**Микропроект №1.**

**Определение коллинеарности точек**

Штырба Владислав Анатольевич

НИУ ВШЭ

Факультет Компьютерных наук

Группа БПИ197

**Текст задания**

Вариант 26

Разработать программу, решающую вопрос о коллинеарности N точек (координаты точек задать целыми со знаком, N=5).

**Применяемые расчётные методы**

В качестве проверки было использована проверка на построение треугольника по координатам точки.

**Используемые источники**

Площадь треугольника через определитель: https://www.matematicus.ru/vysshaya-matematika/analiticheskaya-geometriya-na-ploskosti/ploshhad-treugolnika-na-ploskosti

**Исходный код**

**;Задание:**

**;Разработать программу, решающую вопрос о коллинеарности N точек (координаты точек задать целыми со знаком, N=5).**

**format PE console**

**entry start**

**include 'win32a.inc'**

**section '.data' data readable writable**

**;Строки используемые в программе**

**positiveStr db 'The points are collinear', 0**

**negativeStr db 'The points are not collinear', 0**

**enterStr db 'Enter x%d and y%d', 10, 0**

**digitInOut db '%d', 10, 0**

**coorIn db '%d %d', 0**

**coorOut db '%d %d', 10, 0**

**tmpStack dd ?**

**x1 dd ? ;координата x1**

**x2 dd ? ;координата x2**

**x3 dd ? ;координата x3**

**x4 dd ? ;координата x4**

**x5 dd ? ;координата x5**

**y1 dd ? ;координата y1**

**y2 dd ? ;координата y2**

**y3 dd ? ;координата y3**

**y4 dd ? ;координата y4**

**y5 dd ? ;координата y5**

**p1x dd ?**

**p2x dd ?**

**p3x dd ?**

**p1y dd ?**

**p2y dd ?**

**p3y dd ?**

**s dd 0**

**counter dd 0**

**final dd 0**

**NULL = 0**

**section '.code' code readable executable**

**start:**

**;Ввод первой точки**

**inc [counter]**

**push [counter]**

**push [counter]**

**push enterStr**

**call [printf]**

**push y1**

**push x1**

**push coorIn**

**call [scanf]**

**;Ввод второй точки**

**inc [counter]**

**push [counter]**

**push [counter]**

**push enterStr**

**call [printf]**

**push y2**

**push x2**

**push coorIn**

**call [scanf]**

**;Ввод третьей точки**

**inc [counter]**

**push [counter]**

**push [counter]**

**push enterStr**

**call [printf]**

**push y3**

**push x3**

**push coorIn**

**call [scanf]**

**;Ввод четвёртой точки**

**inc [counter]**

**push [counter]**

**push [counter]**

**push enterStr**

**call [printf]**

**push y4**

**push x4**

**push coorIn**

**call [scanf]**

**;Ввод пятой точки**

**inc [counter]**

**push [counter]**

**push [counter]**

**push enterStr**

**call [printf]**

**push y5**

**push x5**

**push coorIn**

**call [scanf]**

**;Проверка 1, 2, 3 точек на коллинеарность**

**push [y3]**

**push [x3]**

**push [y2]**

**push [x2]**

**push [y1]**

**push [x1]**

**call checkCollinear**

**add [final], eax**

**;Проверка 2, 3, 4 точек на коллинеарность**

**push [y4]**

**push [x4]**

**push [y3]**

**push [x3]**

**push [y2]**

**push [x2]**

**call checkCollinear**

**add [final], eax**

**;Проверка 3, 4, 5 точек на коллинеарность**

**push [y5]**

**push [x5]**

**push [y4]**

**push [x4]**

**push [y3]**

**push [x3]**

**call checkCollinear**

**add [final], eax**

**;Итоговая проверка на коллинеарность**

**mov eax, [final]**

**cmp eax, 0**

**je positiveResult**

**push negativeStr**

**call [printf]**

**call endProg**

**positiveResult:**

**push positiveStr**

**call [printf]**

**call endProg**

**;s = p1x \* (p2y - p3y) + p2x \* (p3y - p2y) + p3x \* (p1y - p2y)**

**;Если s = 0, то все пять точек коллинеарны**

**checkCollinear:**

**pop [tmpStack]**

**pop [p1x]**

**pop [p1y]**

**pop [p2x]**

**pop [p2y]**

**pop [p3x]**

**pop [p3y]**

**mov ebx, [p2y]**

**sub ebx, [p3y]**

**mov eax, [p1x]**

**call mult**

**add [s], eax**

**mov ebx, [p3y]**

**sub ebx, [p1y]**

**mov eax, [p2x]**

**call mult**

**add [s], eax**

**mov ebx, [p1y]**

**sub ebx, [p2y]**

**mov eax, [p3x]**

**call mult**

**add [s], eax**

**mov eax, [s]**

**push [tmpStack]**

**ret**

**;Умножение eax на ebx**

**mult:**

**xor ecx, ecx**

**cmp ebx, 0**

**jg notNegative**

**xor edx, edx**

**sub edx, ebx**

**mov ebx, edx**

**mov ecx, 1**

**notNegative:**

**mov edx, eax**

**xor eax, eax**

**multLoop:**

**cmp ebx, 0**

**je endMultLoop**

**add eax, edx**

**dec ebx**

**jmp multLoop**

**endMultLoop:**

**cmp ecx, 0**

**je notNegativeResult**

**mov ebx, eax**

**xor eax, eax**

**sub eax, ebx**

**notNegativeResult:**

**ret**

**endProg:**

**call [getch]**

**push NULL**

**call [ExitProcess]**

**section 'idata' import data readable**

**library kernel, 'kernel32.dll', \**

**msvcrt, 'msvcrt.dll'**

**import kernel, \**

**ExitProcess, 'ExitProcess'**

**import msvcrt, \**

**printf, 'printf', \**

**scanf, 'scanf', \**

**getch, '\_getch'**

**Тест программы**

Тест с не взаимно простыми числами:

test1.png

Тест с взаимно простыми числами:

test2.png