

Un insieme di punti può essere descritto dalle strutture seguenti:

```
struct punto
{
    int numCoor;
    int ident;
    float coor[M];
};

struct insieme
{
    punto pun[NUM];
    int numPun;
};
```

con **NUM** il numero massimo di punti contenuti nell'insieme e **numPun** il numero di punti effettivamente presenti nell'insieme. Ogni punto è individuato da un numero intero **numCoor** che indica il numero di coordinate del punto, da un numero intero **ident** che identifica univocamente il punto e da un vettore **coor** (al più di **M** elementi) di numeri reali **float** che memorizza i valori delle **numCoor** coordinate. Quando è creata un'istanza della struttura **insieme**, il valore di **numPun** è inizializzato a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **void build(const char nome[], insieme* ins)** che apre in lettura il file il cui nome è passato come primo argomento alla funzione, e legge due numeri interi, che indicano, rispettivamente, il numero **numCoor** di coordinate dei punti ed il numero **numPun** di punti contenuti nel file; allora, per **numPun** volte, legge un numero intero, che costituisce l'identificatore di un punto, e **numCoor** numeri reali, che costituiscono le coordinate del punto, ed inserisce il punto nell'insieme (ricordarsi di assegnare il numero **numCoor** di coordinate al campo **numCoor** del punto);
2. **int trova(const insieme* ins, punto p)** che restituisce l'identificatore del punto dell'insieme meno distante dal punto **p**; la distanza d tra il punto **p**, ed un punto generico x dell'insieme è calcolata come $d = \sum_{f=1}^F |p_f - x_f|$, dove $||$ rappresenta il valore assoluto e F è il numero di coordinate dei punti. Per calcolare il valore assoluto utilizzare la funzione *float fabs(float t)*, che restituisce il valore assoluto di t ;
3. **int maxMod(const insieme* ins)** che restituisce l'identificatore del punto la cui somma del valore assoluto delle coordinate è massima;
4. **bool elimina(insieme* ins, int ind)** che elimina dall'insieme il punto il cui identificatore è passato come secondo argomento alla funzione e ricompatta il vettore **pun**; se il punto non è presente nell'insieme viene restituito **false**; altrimenti viene restituito **true**.
5. **void ordina(insieme* ins, punto p)** che ordina i punti contenuti nell'insieme per distanze crescenti dal punto **p** passato come secondo argomento alla funzione.

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- PER SVOLGERE L'ELABORATO, 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 LO SVOLGIMENTO DELL'ELABORATO, selezionare la voce **Identifica studente** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e inserire i dati richiesti;
- Per svolgere l'elaborato, aprire il progetto *esaInf.dev* presente nel directory *c:\esame\esaInf* e scrivere le funzioni richieste nel file *compito.cpp*, già presente nel progetto;
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina);
- SE L'ELABORATO È STATO COMPILATO SENZA ERRORI, PRIMA DELLA CONSEGNA, selezionare la voce **Consegna Compito** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e premere il tasto INVIO fino a quando non sparisce la finestra che è stata attivata.