|  |
| --- |
| 삼각형의 면적 구하기 |
| ◆ 소스 코드 |
| #include<stdio.h>  #include<math.h>  //2차원 벡터에 대한 구조체  typedef struct \_vec2 { double x; double y; } vec2;  //두 점의 거리를 구하는 함수  double dist(vec2 p1, vec2 p2) {  return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y, 2));  }  //삼각형을 검증하는 함수  int is\_triangle(vec2 p1, vec2 p2, vec2 p3) {  double s1 = dist(p1, p2), s2 = dist(p2, p3), s3 = dist(p1, p3);  if (s1 + s2 <= s3) return 0;  if (s1 + s3 <= s2) return 0;  if (s2 + s3 <= s1) return 0;  return 1;  }  //삼각형의 넓이를 구하는 함수  double triangle\_area(vec2 p1, vec2 p2, vec2 p3) {  double size, s1 = dist(p1, p2), s2 = dist(p2, p3), s3 = dist(p1, p3);  double s = (s1 + s2 + s3) / 2;  size = sqrt(s\*(s - s1)\*(s - s2)\*(s - s3));  return size;  }  //점과 점 사이의 거리를 출력해주는 함수  void prt\_dist(vec2 p1, vec2 p2) {  printf("(%lf, %lf)와 (%lf, %lf)사이의 거리는 %lf입니다.\n", p1.x, p1.y, p2.x, p2.y, dist(p1, p2));  }  //다음 페이지에 계속  int main(void) {  vec2 point\_1, point\_2, point\_3;  printf("첫번째 점의 좌표를 입력해주세요. 콤마(,)로 구분합니다.\n예) 15, 23\n");  scanf("%lf, %lf", &point\_1.x, &point\_1.y);  printf("두번째 점의 좌표를 입력해주세요.\n");  scanf("%lf, %lf", &point\_2.x, &point\_2.y);  printf("마지막 점의 좌표를 입력해주세요.\n");  scanf("%lf, %lf", &point\_3.x, &point\_3.y);  //세 변의 길이 출력  prt\_dist(point\_1, point\_2);  prt\_dist(point\_1, point\_3);  prt\_dist(point\_2, point\_3);  //얘네 삼각형 맞나요?  printf("삼각형이 ");  printf(is\_triangle(point\_1, point\_2, point\_3) ? "맞습니다.\n" : "아닙니다.\n");  //그래서 삼각형의 넓이는요?  if(is\_triangle(point\_1, point\_2, point\_3))//삼각형이 맞을 때에만 출력합니다.  printf("이 삼각형의 넓이는 %lf입니다.\n", triangle\_area(point\_1, point\_2, point\_3));  return 0;  } |

◆ 설명

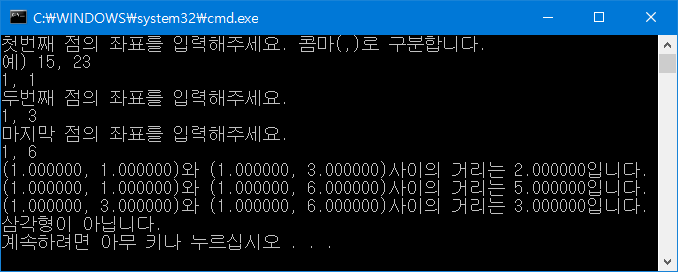
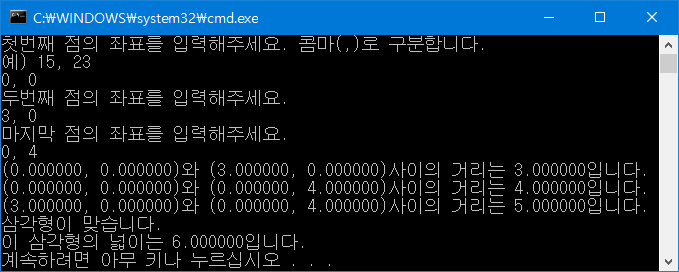
~~사실 주석으로 대부분 적어서, 그리고 애초에 설명할 것이 거의 없지만 해보자면~~

x와 y를 전부 써주는 대신 struct로 x와 y축을 가지는 vec2 자료형을 만듭니다.

scanf로 각 점의 x값과 y값을 입력 받고, dist함수에서 값을 받아서 출력하는 함수로 출력하고,

is\_triangle함수로 삼각형을 검증 한 후에 참이면 삼각형의 넓이를 출력합니다.

◆ 실행 화면



모든 코드는 https://github.com/MU-Software/hs\_c/tree/master/1011에서 확인하실 수 있습니다.

또는 아래의 QR코드로 보실 수 있습니다.

