Арутюнян Давид Арменович
НИУ ВШЭ ФКН Программная инженерия 2 курс
БПИ192 1 подгруппа

Программа на FASM, которая по координатам четырёх точек (задаются целыми без знака) решает, образуют ли заданные точки квадрат.

1. Текст задания

«Разработать программу, которая по координатам четырёх точек (задаются целыми без знака) решает, образуют ли заданные точки квадрат»

2. Применяемые расчетные методы

A(x1, y1), B(x2, y2), C(x3, y3), D(x4, y4) — данные точки.

s_i — расстояние от одной точки до другой точки.

s_1 = (x1 − x2)² + (y1 − y2)²,

s_2 = (x1 − x3)² + (y1 − y3)²,

s_3 = (x1 − x4)² + (y1 − y4)²,

s_4 = (x2 − x3)² + (y2 − y3)²,

s_5 = (x2 − x4)² + (y2 − y4)²,

s_6 = (x3 − x4)² + (y3 − y4)².

Если (($s_1 = s_3 = s_4 = s_6$!= 0) & ($s_2 = s_5 = 2s_1$!= 0)) ⇒ ABCD — квадрат;

Если (($s_1 = s_2 = s_5 = s_6$!= 0) & ($s_3 = s_4 = 2s_1$!= 0)) ⇒ ABDC — квадрат;

Если (($s_2 = s_3 = s_4 = s_5$!= 0) & ($s_1 = s_6 = 2s_2$!= 0)) ⇒ ACBD — квадрат.

3. Используемые источники

- Ответы MailRu [электронный ресурс] — https://otvet.mail.ru/
- SoftCraft [электронный ресурс] — http://www.softcraft.ru/

4. Текст программы

```
format PE console

entry start

include 'win32a.inc'

section '.data' data readable writeable
        strScanInt db '%d', 10, 0
        coords db 'Point %d: (%d, %d)', 10, 0
        scanCoords db '%d %d', 0
        strPoint1 db 'Enter the coordinates of the first point: ', 0
```

```
        strPoint2 db 'Enter the coordinates of the second point: ', 0
        strPoint3 db 'Enter the coordinates of the third point: ', 0
        strPoint4 db 'Enter the coordinates of the fourth point: ', 0
        strTrue db 'Yes, it is a square!', 0
        strFalse db 'No, it is not a square.', 0
        delim db 10, '--------------------', 10, 10, 0

        s1 dd ?
        s2 dd ?
        s3 dd ?
        s4 dd ?
        s5 dd ?
        s6 dd ?

        point1_x dd ?
        point1_y dd ?
        point2_x dd ?
        point2_y dd ?
        point3_x dd ?
        point3_y dd ?
        point4_x dd ?
        point4_y dd ?

        null dd 0

section '.code' code readable executable

        start:
                ; entering the first point
                push strPoint1
                call [printf]

                push point1_y
                push point1_x
                push scanCoords
                call [scanf]
                ; end

                ; entering the second point
                push strPoint2
                call [printf]

                push point2_y
                push point2_x
                push scanCoords
                call [scanf]
                ; end

                ; entering the third point
                push strPoint3
                call [printf]

                push point3_y
                push point3_x
```

```
push scanCoords
call [scanf]
; end

; entering the fourth point
push strPoint4
call [printf]

push point4_y
push point4_x
push scanCoords
call [scanf]
; end

; displaying information about points
push delim
call [printf]

cinvoke printf, coords, 1, [point1_x], [point1_y]
cinvoke printf, coords, 2, [point2_x], [point2_y]
cinvoke printf, coords, 3, [point3_x], [point3_y]
cinvoke printf, coords, 4, [point4_x], [point4_y]

push delim
call [printf]
; end

; calculate s1: (x1 - x2)^2 + (y1 - y2)^2
mov eax, [point1_x]
sub eax, [point2_x]
imul eax, eax
mov [s1], eax

xor eax, eax
mov eax, [point1_y]
sub eax, [point2_y]
imul eax, eax
add [s1], eax

; cinvoke printf, strScanInt, [s1]
; end s1

; calculate s2: (x1 - x3)^2 + (y1 - y3)^2
mov eax, [point1_x]
sub eax, [point3_x]
imul eax, eax
mov [s2], eax

xor eax, eax
mov eax, [point1_y]
sub eax, [point3_y]
imul eax, eax
add [s2], eax
```

