|  |  |
| --- | --- |
| Học trực tuyến CNTT, học lập trình từ cơ bản đến nâng cao |  |

**IOT301 – LẬP TRÌNH C CƠ BẢN**

**THUYẾT MINH ASSIGNMENT 1**

**LẬP TRÌNH GIẢI HÌNH TAM GIÁC**

|  |  |
| --- | --- |
| Tên học viên:  Mã học viên:  Ngày báo cáo: | Thái Tuấn Hiệp  FX49111  19/10/2024 |

**HÀ NỘI, 10/2022**

**MỤC LỤC**

[1. Danh sách thư viện 1](#_heading=h.30j0zll)

[2. Danh sách các biến toàn cục và các macro 1](#_heading=h.1fob9te)

[3. Danh sách các hàm 1](#_heading=h.3znysh7)

[4. Hàm kiemtra\_tamgiac()](#_heading=h.2et92p0) 5

[5.](#_heading=h.3dy6vkm) Hàm calculateDistance 5

6. Hàm radToDegree 5

7. Hàm goccanh\_tamgiac 6

8. Hàm xet\_tamgiac 6

9. Hàm dientich\_tamgiac 6

10. Hàm duongcao\_tamgiac 7

11. Hàm trungtuyen\_tamgiac 7

12. Hàm tam\_tamgiac 7

13. Luồng xử lý code 8

**DANH MỤC HÌNH VẼ, BẢNG BIỂU**

*[Bảng 1: Danh sách thư viện](#DanhSachThuVien)*

*[Bảng 2: Danh sách các hàm](#DanhSachCacHam)*

*[Hình 1: Công thức kiểm tra tam giác cho hàm kiemtra\_tamgiac](#_Hàm kiemtra_tamgiac())*

*[Hình 2: Công thức tính cos 1 góc trong tam giác cho hàm goccanh\_tamgiac](#CongThucTinhCos1GocTrongTamGiac)*

*[Hình 3: Công thức diện tích tam giác Heron cho hàm dientich\_tamgiac](#CongThucTinhDienTichTamGiacHeron)*

*[Hình 4: Công thức tính độ dài trung tuyến cho hàm trungtuyen\_tamgiac](#CongThucTinhDoDaiTrungTuyen)*

*[Hình 5: Công thức tọa độ trọng tâm tam giác Oxy cho hàm tam\_tamgiac](#CongThucTinhTrongTamTamGiacOxy)*

**NỘI DUNG THUYẾT MINH**

# Danh sách thư viện

|  |  |  |
| --- | --- | --- |
| **STT** | **Thư viện** | **Mô tả** |
| 1 | stdio.h | Thư viện có các hàm nhập xuất  + printf: In dữ liệu ra màn hình  + scanf: Nhập liệu |
| 2 | stdbool.h | Thư viện định nghĩa kiểu dữ liệu logic  + true: 1  + false: 0 |
| 3 | math.h | Thư viện toán học  + cos: Tính cos của góc  +... |
| 4 | stdlib.h | Hàm abs tính trị tuyệt đối |

# Danh sách các biến toàn cục và các macro

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến toàn cục và macro** | **Mô tả** |
|  |  |  |
|  |  |  |
|  |  |  |

