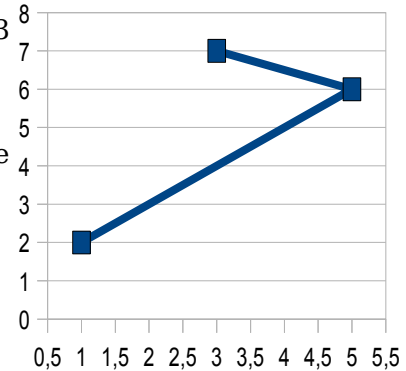
	Liceo B. Russell VIA 4 NOVEMBRE 35, 38023 CLES	Indirizzo: SCIENTIFICO PNI	CLASSE 4 C
		Data: 18 novembre 2011	prof. Armani
		Disciplina: INFORMATICA	ALUNNO
		Prova: INFORMATICA

NB: i programmi vanno **inizialmente salvati e testati nella cartella personale** dell'utente **Tema1Axx**. La versione finale del programma dovrà **poi essere copiata nella cartella condivisa in scrittura** con nome **cognome_nome_classe.cpp** .

Sei un programmatore impegnato nello sviluppo di un programma di grafica. Devi riscrivere in modo semplice ed efficiente la parte di codice che serve per disegnare e manipolare degli elementi "polilinea". Ognuno di questi elementi sarà individuato da una serie di punti P_i di coordinate (x_i, y_i) , e quindi dai relativi segmenti che dovranno essere disegnati a video.

Es: polilinea rappresentato a fianco, costituito da un totale di 3 punti di coordinate $(1,2)$, $(5,6)$ e $(3,7)$.



Dovrai quindi creare una classe polilinea come dal seguente prototipo:

```
class polilinea{
  int n; //numero di punti
  float *px; //vettore con la coordinata x dei punti della polilinea
  float *py; //vettore con la coordinata y dei punti della polilinea
}
```

Per l'esempio sopra riportato, avremo quindi che $n=3$, che px corrisponde al vettore $px=\{1,5,3\}$ e py al vettore $py=\{2,6,7\}$.

1. Crea il prototipo di tale classe ed un costruttore senza parametri che inizializzi n 0:
`polilinea();`
 (0.5 punti)
2. Aggiungere alla classe un costruttore che prenda in input le coordinate del primo punto; si assuma quindi $n=1$ e si inizializzino opportunamente px e py .
`polilinea(float y, float y);`
 (1 punto)
3. Creare l'opportuno distruttore per questa classe:
`~polilinea();`
 (1 punto)
4. Aggiungere il metodo
`void scrivi();`
 che stampi le coordinate dei punti della polilinea (usa il formato (x_1, y_1) , (x_2, y_2) etc.. Per l'esempio si avrà quindi un output simile a: $(1,2)$, $(5,6)$, $(3,7)$).
 (0.5 punti)
5. Definire l'operatore di assegnamento per tale classe:
`void operator=(const polilinea &A);`
 (2.5 punti)
6. Definire l'operatore $+$ per tale classe:
`polilinea operator+(const polilinea &A) const;`
 Tale operatore dovrà "concatenare" le due polilinee e restituirne una terza. In altre parole dovrà restituire una polilinea contenenti i punti della prima più quelli della seconda.
 (2.5 punti)
7. Scrivere un programma principale che, definite tre variabili A , $B(1,2)$ e $C(3,4)$ di tipo polilinea, calcoli $A=B+C$ e stampi A .
 (1 punto)
8. Aggiungere alla classe un metodo che calcoli la lunghezza della polilinea.
`float lunghezza() const;`
 Nel main del punto precedente far stampare la lunghezza di A .
 (1 punto)

Esempio ad uso "promemoria":

```
#include <iostream>
#include <cmath>
using namespace std;

class vettore{
    double *p;
    int n;

    public:
    vettore(int m);           //costruttore
    ~vettore();              //distruttore
    void leggi();
    void operator=(const vettore &ob); //overloadig op. di assegnamento
    vettore operator+(const vettore &ob) const; //somma di due vettori
};

void vettore::vettore(int m){ [...] }
void vettore::scrivi(){ [...] }
[... ]

int main(){
    vettore A(6),B(6),C(12);
    A.leggi();
    B.leggi();
    C=A+B;
    C.scrivi();
    return 0;
}
```