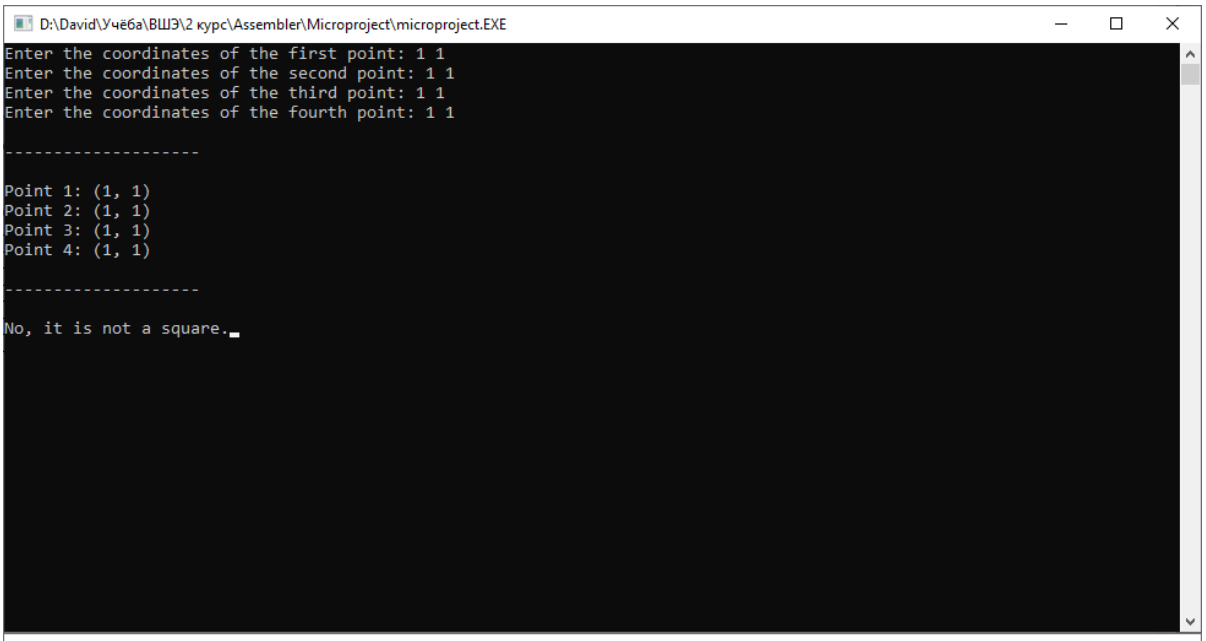
```
; cinvoke printf, strScanInt, [s2]
; end s2

; calculate s3: (x1 - x4)^2 + (y1 - y4)^2
mov eax, [point1_x]
sub eax, [point4_x]
imul eax, eax
mov [s3], eax

xor eax, eax
mov eax, [point1_y]
sub eax, [point4_y]
imul eax, eax
add [s3], eax

; cinvoke printf, strScanInt, [s3]
; end s3

; calculate s4: (x2 - x3)^2 + (y2 - y3)^2
mov eax, [point2_x]
sub eax, [point3_x]
imul eax, eax
mov [s4], eax

xor eax, eax
mov eax, [point2_y]
sub eax, [point3_y]
imul eax, eax
add [s4], eax

; cinvoke printf, strScanInt, [s4]
; end s4

; calculate s5: (x2 - x4)^2 + (y2 - y4)^2
mov eax, [point2_x]
sub eax, [point4_x]
imul eax, eax
mov [s5], eax

xor eax, eax
mov eax, [point2_y]
sub eax, [point4_y]
imul eax, eax
add [s5], eax

; cinvoke printf, strScanInt, [s5]
; end s5

; calculate s6: (x3 - x4)^2 + (y3 - y4)^2
mov eax, [point3_x]
sub eax, [point4_x]
imul eax, eax
mov [s6], eax
```

