**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

**ПРОВЕРКА ВОЗМОЖНОСТИ ТРЕХ ОТРЕЗКОВ БЫТЬ СТОРОНАМИ ПРЯМОУГОЛЬНОГО ТРЕУГОЛЬНИКА**

**Пояснительная записка**

Исполнитель:

студент группы БПИ-191

/ В. Е. Бобров /

«1» ноября 2020 г.

**Москва 2020**

1. ТЕКСТ ЗАДАНИЯ

Разработать программу, которая по параметрам трёх отрезков (задаются декартовыми координатами концов отрезков в виде целых машинных слов без знака) решает, могут ли являться эти отрезки сторонами прямоугольного треугольника

1. РАСЧЕТНЫЕ МЕТОДЫ

При выполнении микропроекта использовалась теорема Пифагора. С ее помощью находились квадраты длин отрезков по их координатам, а также проверялось, могут ли эти отрезки быть сторонами прямоугольного треугольника.

1. ДОПУСТИМЫЕ ЗНАЧЕНИЯ

Для каждой из координат допустимым значением считается целое число от 0 до 65535 включительно.

1. ТЕСТОВЫЕ ДАННЫЕ

Вводится всегда 12 чисел по порядку: x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6. Первые 4 – для первого отрезка, далее 4 – для второго отрезка, последние 4 – для третьего отрезка.

1. Проверим Пифагорову тройку (3 4 5)

Входные данные: 0 0 0 3 0 0 0 4 0 0 0 5

Выходные данные: Yes (см. скрин 1)

1. Проверим, что смещенные параллельно отрезки тоже работают

Входные данные: 1 2 1 5 3 4 3 8 6 7 6 12

Выходные данные: Yes (см. скрин 2)

1. Проверим “непифагорову” тройку (5 6 7)

Входные данные: 0 0 0 5 0 0 0 6 0 0 0 7

Выходные данные: No (см. скрин 3)

1. Проверим что смещенные параллельно отрезки тоже работают

Входные данные: 2 3 2 8 4 5 4 11 10 11 10 18

Выходные данные: No (см. скрин 4)

1. Проверим, что если отрезок нулевой длины, то это уже не треугольник (даже если теорема Пифагора выполняется)

Входные данные: 1 4 1 4 0 0 0 4 0 0 0 5

Выходные данные: No (см. скрин 5)

1. Проверим граничные длины (составим самый большой подходящий треугольник)

Входные данные: 0 0 0 65535 0 0 65535 0 0 65535 65535 0

Выходные данные: Yes (см. скрин 6)

1. Если на любом вводе ввести число, выходящее за границы, придется ввести заново (см. скрин 7)
2. ТЕКСТ ПРОГРАММЫ

format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

; Input data

x1 dd ?

x2 dd ?

x3 dd ?

x4 dd ?

x5 dd ?

x6 dd ?

y1 dd ?

y2 dd ?

y3 dd ?

y4 dd ?

y5 dd ?

y6 dd ?

; Temp data

dx1 dd ? ; delta(x1, x2) = x1 - x2

dx2 dd ? ; delta(x3, x4)

dx3 dd ?

dy1 dd ?

dy2 dd ?

dy3 dd ?

sdx1 dd ? ; squared(delta(x1, x2)) = (x1 - x2)^2

sdx2 dd ?

sdx3 dd ?

sdy1 dd ?

sdy2 dd ?

sdy3 dd ?

sl1 dd ? ; squared length of first section

sl2 dd ?

sl3 dd ?

; strings for printf

strScanInt db '%d', 0

strScanUint db '%u', 0

strNewline db 10, 13, 0

strEnterSection1 db 'Enter section 1:', 10, 13, 0

strEnterSection2 db 'Enter section 2:', 10, 13, 0

strEnterSection3 db 'Enter section 3:', 10, 13, 0

strEnterX1Coordinate db 'Enter x1 coordinate: ', 0

strEnterY1Coordinate db 'Enter y1 coordinate: ', 0

strEnterX2Coordinate db 'Enter x2 coordinate: ', 0

strEnterY2Coordinate db 'Enter y2 coordinate: ', 0

strYes db 'Yes', 10, 13, 0

strNo db 'No', 10, 13, 0

NULL = 0

tempStack dd ? ; for returning from functions

section '.code' code readable executable

start:

call Input ; Input data

call RightTriangle ; Decide whether it can be a triangle

finish:

call [getch]

push NULL

call [ExitProcess]

Input:

mov [tempStack], esp ; remember stack position

push strEnterSection1 ; enter section 1

call [printf]