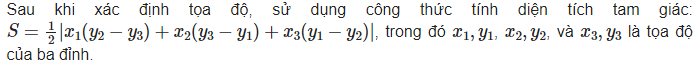
# Danh sách các hàm

|  |  |  |
| --- | --- | --- |
| **STT** | **Danh sách hàm** | **Mô tả** |
| 1 | int main()  {  float Ax, Ay, Bx, By, Cx, Cy;  printf("\nNhap toa do diem A: \n");  printf("Ax: ");  scanf("%f", &Ax);  printf("\nAy: ");  scanf("%f", &Ay);  printf("\nNhap toa do diem B: \n");  printf("Bx: ");  scanf("%f", &Bx);  printf("\nBy: ");  scanf("%f", &By);  printf("\nNhap toa do diem C: \n");  printf("Cx: ");  scanf("%f", &Cx);  printf("\nCy: ");  scanf("%f", &Cy);  if (kiemtra\_tamgiac(Ax, Ay, Bx, By, Cx, Cy)) {  printf("Toa do 3 diem nhap vao tao thanh mot tam giac.");  printf("\n");  goccanh\_tamgiac(Ax, Ay, Bx, By, Cx, Cy);  tam\_tamgiac(Ax, Ay, Bx, By, Cx, Cy);  } else {  printf("Toa do 3 diem nhap vao khong tao thanh mot tam giac.");  }  return 0;  } | Hàm chính của chương trình |
| 2 | bool kiemtra\_tamgiac(float Ax, float Ay, float Bx, float By, float Cx, float Cy) {  int area = abs(Ax\*(By - Cy) + Bx\*(Cy - Ay) + Cx\*(Ay - By));  return area != 0;  } | Kiểm tra tọa độ 3 điểm (Ax, Ay), (Bx, By) và (Cx, Cy) có tạo thành tam giác hay không |
| 3 | double calculateDistance(float x1, float y1, float x2, float y2) {  return sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));  } | Tính độ dài từ 2 điểm trên Oxy |
| 4 | double radToDegree(double radian) {  return radian \* (180.0 / M\_PI);  } | Chuyển radian sang độ để tính góc trong tam giác |
| 5 | void goccanh\_tamgiac(float Ax, float Ay, float Bx, float By, float Cx, float Cy) {  // Tinh do dai cac canh  double a = calculateDistance(Bx, By, Cx, Cy);  double b = calculateDistance(Ax, Ay, Cx, Cy);  double c = calculateDistance(Ax, Ay, Bx, By);  printf("\n");  printf("Chieu dai canh AB: %.2f\n", c);  printf("Chieu dai canh BC: %.2f\n", a);  printf("Chieu dai canh CA: %.2f\n", b);  // Su dung dinh ly cosin tinh cac goc  double alpha = acos((b\*b + c\*c - a\*a) / (2\*b\*c));  double beta = acos((a\*a + c\*c - b\*b) / (2\*a\*c));  double gamma = acos((a\*a + b\*b - c\*c) / (2\*a\*b));  // Chuyen radian sang do  alpha = radToDegree(alpha);  beta = radToDegree(beta);  gamma = radToDegree(gamma);  // In ra ket qua  printf("\n");  printf("Goc tai diem A: %.2f \n", alpha);  printf("Goc tai diem B: %.2f \n", beta);  printf("Goc tai diem C: %.2f \n", gamma);  // Xet tam giac  xet\_tamgiac(c, a, b, alpha, beta, gamma);  // Dien tich tam giac  float area = dientich\_tamgiac(c, a, b);  printf("\nDien tich tam giac ABC la: %.2f", area);  // Duong cao  duongcao\_tamgiac(area, c, a, b);  trungtuyen\_tamgiac(c, a, b);  } | Hàm tính độ dài cạnh và số độ các góc |
| 6 | float dientich\_tamgiac(float AB, float BC, float CA) {  float p = (AB + BC + CA) / 2;  return sqrt(p \* (p - AB) \* (p - BC) \* (p - CA));  } | Tính diện tích tam giác dựa trên độ dài 3 cạnh |
| 7 | void duongcao\_tamgiac(float dientich, float AB, float BC, float CA) {  float hA = 2 \* dientich / BC;  float hB = 2 \* dientich / CA;  float hC = 2 \* dientich / AB;  printf("\nDo dai duong cao tu dinh A: %f", hA);  printf("\nDo dai duong cao tu dinh B: %f", hB);  printf("\nDo dai duong cao tu dinh C: %f", hC);  } | Tính độ dài 3 đường cao tam giác dựa vào diện tích và độ dài các cạnh |
| 8 | void trungtuyen\_tamgiac(float AB, float BC, float CA) {  float m\_A = 0.5 \* sqrt(2 \* (AB \* AB + CA \* CA) - BC \* BC);  float m\_B = 0.5 \* sqrt(2 \* (AB \* AB + BC \* BC) - CA \* CA);  float m\_C = 0.5 \* sqrt(2 \* (BC \* BC + CA \* CA) - AB \* AB);  printf("\nDo dai trung tuyen tu dinh A: %.2f\n", m\_A);  printf("Do dai trung tuyen tu dinh B: %.2f\n", m\_B);  printf("Do dai trung tuyen tu dinh C: %.2f\n", m\_C);  } | Tính độ dài 3 đường trung tuyến trong tam giác dựa vào độ dài 3 cạnh |
| 9 | void tam\_tamgiac(float Ax, float Ay, float Bx, float By, float Cx, float Cy) {  float xG = (Ax + Bx + Cx) / 3;  float yG = (Ay + By + Cy) / 3;  printf("\nToa do trong tam: [%.2f, %.2f]\n", xG, yG);  } | Tính trọng tâm của tam giác dựa vào tọa độ 3 điểm trên Oxy |