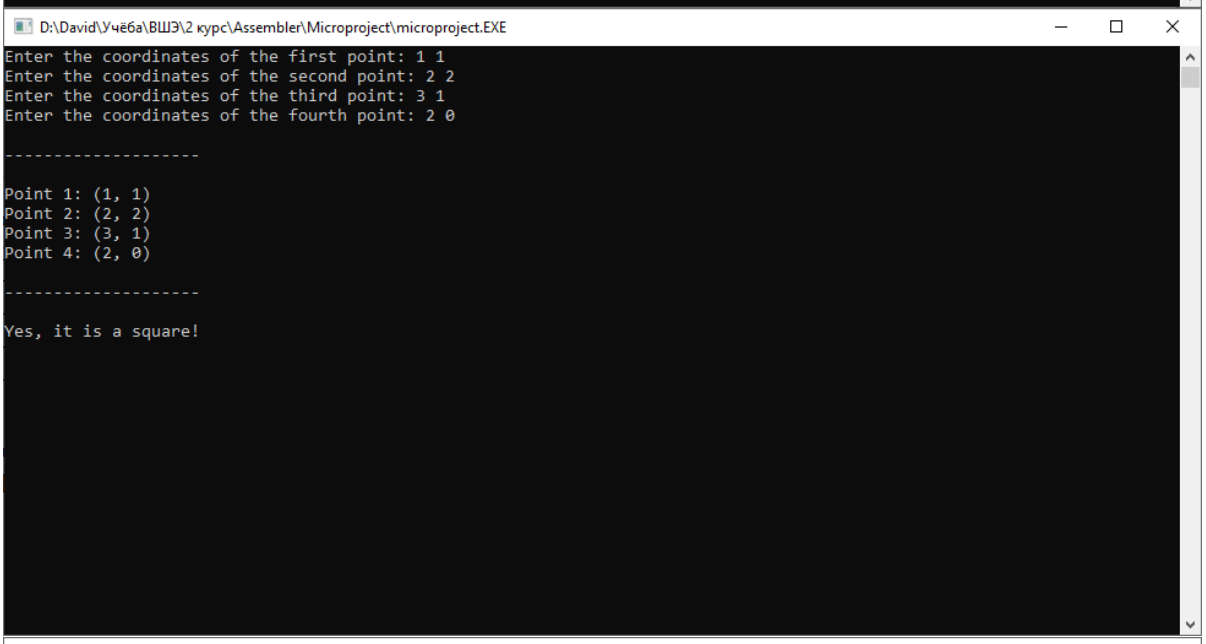
```
        xor eax, eax
        mov eax, [point3_y]
        sub eax, [point4_y]
        imul eax, eax
        add [s6], eax

        ; cinvoke printf, strScanInt, [s6]
        ; end s6

        ; checking
        FirstCondition: ; (s1 = s3 = s4 = s6 != 0) & (s2 = s5 = 2s1 != 0) => It's a square

            xor ebx, ebx ; ebx = 0
            mov ebx, [s3]
            cmp [s1], ebx
            jne SecondCondition
            mov ebx, [s4]
            cmp [s1], ebx
            jne SecondCondition
            mov ebx, [s6]
            cmp [s1], ebx
            jne SecondCondition
            mov ebx, 0
            cmp [s1], ebx
            je NotASquare

            xor ebx, ebx ; ebx = 0
            mov ebx, [s5]
            cmp [s2], ebx
            jne SecondCondition
            mov ebx, [s1]
            imul ebx, 2
            cmp [s2], ebx
            jne SecondCondition
            mov ebx, 0
            cmp [s2], ebx
            je NotASquare

            push strTrue ; It is a square
            jmp Output ; output and finish

        SecondCondition: ; (s1 = s2 = s5 = s6 != 0) & (s3 = s4 = 2s1 != 0) => It's a
square

            xor ebx, ebx ; ebx = 0
            mov ebx, [s2]
            cmp [s1], ebx
            jne ThirdCondition
            mov ebx, [s5]
            cmp [s1], ebx
            jne ThirdCondition
            mov ebx, [s6]
            cmp [s1], ebx
            jne ThirdCondition
```

