

# Пояснительная записка по микропроекту №1

ВУЗ: НИУ "Высшая школа экономики"

Факультет: ФКН

Образовательная программа: Программная инженерия

Название разработки: "Микропроект №1 (Square)"

Студент: Елесин Сергей Владимирович

Группа: БПИ-193

Вариант: 5

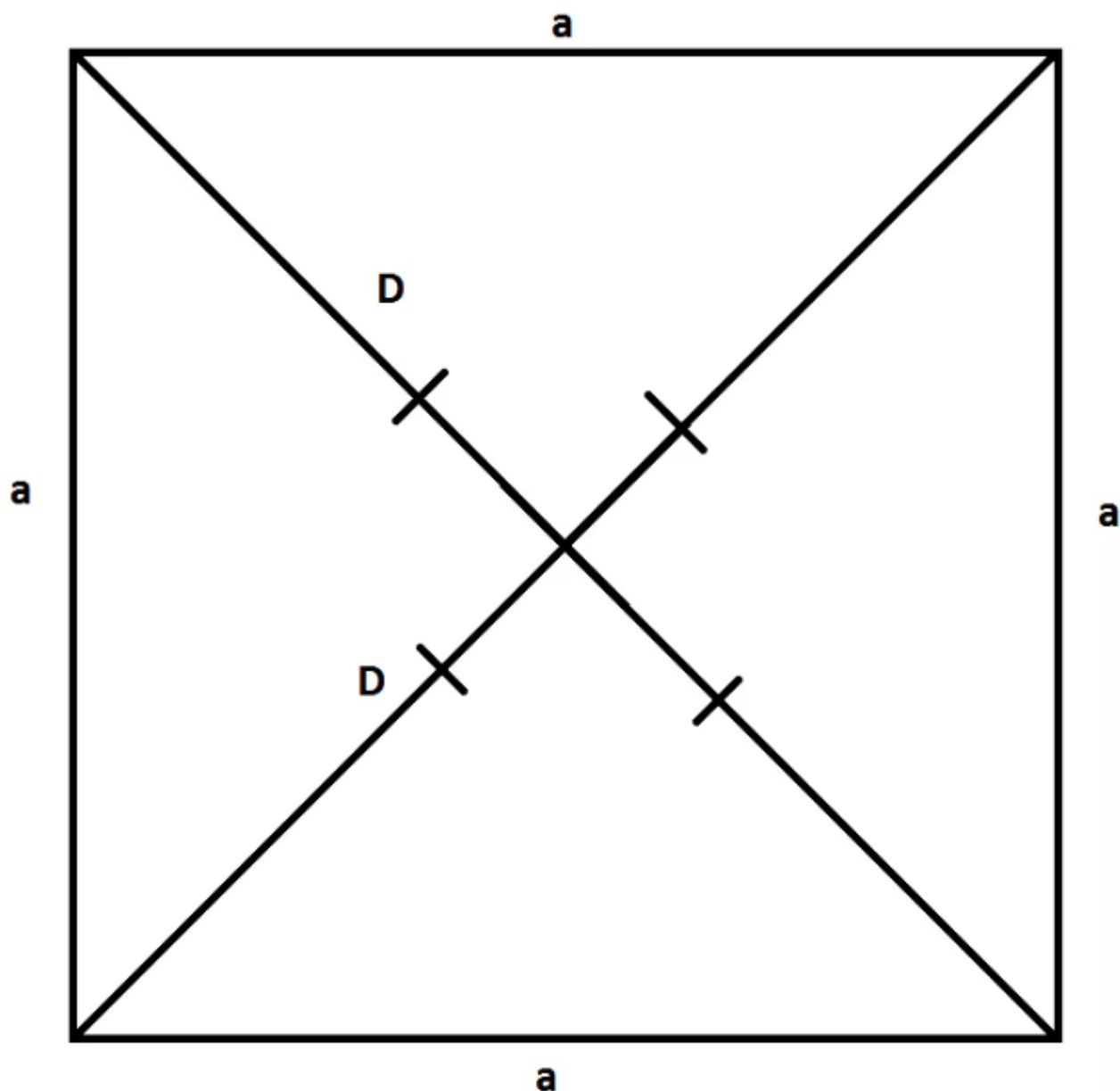
## Задание:

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

## Идея решения:

Из школьного курса геометрии мы знаем, что у квадрата все стороны равны. Однако это же свойство относится и к ромбу. В отличие от ромба, у квадрата диагонали равны.

