

Si supponga di voler rappresentare una mappa geografica dove sono raffigurati i punti di interesse del luogo che si sta visitando. Per ogni punto di interesse sono memorizzati il nome e le coordinate (x,y) espresse rispetto ad un riferimento cartesiano. Le coordinate sono rappresentate come numeri reali. Realizzare le seguenti operazioni che possono essere effettuate sulla Mappa:

- `Mappa m(N, nome)`
Costruttore, che crea una mappa che contiene N punti di interesse. I punti di interesse sono inizializzati prelevando i valori dal file `nome`. Ogni riga nel file contiene una stringa, che identifica il nome del punto di interesse, e due numeri reali che individuano le coordinate x e y . Si supponga che nel file siano presenti N righe che rappresentano N punti di interessi diversi;
- `m.trova(x, y)`
Operazione che trova il punto di interesse più vicino al punto (x, y) passato come argomento alla funzione. Per calcolare la vicinanza si adotti la distanza Euclidea. L'operazione restituisce il nome del punto di interesse trovato. Se nella mappa sono presenti più punti di interesse alla stessa distanza minima, si restituisca il nome di un punto di interesse a caso tra quelli a distanza minima;
- `m2 = m1`
Operatore di assegnamento per la classe `Mappa`; si tenga presente che la mappa `m1` viene assegnata ad `m2` anche se il numero di punti di interesse è diverso; una volta applicato l'operatore, le due mappe avranno lo stesso numero di punti di interesse;
- `m2 += m1`
Operatore che aggiunge ad `m2` i punti di interesse contenuti nella mappa `m1`; si supponga che i punti di interesse memorizzati in `m1` siano diversi da quelli memorizzati in `m2`;
- `~Mappa()` ;
Distruttore;
- `cout<<m;`
Operatore di uscita per il tipo `Mappa`. L'operatore stampa su video ogni punto di interesse. Per esempio, per una `Mappa` che memorizza quattro punti di interesse il formato dell'uscita è il seguente:

Nome	X	Y
D	120.3	240.5
B	200.3	400.4
A	450.2	102.3
C	10.3	103.4

Mediante il linguaggio C++, realizzare il tipo di dato astratto `Mappa`, definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esaInf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- **Per una corretta stampa dell'elaborato**, non scrivere righe di codice di lunghezza eccessiva (mantenersi entro i margini imposti dall'ambiente *Dev-C++* cioè entro la linea verticale presente alla destra della pagina);
- **Per la Consegna:**
 - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

Condizione necessaria per la correzione dell'elaborato è che le prime tre funzioni siano state implementate correttamente.