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

// Costruttori
Edge::Edge(int p1_index, int p2_index, int index){
    p1_i = p1_index;
    p2_i = p2_index;
    i = index;
}
Edge::Edge(){
    p1_i = -1;
    p2_i = -1;
    i = -1;
}

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

// Costruttore di copia
Edge::Edge(const Edge &e){
    p1_i = e.p1_i;
    p2_i = e.p2_i;
    i = e.i;
}

// Metodo setter per il vettore statico
Point2D* Edge::points = NULL;
void Edge::set_points(Point2D* p){
    points = p;
}

// Metodi getter
int Edge::get_p1_i(){
    return p1_i;
}
int Edge::get_p2_i(){
    return p2_i;
}
int Edge:: get_i(){
    return i;
}
Point2D Edge::get_p1(){
    return points[p1_i];
}
Point2D Edge::get_p2(){
    return points[p2_i];
}
float Edge::get_p1_x(){
    return points[p1_i].get_x();
}
float Edge::get_p1_y(){
    return points[p1_i].get_y();
}
float Edge::get_p2_x(){
    return points[p2_i].get_x();
}
float Edge::get_p2_y(){
    return points[p2_i].get_y();
}

// Metodi setter
void Edge::set_p1_i(int j){
    p1_i = j;
}
void Edge::set_p2_i(int j){
    p2_i = j;
}
void Edge::set_i(int j){
    i = j;
}

// Metodo per il calcolo della lunghezza del lato
float Edge::lunghezza(){
    return sqrt((get_p1_x()-get_p2_x())*(get_p1_x()-get_p2_x()) + (get_p1_y()-get_p2_y())*(get_p1_y()-get_p2_y()));
}
```