Следовательно, набор точек является квадратом тогда и только тогда, когда у образованного им четырёхугольника 4 стороны равны и 2 диагонали равны. То есть, среди всех 6 попарных расстояний между точками: 4 равных расстояния (стороны) + ещё 2 равных расстояния (диагонали).



### Алгоритм:

1. Считываем координаты 4-х точек
2. Вычисляем все 6 попарных расстояний между точками по формуле  $(x_1 - x_2)^2 + (y_1 - y_2)^2$  (мы не вычисляем корень, так как нам не нужно знать точное расстояние. Для простоты и эффективности мы будем сравнивать квадраты расстояний)
3. Если все 6 попарных расстояний можно разделить по величине на 2 категории, в которых будет 4 и 2 значения соответственно, то точки образуют квадрат

### Ограничения на входные данные:

- числа должны быть неотрицательными (по условию)
- числа должны быть меньше  $2^{15}$  (иначе расстояние между точками может не влезть в 32-битную переменную)

### Исходный код программы:

```
format PE console
entry start
```

```
; Студент: Елесин Сергей
; Группа: БПИ-193
; Вариант: 5
```

```
include 'win32a.inc'
```

```
;----- Data -----
```

```
section '.data' data readable writable
```

```
    strInputCoords      db 'Input the coordinates of point %d: ', 0
    strScanCoords       db '%d %d', 0
    strNotSquare         db 'NOT SQUARE!', 10, 0
    strIsSquare          db 'IT IS SQUARE!!!', 10, 0
    strErrorSameCoords   db 'Incorrect input. Points cannot have the same coordinates',
    strErrorValueLessZero db 'Incorrect input. All values must be greater or equal zero',
    strErrorValueVeryBig db 'Incorrect input. All values must be less than 32768', 10, 0
```

```
    x                   dd ?
    y                   dd ?
    i                   dd ?
    count1              dd 0
    count2              dd 0
    length              dd 0
    value1              dd 0
    value2              dd 0
    counter1            dd 0
    counter2            dd 0
    Xpoints             rd 4 ; массив координат точек по X
    Ypoints             rd 4 ; массив координат точек по Y
```

```
;----- Code -----
```

```
section '.code' code readable executable
```

```
start:
```

```
; ввод координат точек
    call PointsInput
; вычисляем все попарные расстояния между точками
    call PassingAllPairs
; определяем, является ли набор точек квадратом
    call GetResult
```

```
; ниже представлены несколько вариантов завершения программы
```

```
finish:
```

```
    call [getch]
    push 0
    call [ExitProcess]
```

```
notSquare:
```

```
    push strNotSquare
    call [printf]
    jmp finish
```

isSquare:

```
push strIsSquare
call [printf]
jmp finish
```

errorSameCoords:

```
push strErrorSameCoords
call [printf]
jmp finish
```

errorValueLessZero:

```
push strErrorValueLessZero
call [printf]
jmp finish
```

errorValueVeryBig:

```
push strErrorValueVeryBig
call [printf]
jmp finish
```

;-----

PointsInput:

; задаём начальные значения

```
xor ecx, ecx
mov eax, Xpoints
mov ebx, Ypoints
```

beginInput:

; сохраняем значения регистров во временные переменные

```
mov [x], eax
mov [y], ebx
mov [i], ecx
```

