции натурального логарифма

Поскольку подкоренное выражение не может быть отрицательно, то значение знаменателя подкоренного выражения строго положительно (). Значение знаменателя подкоренного выражения при , следовательно :

В совокупности (2) и (3):

Область Определения Функции:

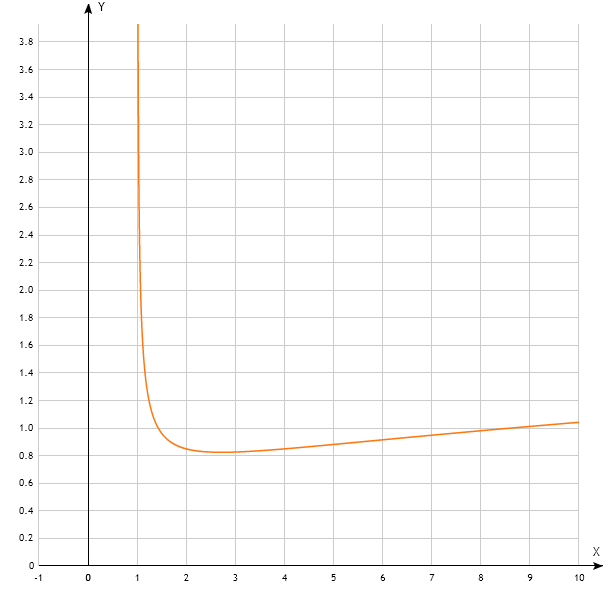
Стационарные точки:

*– точка минимума.*

Минимальное значение функция принимает в:

Следовательно:

График функции:



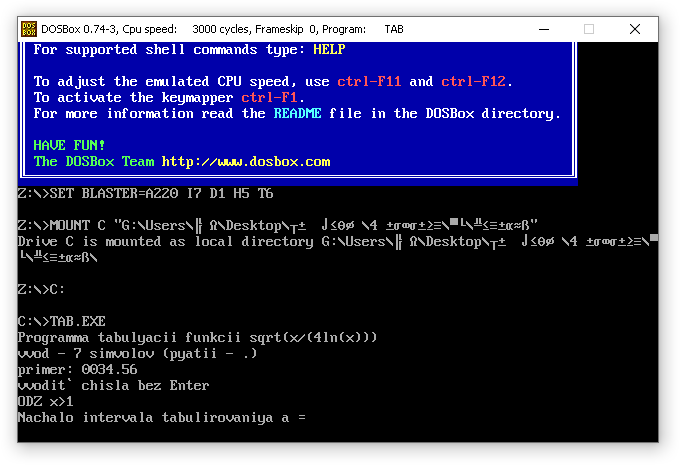
*Рисунок 1 – график функции 16 варианта*

Детали реализации

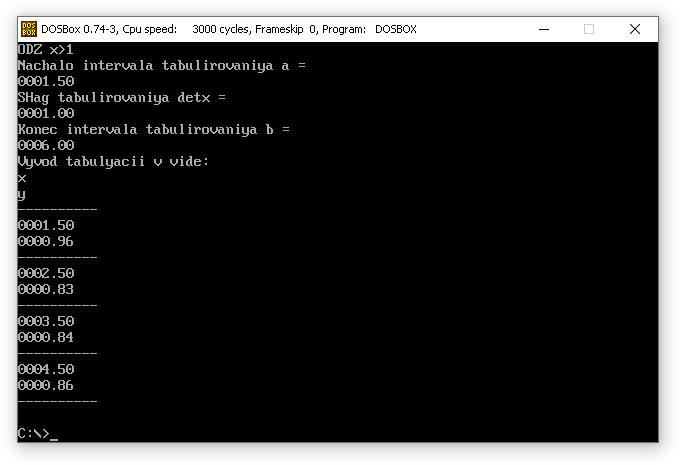
Сообщения

В поставленной задаче существуют вводные данные при которых вывода быть не может. Следовательно программа должна уведомить об этом пользователя.

# Пример работы

**

*Рисунок 2 – Скриншот консоли программы*



*Рисунок 3 – Скриншот работы программы*

# ЗАКЛЮЧЕНИЕ

В курсовой работе были рассмотрены табулирование и исследование функции.

В ходе выполнения были решены все поставленные задачи.

Проведено математическое исследование функции, в ходе которого была найдена область допустимых значений функции и произведены исключения в вводных данных, что позволило исключить ошибки при табулировании и правильно вывести график.

Используя методы структурного программирования, написана программа для табулирования функции. Составлена блок-схема, которая схематично описывает алгоритм решения задачи.

В процессе выполнения курсовой работы усовершенствованы навыки программирования на языке ассемблера FASM 1.73.30.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рудольф, М. Ассемблер на примерах. Базовый курс. / М. Рудольф. — Санкт-Петербург: Наука и Техника, 2005. — 240 c. — Текст: непосредственный.
2. ГОСТ 7.32-2017 СИБИД. — Текст : электронный // Техэксперт : [сайт]. — URL: https://docs.cntd.ru/document/1200157208 (дата обращения: 06.12.2021).
3. Генератор библиографического описания. — Текст : электронный // Библиотеки Тольятти : [сайт]. — URL: https://cls.tgl.ru/generator-bo/ (дата обращения: 16.12.2021).