# Hàm kiemtra\_tamgiac()

1. *Mô tả*

Hàm kiemtra\_tamgiac() thực hiện việc tính diện tích tam giác tọa độ xy, nếu nó khác không thì 3 điểm tạo nên 1 tam giác

  
Tham khảo: https://rdsic.edu.vn/blog/toan/huong-dan-tinh-dien-tich-tam-giac-trong-he-toa-do-oxy-vi-cb.html

1. *Tham số truyền vào*

Tọa độ 3 điểm A(Ax, Ay); B(Bx, By); C(Cx,Cy)

Trong đó Ax, Ay, Bx, By, Cx, Cy là số thực(float)

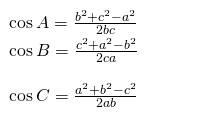
1. *Giá trị trả về*

Kiểu dữ liệu trả về là bool. Trong đó

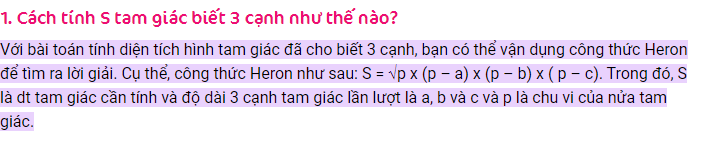
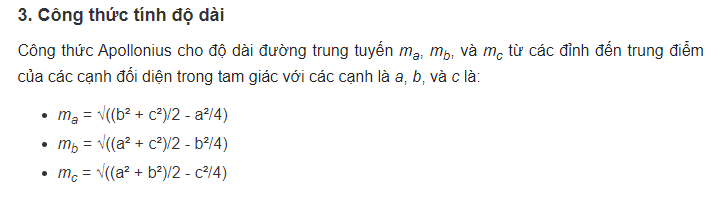
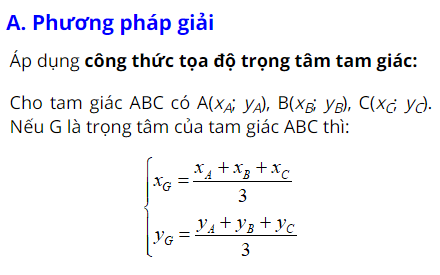
+ true : Là tam giác

+ false: Không là tam giác

# Hàm calculateDistance()

1. Mô tả: tính độ dài các cạnh dựa trên tọa độ 2 điểm ở Oxy  
   Sử dụng công thức tính cạnh huyền với khoảng cách x và khoảng cách y là 2 cạnh góc vuông.
2. Tham số truyền vào:  
   Tọa độ x1, y1, x2, y2 là số thực(float) của 2 điểm cần tính khoảng cách.
3. Gía trị trả về   
   double độ dài cạnh cần tính
4. **Hàm radToDegree()**a. Mô tả: Chuyển từ radian sang độ để tính góc các cạnh  
   b. Tham số truyền vào: số radian của góc đó (double)  
   c. Gía trị trả về: double (độ của góc)
5. **Hàm goccanh\_tamgiac**a. Mô tả:   
   Tính độ dài các cạnh, gọi đến hàm calculateDistance.  
   Tính cos các góc bằng định lý Cosin từ độ dài 3 cạnh đã biết  
    <https://thapsang.vn/cach-van-dung-dinh-ly-cosin-trong-tam-giac>  
     
