Федеральное государственное автономное

образовательное учреждение

Высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Программная инженерия |
| кафедра |

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ**

|  |
| --- |
| Метки, логические переходы, циклы |
| тема |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  | М. В. Сарамуд |
|  | |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ21-17/1Б, 032156940 |  |  |  | Н. А. Самарин |
|  | номер группы, зачётной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2024

**СОДЕРЖАНИЕ**

1 Задание............................................................................................................... 3

2 Вариант.............................................................................................................. 3

3 Ход работы........................................................................................................ 3

4 Вывод................................................................................................................. 6

**1 Задание**

Разработать программу на Ассемблере, реализующую вычисление Y для  
заданных пользователем X и A. Выполните упражнение из ниже приведенного  
списка, выбирая вариант соответственно номеру студента в группе. А остается  
заданной пользователем, Х меняется в цикле как Х+i, i меняется от 0 до 9 с  
шагом 1. Программа должна работать с числами с плавающей точкой,  
рекомендуется использовать функции FPU.

**2 Вариант**

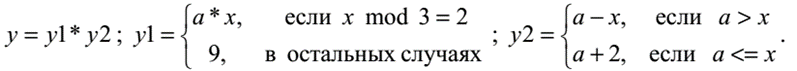


Рисунок 1 – Вариант 24

**3 Ход работы**

В работе был использован FASM на операционной системе Windows 10.  
Был использован набор инструкций FPU. Ниже представлен код разработанных  
программ и скриншоты их работы.

Листинг 1 – Код программы

; Самарин Никита КИ21-17/1 Вариант 24  
format PE console  
entry start  
include 'win32a.inc'  
section '.data' data readable writeable  
 resStr db 'y = %0.2f', 13, 10, 0  
 enterXStr db 'enter X: ', 0  
 enterAStr db 'enter A: ', 0  
 spaceStr db ' %f', 0   
 y1Str db 'y1 = %0.2f', 13, 10, 0  
 y2Str db 'y2 = %0.2f', 13, 10, 0  
 counterStr db 'counter: %0.2f', 13, 10, 0  
 xStr db 'x: %0.2f', 13, 10, 0  
 aStr db 'a: %0.2f', 13, 10, 0  
 X dd 0.0  
 A dd 0.0   
 y dd 0.0  
 y1 dd 0.0  
 y2 dd 0.0  
 two dd 2.0  
 three dd 3.0  
 nine dd 9.0  
 counter dd 0.0   
 NULL = 0  
section '.code' code readable executable  
start:  
 push dword enterAStr  
 call [printf]  
 add esp, 4  
 ; Ввод числа A   
 push A  
 push dword spaceStr

Продолжение листинга 1

call [scanf]  
 add esp, 8  
 push dword enterXStr  
 call [printf]  
 add esp, 4  
 ; Ввод числа X   
 push X  
 push dword spaceStr  
 call [scanf]  
 add esp, 8  
 mov ecx, 0  
iter\_loop:  
 finit  
 push ecx  
 fld dword [three]  
 fld dword [X]  
 fprem  
 fld dword [two]  
 fcomip st0, st1  
 jne else1  
 fld dword [A]  
 fmul dword [X]  
 fstp dword [y1]  
 jmp end\_if1  
else1:   
 fld dword [nine]  
 fstp dword [y1]  
end\_if1:   
 fld dword [X]  
 fld dword [A]  
 fcomip st0, st1  
 jbe else2  
 fld dword [A]  
 fsub dword [X]  
 fstp dword [y2]  
 jmp after\_calculation  
else2:  
 fld dword [two]  
 fadd dword [A]  
 fstp dword [y2]  
after\_calculation:  
 ;call [getch]  
 fld dword [y1] ; Загрузить результат в стек FPU  
 sub esp, 8 ; Сделать место для временного хранения значения на ст  
 еке  
 fstp qword [esp] ; Сохранить результат во временном месте на стеке  
 push y1Str ; Загрузить формат вывода в стек  
 call [printf] ; Вызвать функцию printf  
 add esp, 12 ; Очистить стек  
 fld dword [y2] ; Загрузить результат в стек FPU  
 sub esp, 8 ; Сделать место для временного хранения значения на ст  
 еке  
 fstp qword [esp] ; Сохранить результат во временном месте на стеке  
 push y2Str ; Загрузить формат вывода в стек  
 call [printf] ; Вызвать функцию printf  
 add esp, 12 ; Очистить стек  
 fld dword [y1]  
 fmul dword [y2]

Окончание листинга 1

fstp dword [y]  
 fld dword [y] ; Загрузить результат в стек FPU  
 sub esp, 8 ; Сделать место для временного хранения значения на ст  
 еке  
 fstp qword [esp] ; Сохранить результат во временном месте на стеке  
 push resStr ; Загрузить формат вывода в стек  
 call [printf] ; Вызвать функцию printf  
 add esp, 12 ; Очистить стек  
 fld1  
 fadd dword [X]  
 fstp dword [X]  
 pop ecx  
 inc ecx  
 cmp ecx, 10  
 jl iter\_loop  
 jmp finish   
 finish:  
 call [getch]  
 push NULL  
 call [ExitProcess]  
section '.idata' import data readable  
 library kernel, 'kernel32.dll', \  
 msvcrt, 'msvcrt.dll'  
 import kernel, \  
 ExitProcess, 'ExitProcess'  
 import msvcrt, \  
 printf, 'printf', \  
 getch, '\_getch', \  
 scanf, 'scanf'

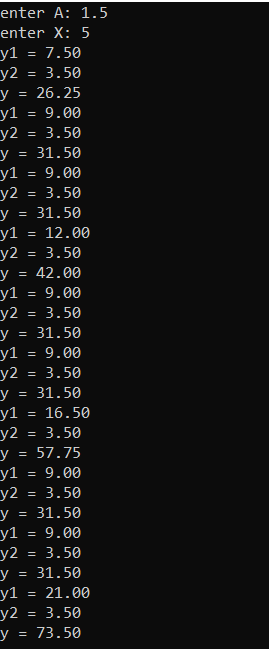


Рисунок 2 – Работа программы

**4 Вывод**

После изучения необходимой теории задание было выполнено.