```cpp
#include <iostream>
#include <string>
#include <math.h>
#include "Polygon.hpp"
using namespace std;

// Costruttori
Polygon::Polygon(int m, int* vett){
    n = m;
    self_edges = vett;
}
Polygon::Polygon(){
    n = 0;
    self_edges = NULL;
}

// Distruttore
Polygon::~Polygon(){ }

// Costruttore di copia
Polygon::Polygon(const Polygon &p){
    n = p.n;
    self_edges = p.self_edges;
}

// Metodo setter per il vettore statico
Edge* Polygon::edges = NULL;
void Polygon::set_edges(Edge* e){
    edges = e;
}

// Metodi getter
int Polygon::get_n(){
    return n;
}
int* Polygon::get_self_edges(){
    return self_edges;
}
Edge Polygon::get_edge(int index){
    return edges[index];
}
Point2D Polygon::get_edge_point1(int index){
    return edges[self_edges[index]].get_p1();
}
Point2D Polygon::get_edge_point2(int index){
    return edges[self_edges[index]].get_p2();
}
float Polygon::get_edge_point1_x(int index){
    return edges[self_edges[index]].get_p1_x();
}
float Polygon::get_edge_point1_y(int index){
    return edges[self_edges[index]].get_p1_y();
}
float Polygon::get_edge_point2_x(int index){
    return edges[self_edges[index]].get_p2_x();
}
float Polygon::get_edge_point2_y(int index){
    return edges[self_edges[index]].get_p2_y();
}

// Metodi setter
void Polygon::set_n(int N){
    n = N;
}
void Polygon::set_self_edges(int* vett){
    self_edges = vett;
```

```
}
// Metodi per il calcolo del perimetro e dell'area del poligono
float Polygon::perimetro(){
    float perim = 0;
    for (int i = 0; i < n; i++){
        perim += edges[self_edges[i]].lunghezza();
    }
    return perim;
}
float Polygon::area(){
    float ar = 0, a , b, c, p;
    Edge lato1, lato2, lato3;

    lato1.set_p1_i(self_edges[0]);
    lato3.set_p1_i(self_edges[0]);

    for (int i = 1; i < n - 1; i++){
        lato1.set_p2_i(self_edges[i]);
        lato2.set_p1_i(self_edges[i]);
        lato2.set_p2_i(self_edges[i+1]);
        lato3.set_p2_i(self_edges[i+1]);
        a = lato1.lunghezza();
        b = lato2.lunghezza();
        c = lato3.lunghezza();
        p = (a + b + c)/2.0;
        ar += sqrt(p*(p-a)*(p-b)*(p-c));
    }
    return ar;
}
```