; просим пользователя ввести координаты

```
push ecx
push strInputCoords
call [printf]
add esp, 8
```

; восстанавливаем значения регистров

```
mov eax, [x]
mov ebx, [y]
```

; считываем координаты

```
push ebx
push eax
push strScanCoords
call [scanf]
add esp, 12
```

; восстанавливаем значения регистров

```
mov eax, [x]
mov ebx, [y]
mov ecx, [i]
```

; проверяем, что введенные координаты не меньше нуля

```
cmp [Xpoints + ecx*4], 0
jnl errorValueLessZero
cmp [Ypoints + ecx*4], 0
```

```

        jl errorValueLessZero
; проверяем, что введенные координаты меньше 2^15
        cmp [Xpoints + ecx*4], 32768
        jge errorValueVeryBig
        cmp [Ypoints + ecx*4], 32768
        jge errorValueVeryBig
; переходим на следующую итерацию
        add eax, 4
        add ebx, 4
        inc ecx
        cmp ecx, 4
        jne beginInput

        ret

```

```

;-----
PassingAllPairs:

```

```

loop1:
        mov edx, [count1]
        mov [count2], edx
        inc [count2]
; используем вложенный цикл, чтобы пройти по всем парам точек
        loop2:
; вычисляем расстояние между точками
            call CalculateLength
; "обрабатываем" расстояние
            call CheckLength

            inc [count2]
            cmp [count2], 4
            jl loop2

        inc [count1]
        cmp [count1], 3
        jl loop1

        ret

```

```

;-----
CalculateLength:
        mov edx, [count1]
        imul edx, 4
; записываем в регистры координаты первой точки
        mov eax, [Xpoints + edx]
        mov ebx, [Ypoints + edx]

        mov edx, [count2]
        imul edx, 4
; вычитаем из координат первой точки координаты второй
        sub eax, [Xpoints + edx]
        sub ebx, [Ypoints + edx]
; возводим в квадрат
        imul eax, eax
        imul ebx, ebx

```

```

; складываем полученные значения
    add eax, ebx
    mov [length], eax

    ret

;-----
CheckLength:
    mov eax, [length]
; выводим ошибку, если расстояние между точками равно 0
    cmp eax, 0
    je errorSameCoords

; если значение нулевое, заносим в него текущую длину
    cmp [value1], 0
    jne compareWithValue1
    mov [value1], eax

; если текущая длина равна значению value1, то увеличиваем первый счётчик
compareWithValue1:
    cmp [value1], eax
    jne checkValue2

    inc [counter1]
    ret

checkValue2:
; если значение нулевое, заносим в него текущую длину
    cmp [value2], 0
    jne compareWithValue2
    mov [value2], eax

; если текущая длина равна значению value2, то увеличиваем второй счётчик
compareWithValue2:
    cmp [value2], eax
    jne notSquare

    inc [counter2]
    ret

;-----
GetResult:
; точки образуют квадрат, если все 6 попарных расстояний можно разделить
; по величине на 2 категории, в которых будет 4 и 2 значения соответственно
    cmp [counter1], 4
    jne anotherCase

    cmp [counter2], 2
    jne notSquare

    jmp isSquare

anotherCase:
    cmp [counter1], 2
    jne notSquare