   Tính radian của góc bằng cách arccos của góc dựa trên cos, dùng hàm acos của math.h (truyền vào cos của góc vừa tính phía trên)  
   Sử dụng hàm radToDegree để tính được từng góc dựa trên radian vừa tính phía tr  
   b. Tham số truyền vào:  
   Tọa độ 3 điểm A(Ax, Ay); B(Bx, By); C(Cx,Cy)

Trong đó Ax, Ay, Bx, By, Cx, Cy là số thực(float)

1. **Hàm xet\_tamgiac()**a. Mô tả:  
   Dựa vào độ dài các cạnh và góc để trả về thông tin tam giác vuông, cân hay tù ở đâu  
   b. Tham số truyền vào:  
   Độ dài các cạnh và số đo các góc float
2. **Hàm dientich\_tamgiac()**a. Mô tả:  
   Dùng công thức Heron tính diện tích tam giác từ độ dài 3 cạnh:  
   <https://sakuramontessori.edu.vn/cong-thuc-tinh-dien-tich-tam-giac/#:~:text=V%E1%BB%9Bi%20b%C3%A0i%20to%C3%A1n%20t%C3%ADnh%20di%E1%BB%87n,vi%20c%E1%BB%A7a%20n%E1%BB%ADa%20tam%20gi%C3%A1c.>  
   b. Tham số truyền vào  
   Độ dài 3 cạnh float  
   c. Gía trị trả về  
   float diện tích tam giác
3. **Hàm duongcao\_tamgiac()**a. Mô tả:   
   Tính đường cao tam giác bằng 2 lần diện tích chia cho độ dài cạnh đối diện  
   b. Tham số truyền vào:  
   diện tích và độ dài 3 cạnh float
4. **Hàm trungtuyen\_tamgiac**a. Mô tả:  
   https://rdsic.edu.vn/blog/toan/tinh-duong-trung-tuyen-trong-tam-giac-mot-phuong-phap-hoan-hao-de-tinh-toan-vi-cb.html  
     
   b. Tham số truyền vào:  
   độ dài 3 cạnh float
5. **Hàm tam\_tamgiac()**a. Mô tả: trọng tâm tam giác từ tọa độ 3 điểm trong Oxy  
   <https://vietjack.com/toan-lop-10/cach-tim-toa-do-cua-trong-tam-tam-giac-cuc-hay.jsp>  
   b. Tham số truyền vào:  
   Tọa độ 3 điểm A(Ax, Ay); B(Bx, By); C(Cx,Cy)

Trong đó Ax, Ay, Bx, By, Cx, Cy là số thực(float)

# Luồng xử lý code

*(Nếu vẽ được sơ đồ thuật toán sẽ được đánh giá cao hơn)*

Các bước thực xử lý:

*Bước 1:* Nhập tọa độ 3 điểm của tam giác và lưu vào biến toàn cục toado[6];

Trong đó:

Ax = toado[0];

Bx = toado[1];

Cx = toado[2];

…

*Bước 2:* Kiểm tra 3 điểm nhập vào có phải tam giác hay không?

Bước 3: Nếu là tam giác, gọi hàm goccanh\_tamgiac để tính độ dài 3 cạnh và số đo các góc.  
Bước 4: Trong hàm goccanh\_tamgiac, lấy kết quả các cạnh và các góc gọi hàm xet\_tamgiac để kết luận tam giác có cân, tù hay vuông không.

Bước 5: Trong hàm goccanh\_tamgiac, tiếp tục gọi hàm dientich\_tamgiac từ độ dài 3 cạnh.

Bước 6: Trong hàm goccanh\_tamgiac, tiếp tục gọi hàm duongcao\_tamgiac từ diện tích và độ dài 3 cạnh

Bước 7: Trong hàm goccanh\_tamgiac, tiếp tục gọi hàm trungtuyen\_tamgiac từ độ dài 3 cạnh

Bước 8: Vẫn ở trong trường hợp nếu là tam giác, gọi hàm tam\_tamgiac dựa trên tọa độ 3 điểm.