EnterX1: ; enter x1

push strEnterX1Coordinate

call [printf]

push x1

push strScanInt

call [scanf]

cmp [x1], 0 ; reenter if out of bounds

jl EnterX1

cmp[x1], 65535

jg EnterX1

EnterY1: ; same for all coordinates

push strEnterY1Coordinate

call [printf]

push y1

push strScanInt

call [scanf]

cmp [y1], 0

jl EnterY1

cmp[y1], 65535

jg EnterY1

EnterX2:

push strEnterX2Coordinate

call [printf]

push x2

push strScanInt

call [scanf]

cmp [x2], 0

jl EnterX2

cmp[x2], 65535

jg EnterX2

EnterY2:

push strEnterY2Coordinate

call [printf]

push y2

push strScanInt

call [scanf]

cmp [y2], 0

jl EnterY2

cmp[y2], 65535

jg EnterY2

push strEnterSection2 ; same for all sections

call [printf]

EnterX3:

push strEnterX1Coordinate

call [printf]

push x3

push strScanInt

call [scanf]

cmp [x3], 0

jl EnterX3

cmp[x3], 65535

jg EnterX3

EnterY3:

push strEnterY1Coordinate

call [printf]

push y3

push strScanInt

call [scanf]

cmp [y3], 0

jl EnterY3

cmp[y3], 65535

jg EnterY3

EnterX4:

push strEnterX2Coordinate

call [printf]

push x4

push strScanInt

call [scanf]

cmp [x4], 0

jl EnterX4

cmp[x4], 65535

jg EnterX4

EnterY4:

push strEnterY2Coordinate

call [printf]

push y4

push strScanInt

call [scanf]

cmp [y4], 0

jl EnterY4

cmp[y4], 65535

jg EnterY4

push strEnterSection3

call [printf]

EnterX5:

push strEnterX1Coordinate

call [printf]

push x5

push strScanInt

call [scanf]

cmp [x5], 0

jl EnterX5

cmp[x5], 65535

jg EnterX5

EnterY5:

push strEnterY1Coordinate

call [printf]

push y5

push strScanInt

call [scanf]

cmp [y5], 0

jl EnterY5

cmp[y5], 65535

jg EnterY5

EnterX6:

push strEnterX2Coordinate

call [printf]

push x6

push strScanInt

call [scanf]

cmp [x6], 0

jl EnterX6

cmp[x6], 65535

jg EnterX6

EnterY6:

push strEnterY2Coordinate

call [printf]

push y6

push strScanInt

call [scanf]

cmp [y6], 0

jl EnterY6

cmp[y6], 65535

jg EnterY6

jmp EndInput ; return from function

EndInput:

mov esp, [tempStack] ; restore stack position

ret ; return from function

RightTriangle:

mov eax, [x1] ; count x1 - x2

sub eax, [x2]

mov [dx1], eax

mov ecx, [dx1] ; count (x1 - x2)^2

imul ecx, [dx1]

mov [sdx1], ecx

mov eax, [y1] ; same for y1, y2

sub eax, [y2]

mov [dy1], eax

mov ecx, [dy1]

imul ecx, [dy1]

mov [sdy1], ecx

mov eax, [sdx1] ; count squared length of first section, using Pifagor's theorem

add eax, [sdy1]

mov [sl1], eax

cmp [sl1], 0 ; if section's length equals to 0, then it can't be a triangle

je No

mov eax, [x3] ; same for all sections

sub eax, [x4]

mov [dx2], eax

mov ecx, [dx2]

imul ecx, [dx2]

mov [sdx2], ecx

mov eax, [y3]

sub eax, [y4]

mov [dy2], eax

mov ecx, [dy2]

imul ecx, [dy2]

mov [sdy2], ecx

mov eax, [sdx2]

add eax, [sdy2]

mov [sl2], eax

cmp [sl2], 0

je No

mov eax, [x5]

sub eax, [x6]

mov [dx3], eax

mov ecx, [dx3]

imul ecx, [dx3]

mov [sdx3], ecx

mov eax, [y5]

sub eax, [y6]

mov [dy3], eax

mov ecx, [dy3]

imul ecx, [dy3]

mov [sdy3], ecx

mov eax, [sdx3]

add eax, [sdy3]

mov [sl3], eax

cmp [sl3], 0

je No

mov eax, [sl1] ; check three combinations of sides using Pifagor's theorem

add eax, [sl2]

cmp eax, [sl3]

je Yes

mov eax, [sl2]

add eax, [sl3]

cmp eax, [sl1]

je Yes

mov eax, [sl1]

add eax, [sl3]

cmp eax, [sl2]

je Yes

No: ; can't be a triangle

push strNo

call [printf]

jmp EndRightTriangle

Yes: ; can be a triangle

push strYes

call [printf]

jmp EndRightTriangle

EndRightTriangle:

mov esp, [tempStack]

ret

section '.idata' import data readable

library kernel, 'kernel32.dll',\

msvcrt, 'msvcrt.dll'

import kernel,\

ExitProcess, 'ExitProcess'

import msvcrt,\

printf, 'printf',\

getch, '\_getch',\

scanf, 'scanf'

## **ИСТОЧНИКИ:**

1. YouTube Byte++ FASM [Электронный ресурс] – Режим доступа: https://www.youtube.com/playlist?list=PLH3y3SWteZd3Pwn81m\_Z-iHp3imgkVUcs, свободный. (дата обращения: 01.11.20)