МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №3

«ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКОГО СОПРОЦЕССОРА»

Вариант 11

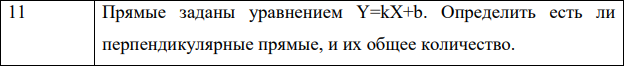
|  |  |
| --- | --- |
| Выполнил: студент группы ИНБс–3301 | Д.А. Загарских |
|  |  |
| Проверил: доцент кафедры РЭС | М.А. Земцов |

Киров

2024

**Цель работы:** изучение принципов выполнения арифметических команд с помощью математического сопроцессора FPU микропроцессоров с архитектурой x86.

**Текст задания:**



**Ход работы:**

Даны 10 значений K у 10 прямых: 1.0, 1.0, 2.0, 1.0, -5.0, 5.0, 8.0, -2.0, 3.0, 9.0.  
Сравниваем числа из массива со следующим числом из массива.



Если произведение чисел равно 1, то увеличиваем внутренний счётчик на 1





Если произведение не равно 1, то мы проверяем следующее число  


В конце программа выводит общее количество перпендикулярных прямых  
  
**Текст программы:**

.686

.model flat, stdcall

.stack 100h

.data

array dd 1.0, 1.0, 2.0, 1.0, -5.0, 5.0, 8.0, -2.0, 3.0, 9.0;

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eAx,eAx

xor eDx,eDx

xor eCx,eCx

xor eBx,eBx; Очищаем усё

call fpu\_use

jmp exit

fpu\_use proc

loop\_start:

xor eAx,eAx

mov ebx, ecx ; Загружаем текущий индекс массива в eax

cmp ebx, 10 ; Проверяем, достигли ли мы последнего элемента массива

je end\_loop ; Если да, завершаем программу

fld [array + eCx \* 4] ; Загружаем число с плавающей запятой из массива в стек

fmul [array + eBx \* 4] ; Умножаем на следующее число в массиве

fld1 ; Загружаем число 1 для сравнения

fcompp ; Сравниваем умноженное число с 1

fstsw ax

sahf

je greater ; Если равно 1, переходим к метке greater

jb less ; Если неравно 1, переходим к метке less

jmp next\_iteration ; Переходим к следующей итерации

greater:

; Обработка, если умноженное число равно 1

add edx, 1;

jmp next\_iteration

less:

; Обработка, если умноженное не равно 1

jmp next\_iteration

next\_iteration:

inc ecx ; Увеличиваем счетчик элементов массива

jmp loop\_start ; Переходим к следующей итерации

end\_loop:

fpu\_use endp

exit:

Invoke ExitProcess, edx;

End Start

[Репозиторий](https://github.com/HeNtiFox/CUiM/tree/main/lab3)

**Вывод**: были изучены принципы выполнения арифметических команд с помощью математического сопроцессора FPU микропроцессоров с архитектурой x86.