# ПРИЛОЖЕНИЕ А

**Tabulation.asm**

format MZ

org 100h

include 'Macrosbi.inc'

main:

finit

fldz

fst [a]

fst [b]

fst [detx]

fstp [y]

prent GlavUved

prent aravn

chitaem a

prent detxravn

chitaem detx

prent bravn

chitaem b

finit

fld1

fld [a]

fcomip st1

jnb odz1

prent aerrg

jmp main

odz1:

finit

mov [buff],0

fstcw [buff]

and [buff], 1111001111111111b

or [buff], 1111011111111111b

fldcw [buff]

fld [b]

fsub [a]

fldz

fxch

fcomi st1

faddp

jnb odz2

prent ierrg

jmp main

odz2:

fdiv [detx]

FRNDINT

fist dword [buff]

mov ecx,dword [buff]

prent subtabUved

prent palochki

tab:

prentx a

finit

fild [four]

fldln2

fld [a]

fyl2x

fmulp

fld [a]

fdivrp

fsqrt

fstp [y]

prentx y

jmp aoao

tab1: jmp tab

aoao:

prent palochki

finit

fld [a]

fld [detx]

fadd st0,st1

fst [a]

fld [b]

loop tab1

endinghell:

getouthere

four dq 4

a rq 1

b rq 1

detx rq 1

y rq 1

buff rw 1

dodot rw 1

aftdot rw 1

GlavUved: db "Programma tabulyacii funkcii sqrt(x/(4ln(x)))",0Ah,"vvod - 7 simvolov (pyatii - .)",0Ah,"primer: 0034.56",0Ah,"vvodit` chisla bez Enter",0Ah,"ODZ x>1",0Ah,"$"

subtabUved: db "Vyvod tabulyacii v vide:",0Ah,"x",0Ah,"y",0Ah,"$"

palochki: db 10 dup "-",0Ah,"$"

aerrg: db 0Ah,"a ne udovletvoryaet ODZ, vvedite vse zanovo",0Ah,0Ah,"$"

ierrg: db 0Ah,"interval otricatelnii ili = 0, vvedite vse zanovo",0Ah,0Ah,"$"

aravn: db "Nachalo intervala tabulirovaniya a = ",0Ah,"$"

detxravn: db "SHag tabulirovaniya detx = ",0Ah,"$"

bravn: db "Konec intervala tabulirovaniya b = ",0Ah,"$"

**Macrosbi.inc**

macro getouthere

{

mov eax,4C00h

int 21h

}

macro prent streng

{

mov edx,streng

mov eax,0900h

int 21h

}

macro prentOx polovina,del

{

xor eax,eax

xor edx,edx

xor ebx,ebx

mov ax,[polovina]

mov bx,del

div bx

mov [polovina],dx

xor edx,edx

add al,'0'

mov dl,al

mov ah,06h

int 21h

}

macro prentx valux

{

local jgood,negood,good2,jgood2,negood2,neggood,good

finit

fld1

fld [valux]

fprem

mov [buff],100

fild [buff]

fmul st0,st1

frndint

fist [aftdot]

cmp [aftdot],0

jge jgood2

jmp negood2

jgood2:

jmp good2

negood2:

neg [aftdot]

good2:

finit

fld [valux]

mov [buff],0

fstcw [buff]

and [buff], 1111001111111111b

or [buff], 1111011111111111b

fldcw [buff]

frndint

fist [dodot]

cmp [dodot],0

jge jgood

jmp negood

jgood:

jmp good

negood:

xor edx,edx

add dx,'-'

mov ah,06h

int 21h

cmp [aftdot],0

je neggood

add [dodot],1

neggood:

neg [dodot]

good:

prentOx dodot, 1000

prentOx dodot, 100

prentOx dodot, 10

prentOx dodot, 1

xor edx,edx

add dx,'.'

mov ah,06h

int 21h

prentOx aftdot, 10

prentOx aftdot, 1

xor edx,edx

mov dl,0Ah

mov ah,06h

int 21h

}

macro chitaemOx

{

xor eax,eax

mov ah,01h

int 21h

}

macro domnoghaemOx mod,Oxe

{

xor ah,ah

sub ax,'0'

mov [buff],ax

finit

fild [buff]

mov [buff],mod

fild [buff]

fmul st0,st1

fst [Oxe]

}

macro delimOx mod,Oxe

{

xor ah,ah

sub ax,'0'

mov [buff],ax

finit

fild [buff]

mov [buff],mod

fild [buff]

fdiv st1,st0

fld [Oxe]

fadd st0,st2

fst [Oxe]

}

macro chitaem valuxe

{

chitaemOx

domnoghaemOx 1000,valuxe

chitaemOx

domnoghaemOx 100,valuxe

chitaemOx

domnoghaemOx 10,valuxe

chitaemOx

domnoghaemOx 1,valuxe

chitaemOx

chitaemOx

delimOx 10,valuxe

chitaemOx

delimOx 100,valuxe

xor edx,edx

mov dl,0Ah

mov ah,06h

int 21h

}