

Un insieme di punti definiti in  $\mathcal{R}^N$  può essere descritto in C++ come:

```
struct elem
{
    float punto[N];
    char classe;
};

struct insieme
{
    int numPunti;
    int dimen;
    elem el[SIZE];
};
```

con **SIZE** il numero massimo di punti contenibili nell'insieme, **N** il numero massimo di coordinate di un punto, **numPunti** il numero di elementi attualmente nell'insieme e **dimen** la dimensione dei punti attualmente nell'insieme. Il membro **classe** della struttura **elem** contiene la classe a cui appartiene il punto. Quando un insieme viene creato, i membri **numPunti** e **dimen** vengono inizializzati a 0.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(const char nome[], insieme\* ins, int num, int dim)** che legge dal file il cui nome è passato come primo argomento formale **num** elementi dell'insieme; ogni elemento è memorizzato nel file come **dim** numeri **float**, che costituiscono le coordinate di un punto, ed un carattere, che indica la classe (per esempio, un elemento composto da un punto definito in  $\mathcal{R}^2$  ed appartenente alla classe A appare nel file come **1 2 A**); la funzione restituisce **false** se il file non esiste oppure non è nel formato previsto; altrimenti restituisce **true**.
2. **void ordina(insieme\* ins, int n)** che ordina gli elementi contenuti nell'insieme per distanze crescenti rispetto all'origine. Si consiglia di utilizzare una funzione **float calcDist(const float p1[], const float p2[], int dim)** che restituisce la distanza Euclidea tra due punti di dimensione **dim**. Si ricorda che per eseguire la radice quadrata di un numero si usa la funzione **sqrt(float a)** che è dichiarata nel file **math.h**.
3. **char classifica(const insieme\* ins, const float p[])** che associa una classe al punto **p** passato come argomento. La funzione restituisce la classe a cui appartiene il punto dell'insieme meno distante dal punto **p**.
4. **bool cancella(insieme\* ins, int ind)** che, se esiste, cancella l'elemento il cui indice è **ind** e ricompatta il vettore che contiene gli elementi. La funzione restituisce **false** se l'indice non è valido; altrimenti restituisce **true**.

Si ricorda che:

- il compito deve essere svolto aprendo il progetto *esaInf.dev* nel directory *c:\esame\esaInf* e scrivendo le funzioni richieste nel file *compito.cpp*;
- per la produzione dei risultati, aprire una finestra DOS, posizionarsi nel directory *c:\esame\esaInf*, digitare il comando **esaInf.exe > output.txt** e premere due volte il tasto INVIO;
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++.