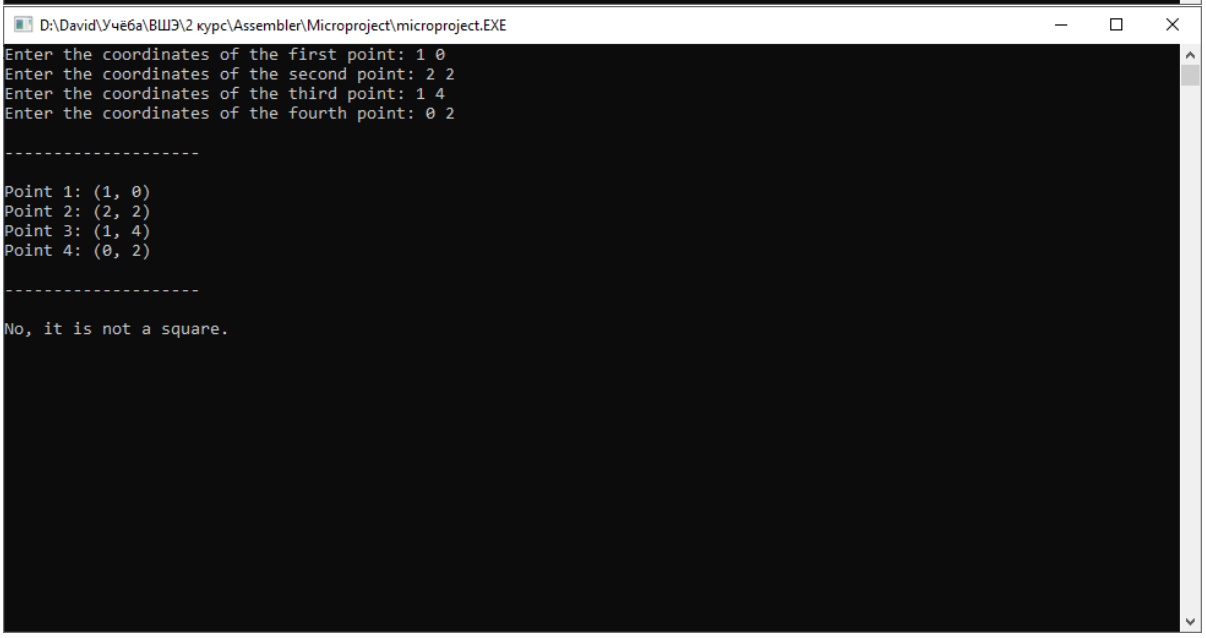
```
        mov ebx, 0
        cmp [s1], ebx
        je NotASquare

        xor ebx, ebx ; ebx = 0
        mov ebx, [s4]
        cmp [s3], ebx
        jne ThirdCondition
        mov ebx, [s1]
        imul ebx, 2
        cmp [s3], ebx
        jne ThirdCondition
        mov ebx, 0
        cmp [s3], ebx
        je NotASquare

        push strTrue ; It is a square
        jmp Output ; output and finish

ThirdCondition: ; (s2 = s3 = s4 = s5 != 0) & (s1 = s6 = 2s2 != 0) => It's a square

        xor ebx, ebx ; ebx = 0
        mov ebx, [s3]
        cmp [s2], ebx
        jne NotASquare
        mov ebx, [s4]
        cmp [s2], ebx
        jne NotASquare
        mov ebx, [s5]
        cmp [s2], ebx
        jne NotASquare
        mov ebx, 0
        cmp [s2], ebx
        je NotASquare

        xor ebx, ebx ; ebx = 0
        mov ebx, [s6]
        cmp [s1], ebx
        jne NotASquare
        mov ebx, [s2]
        imul ebx, 2
        cmp [s1], ebx
        jne NotASquare
        mov ebx, 0
        cmp [s1], ebx
        je NotASquare

        push strTrue ; It is a square
        jmp Output ; output and finish

NotASquare:

        push strFalse ; It is not a square
        jmp Output ; output and finish
```

