

**Міністерство освіти і науки України**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**

**Звіт**  
**про виконання лабораторної роботи №2**  
**«Реалізація алгоритмів у функціональному стилі»**

**Виконав:**  
**студентгрупи ФЕП-11с**  
**Качмар Д.Б**

**Викладач:**  
**Кушнір Олексій**  
**Олександрович**

**Львів-2025**

**Мета роботи:** Реалізація простого та ефективного методу перевірки того, чи належить задана точка трикутнику (всередині, на межі або поза ним) за допомогою методу векторного добутку.

### Виконання роботи

Посилання на GitHub: <https://github.com/D-Kachm/OOP/tree/main/Lab2>

```
h geometry.h > ...
1  #ifndef GEOMETRY_H
2  #define GEOMETRY_H
3
4  struct Point {
5      double x, y;
6  };
7
8  bool isDegenerateTriangle(const Point& A, const Point& B, const Point& C);
9  int pointInTriangle(const Point& A, const Point& B, const Point& C, const Point& P);
10
11 #endif
12
```

```
G+ geometry.cpp > ...
1  #include "geometry.h"
2  #include <cmath>
3
4  const double EPS = 1e-9;
5
6  double crossProduct(const Point& A, const Point& B, const Point& C) {
7      double x1 = B.x - A.x;
8      double y1 = B.y - A.y;
9      double x2 = C.x - A.x;
10     double y2 = C.y - A.y;
11     return x1 * y2 - y1 * x2;
12 }
13
14 bool isDegenerateTriangle(const Point& A, const Point& B, const Point& C) {
15     return std::abs(crossProduct(A, B, C)) < EPS;
16 }
17
18 int pointInTriangle(const Point& A, const Point& B, const Point& C, const Point& P) {
19     double d1 = crossProduct(A, B, P);
20     double d2 = crossProduct(B, C, P);
21     double d3 = crossProduct(C, A, P);
22
23     bool has_neg = (d1 < -EPS) || (d2 < -EPS) || (d3 < -EPS);
24     bool has_pos = (d1 > EPS) || (d2 > EPS) || (d3 > EPS);
25
26     if (!has_neg && !has_pos)
27         return 1; // On the edge
28     if (!has_neg || !has_pos)
29         return 2; // Inside
30     return 0; // Outside
31 }
32
```

```

G+ main.cpp > ...
1  #include <iostream>
2  #include "geometry.h"
3
4  int main() {
5      Point A, B, C;
6      std::cout << "Enter coordinates of the triangle (Ax Ay Bx By Cx Cy): ";
7      std::cin >> A.x >> A.y >> B.x >> B.y >> C.x >> C.y;
8
9      if (isDegenerateTriangle(A, B, C)) {
10         std::cout << "The triangle is degenerate (its area is zero).\n";
11         return 1;
12     }
13
14     int n;
15     std::cout << "Enter number of points to check: ";
16     std::cin >> n;
17
18     for (int i = 0; i < n; ++i) {
19         Point P;
20         std::cout << "Enter coordinates of point " << i + 1 << " (x y): ";
21         std::cin >> P.x >> P.y;
22
23         int result = pointInTriangle(A, B, C, P);
24
25         std::cout << "Point (" << P.x << ", " << P.y << ") is ";
26         if (result == 2)
27             std::cout << "inside the triangle.\n";
28         else if (result == 1)
29             std::cout << "on the edge of the triangle.\n";
30         else
31             std::cout << "outside the triangle.\n";
32     }
33
34     return 0;
35 }
36

```

## Результати

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Github\OOP\Lab2> ./main
Enter coordinates of the triangle (Ax Ay Bx By Cx Cy): 0 0 4 0 2 3
Enter number of points to check: 5
Enter coordinates of point 1 (x y): 2 1
Point (2, 1) is inside the triangle.
Enter coordinates of point 2 (x y): 0 0
Point (0, 0) is inside the triangle.
Enter coordinates of point 3 (x y): 4 3
Point (4, 3) is outside the triangle.
Enter coordinates of point 4 (x y): 2 3
Point (2, 3) is inside the triangle.
Enter coordinates of point 5 (x y): 2 0
Point (2, 0) is inside the triangle.
PS D:\Github\OOP\Lab2> 

```

**Висновок:** У результаті виконання цього проєкту була розроблена надійна програма, яка може визначити розташування точки відносно трикутника (всередині, на межі або поза ним), а також перевірити, чи є трикутник виродженим на основі його площі (використовуючи векторний добуток). Програма здатна ефективно обробляти кілька точок, надаючи чіткий зворотний зв'язок для кожного тесту. Метод векторного добутку виявився простим і потужним інструментом для вирішення геометричних задач, пов'язаних із визначенням приналежності точки трикутнику. Модульна структура коду з окремими заголовковими та вихідними файлами забезпечує легкість підтримки та можливість подальшого розширення.