```

```

    cmp [counter2], 4
    jne notSquare

    jmp isSquare
;----- Imports -----

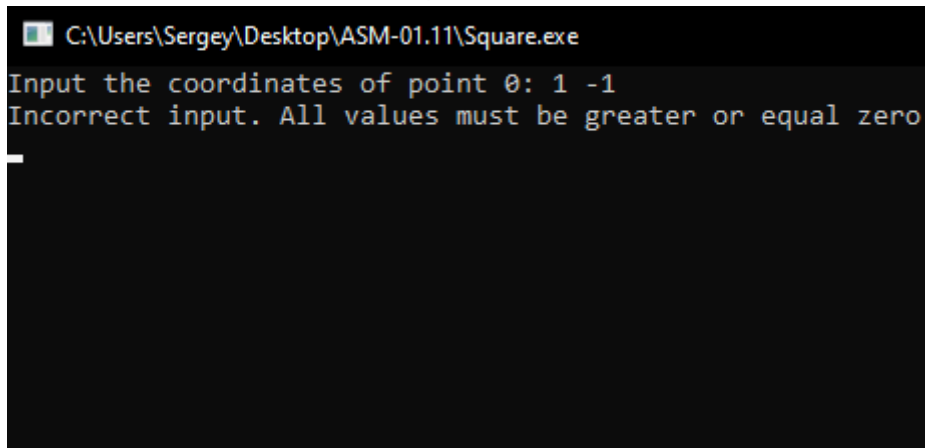
section '.idata' import data readable
    library kernel, 'kernel32.dll',\
        msvcrt, 'msvcrt.dll',\
        user32, 'USER32.DLL'

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'

```

## Тесты:

1. Ввод отрицательных чисел (по условию разрешены только неотрицательные координаты):

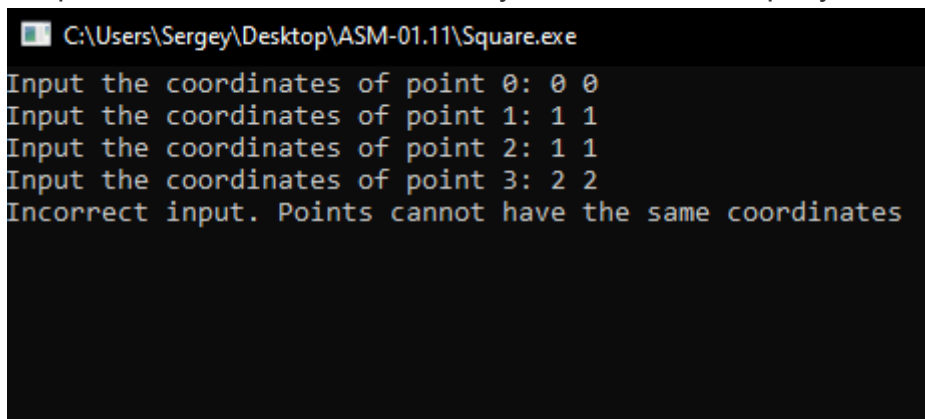


```

C:\Users\Sergey\Desktop\ASM-01.11\Square.exe
Input the coordinates of point 0: 1 -1
Incorrect input. All values must be greater or equal zero

```

2. Ввод точек, расположенных на одной координате (2 точки не могут иметь одинаковые координаты, иначе точки не смогут составить четырёхугольник):



```

C:\Users\Sergey\Desktop\ASM-01.11\Square.exe
Input the coordinates of point 0: 0 0
Input the coordinates of point 1: 1 1
Input the coordinates of point 2: 1 1
Input the coordinates of point 3: 2 2
Incorrect input. Points cannot have the same coordinates

```

3. Ввод слишком больших чисел (см. раздел "Ограничения входных данных"):

```
C:\Users\Sergey\Desktop\ASM-01.11\Square.exe
Input the coordinates of point 0: 123456 123456
Incorrect input. All values must be less than 32768
```

4. Ввод точек, НЕ образующих квадрат:

```
C:\Users\Sergey\Desktop\ASM-01.11\Square.exe
Input the coordinates of point 0: 0 0
Input the coordinates of point 1: 1 0
Input the coordinates of point 2: 0 2
Input the coordinates of point 3: 3 3
NOT SQUARE!
```

5. Ввод точек, образующих квадрат:

```
C:\Users\Sergey\Desktop\ASM-01.11\Square.exe
Input the coordinates of point 0: 0 0
Input the coordinates of point 1: 5 0
Input the coordinates of point 2: 0 5
Input the coordinates of point 3: 5 5
IT IS SQUARE!!!
```