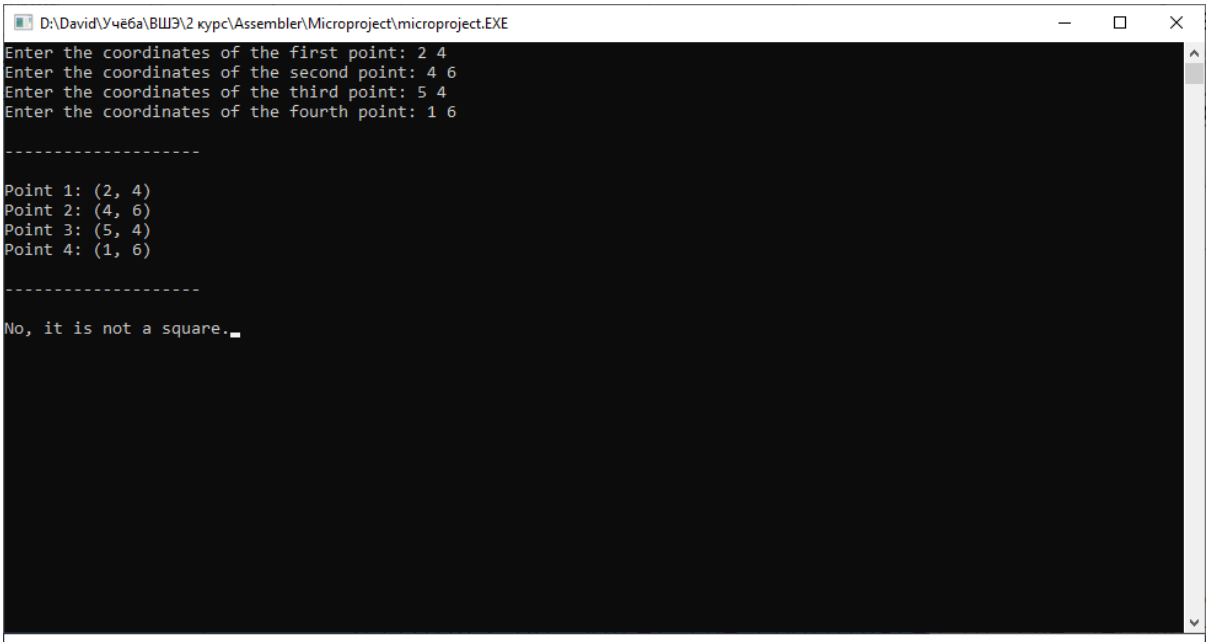
```
        Output:

            call [printf]

    finish:

        call [getch]

        push [null]
        call [ExitProcess]

section '.idata' import data readable

    library kernel, 'kernel32.dll',\
        msvcrt, 'msvcrt.dll',\
        user32,'USER32.DLL'

    include 'api\user32.inc'
    include 'api\kernel32.inc'

    import kernel,\
       ExitProcess, 'ExitProcess',\
       HeapCreate,'HeapCreate',\
       HeapAlloc,'HeapAlloc'

    include 'api\kernel32.inc'

    import msvcrt,\
       printf, 'printf',\
       scanf, 'scanf',\
       getch, '_getch'
```

5. Тесты


D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 1 1
Enter the coordinates of the second point: 1 1
Enter the coordinates of the third point: 1 1
Enter the coordinates of the fourth point: 1 1
Point 1: (1, 1)
Point 2: (1, 1)
Point 3: (1, 1)
Point 4: (1, 1)
No, it is not a square.

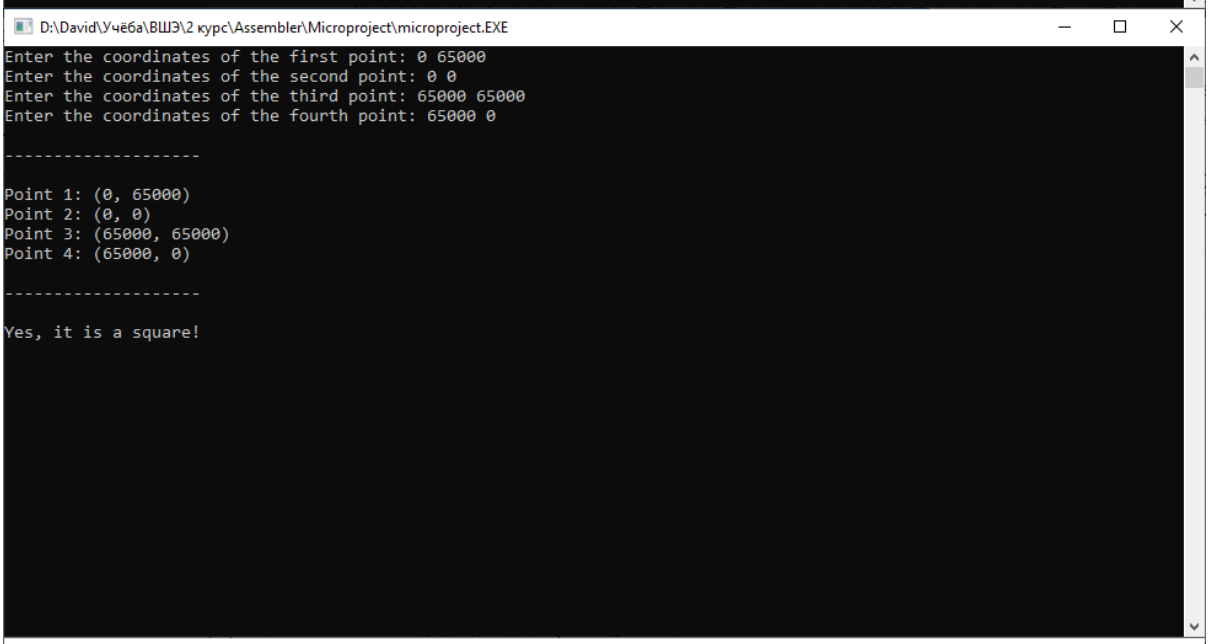

D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 1 1
Enter the coordinates of the second point: 2 2
Enter the coordinates of the third point: 3 1
Enter the coordinates of the fourth point: 2 0
Point 1: (1, 1)
Point 2: (2, 2)
Point 3: (3, 1)
Point 4: (2, 0)
Yes, it is a square!


