НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Архитектура вычислительных систем»

Домашнее задание

Выполнил: Казанцева Анастасия,

студент гр. БПИ162.

Преподаватель: Баканов В.М.,

профессор департамента

программной инженерии

факультета компьютерных наук

Москва 2017

**ТЕКСТ ЗАДАНИЯ**

Разработать программу которая по параметрам трех отрезков (задаются в исходном коде как декартовы координаты исходных отрезков в виде целых без знака) решает, могут ли они являться сторонами равностороннего треугольника.

**РАСЧЕТНЫЕ МЕТОДЫ**

Для произведения расчетов использована стандартная формула подсчета длины отрезка

Формула длины отрезка.

Но т.к. для ответа на поставленную задачу не требуется конкретно нахождение длины – сравниваются не длины отрезков, а квадраты длин.

**ТЕКСТ ПРОГРАММЫ**

;СТУДЕНТКА БПИ162 КАЗАНЦЕВА АНАСТАСИЯ РОМАНОВНА

;ЗАДАЧА: РАЗРАБОТАТЬ ПРОГРАММУ КОТОРАЯ ПО ПАРАМЕТРАМ ТРЕХ ОТРЕЗКОВ

; (ЗАДАЮТСЯ В ИСХОДНОМ КОДЕ КАК ДЕКАРТОВЫ КООРДИНАТЫ ИСХОДНЫХ

; ОТРЕЗКОВ В ВИДЕ ЦЕЛЫХ БЕЗ ЗНАКА) РЕШАЕТ, МОГУТ ЛИ ОНИ ЯВЛЯТЬСЯ

; СТОРОНАМИ РАВНОСТОРОННЕГО ТРЕУГОЛЬНИКА.

;---------------------------------------------

format PE GUI 4.0

entry start

include 'c:\fasmw\INCLUDE\WIN32AX.INC'

include 'C:\fasmw\INCLUDE\MACRO\PROC32.INC'

;---------------------------------------------

section '.data' data readable writeable

title db 'Это равносторонний треугольник?',0

; начальные x-координаты 1,2 и 3 отрезков соответственно

bx1 dd 0

bx2 dd 0

bx3 dd 0

; начальные y-координаты 1,2 и 3 отрезков соответственно

by1 dd 3

by2 dd 1

by3 dd 1

; конечные x-координаты 1,2 и 3 отрезков соответственно

ex1 dd 2

ex2 dd 2

ex3 dd 2

; конечные y-координаты 1,2 и 3 отрезков соответственно

ey1 dd 3

ey2 dd 1

ey3 dd 1

;---------------------------------------------

section '.code' code readable executable

start:

; считаем длину первого отвезка

stdcall length, [bx1], [by1], [ex1], [ey1]

; запоминаем вернувшееся значение в ax

mov ax, cx

; считаем длину второго отвезка

stdcall length, [bx2], [by2], [ex2], [ey2]

; проверяем равны ли длины двух первых отрезков

cmp cx, ax

jne .fal

; считаем длину третьего отвезка

stdcall length, [bx3], [by3], [ex3], [ey3]

; проверяем равны ли длины первого и второго отрезков

cmp cx, ax

jne .fal

invoke MessageBox,NULL,"ДА:)",title,MB\_OK

jmp .exit

.fal:

invoke MessageBox,NULL,"НЕТ:(",title,MB\_OK

jmp .exit

.exit:

invoke ExitProcess,0

;-----------------------------------------------------

; принемает координаты начала и координаты конца, возвращает длину отрезка через регистр cx

proc length uses ax, x1:WORD, y1:WORD, x2:WORD, y2:WORD

; считаем все по формуле (x2-x1)^2 + (y2-y1)^2

; (x2-x1)^2

mov ax,[x2]

;(x2-x1)

sub ax,[x1]

; возводим в квадрат, резулитат в ax

imul ax,ax

; пересохраним первое слагаемое в cx

mov cx,ax

; (y2-y1)^2

mov ax,[y2]

; (y2-y1)

sub ax,[y1]

; возводим в квадрат, резулитат в ax

imul ax, ax

; добавляем второе слагаемое к результату

add cx,ax

ret

endp

;---------------------------------------------

section '.idata' import data readable writeable

library kernel32,'kernel32.dll',\

user32,'user32.dll'

include 'C:\fasmw\INCLUDE\API\KERNEL32.INC'

include 'C:\fasmw\INCLUDE\API\USER32.INC'

**ПРИМЕНЯЕМЫЕ МАКРОСЫ И ПРОЦЕДУРЫ**

Макросы PROC и ENDP:

Позволяют создать вызываемый блок кода, т.е. процедуру. Они сами организовывают укладывание параметров в стек, а также благодаря ключевому слову uses дают возможность восстанавливать регистры, использованные внутри нашего блока кода. Невосстановленные регистры служат для возврата значений.