

### Esercizio n.1

Scrivere una funzione *trasposta* che prende in ingresso una matrice di interi di dimensione  $N \times M$ , e restituisce una nuova matrice  $M \times N$  che è la trasposta della matrice passata alla funzione.

Scrivere un main che:

1. Legge da tastiera le due dimensioni  $N$  ed  $M$  della matrice;
2. Crea una matrice di interi di dimensioni  $N \times M$  (in memoria dinamica);
3. Legge  $N \times M$  numeri interi e li salva nella matrice (salvataggio per riga);
4. Stampa a video la matrice;
5. Chiama la funzione passando la matrice creata;
6. Stampa a video la trasposta della matrice;
7. Distrugge le due matrici.

### Esercizio n.2

Scrivere una funzione che prende in ingresso due array di interi **v1** e **v2**, rispettivamente di dimensione  $N$  e  $M$ . Supporre che i due array siano ordinati in maniera crescente. La funzione deve creare e restituire un nuovo array **v** che contiene tutti gli elementi di  $v1$  e di  $v2$ , e deve essere anch'esso ordinato in maniera crescente. Inoltre, i due array  $v1$  e  $v2$  possono avere duplicati ed essere di dimensioni diverse

Scrivere un main:

1. Legge da tastiera le due dimensioni  $N$  ed  $M$  degli array;
2. Crea due array di interi, uno di dimensione  $N$  e uno di dimensione  $M$  (mem. dinamica);
3. Legge da tastiera  $N+M$  numeri interi e li salva nei due array (inserirli in maniera ordinata crescente);
4. Stampa a video i due array;
5. Chiama la funzione passando i due array creati;
6. Stampa a video il nuovo array;
7. Distrugge tutti gli array.

### Esercizio n.3

Sia data la seguente struttura:

```
struct punto{
    double x;
    double y;
};
```

Scrivere una funzione che prende in ingresso due array di reali **v1** e **v2**, rispettivamente di dimensione N e M.

La funzione deve creare e restituire un **array di punti** di dimensione pari al massimo tra N e M.

L'elemento i-esimo dell'array restituito deve avere come campo x l'elemento i-esimo di v1 e come campo y l'elemento i-esimo di v2.

Se l'i-esimo elemento di v1 (o di v2) non esiste, assumere per il campo associato il valore 0.

Scrivere inoltre un main:

1. Legge da tastiera le due dimensioni N ed M degli array;
2. Crea due array di interi, uno di dimensione N e uno di dimensione M (mem. dinamica);
3. Legge da tastiera N+M numeri interi e li salva nei due array (inserirli in maniera ordinata crescente);
4. Stampa a video i due array;
5. Chiama la funzione passando i due array creati;
6. Stampa a video il nuovo array;
7. Distrugge tutti gli array.

*Hint:* stampare una variabile di tipo `punto` nella forma `<x, y>`