D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 1 0
Enter the coordinates of the second point: 2 2
Enter the coordinates of the third point: 1 4
Enter the coordinates of the fourth point: 0 2
Point 1: (1, 0)
Point 2: (2, 2)
Point 3: (1, 4)
Point 4: (0, 2)
No, it is not a square.


D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 2 4
Enter the coordinates of the second point: 4 6
Enter the coordinates of the third point: 5 4
Enter the coordinates of the fourth point: 1 6

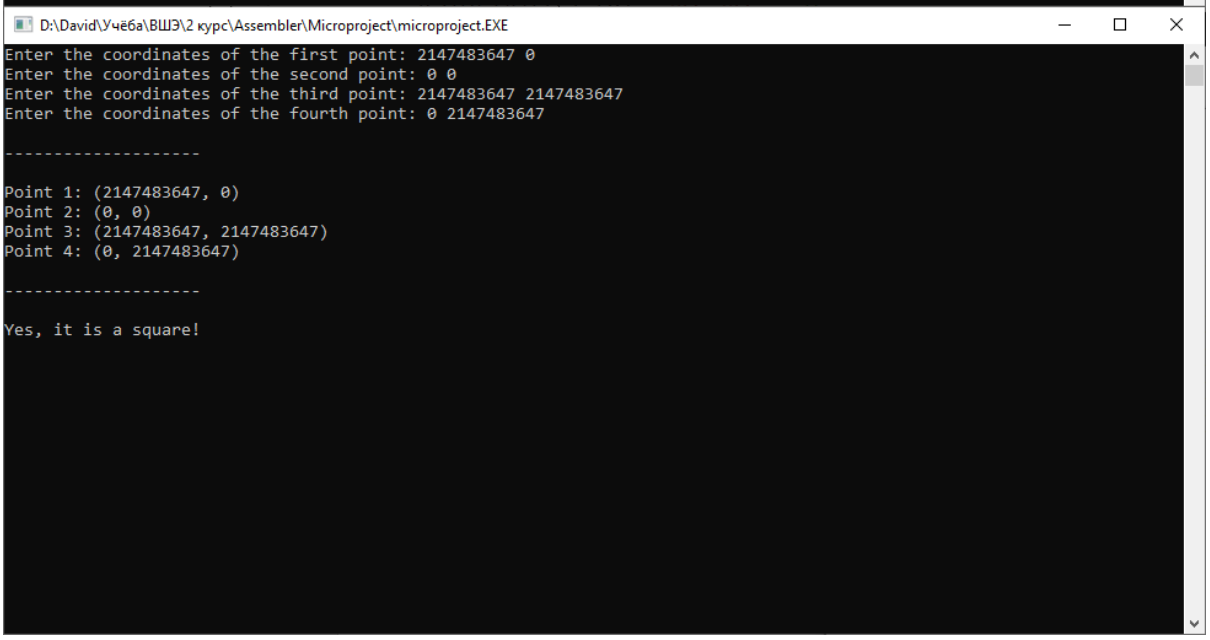
Point 1: (2, 4)
Point 2: (4, 6)
Point 3: (5, 4)
Point 4: (1, 6)

No, it is not a square.


D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 0 65000
Enter the coordinates of the second point: 0 0
Enter the coordinates of the third point: 65000 65000
Enter the coordinates of the fourth point: 65000 0

Point 1: (0, 65000)
Point 2: (0, 0)
Point 3: (65000, 65000)
Point 4: (65000, 0)

Yes, it is a square!


D:\David\Учёба\ВШЭ\2 курс\Assembler\Microproject\microproject.EXE
Enter the coordinates of the first point: 2147483647 0
Enter the coordinates of the second point: 0 0
Enter the coordinates of the third point: 2147483647 2147483647
Enter the coordinates of the fourth point: 0 2147483647

Point 1: (2147483647, 0)
Point 2: (0, 0)
Point 3: (2147483647, 2147483647)
Point 4: (0, 2147483647)

Yes, it is a square!

6. Приложения

a. Текст программы: сама программа находится в файле microproject.asm
b. Тесты: входные значения в файле input.txt, выходные данные в файле output.txt
c. Входные данные программы: целые числа в диапазоне от 0 до $2^{31} - 1$