



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria e
Scienza dell'Informazione
DISI - Trento

Programmazione 1

16 - Esercitazione

Giovanna Varni
giovanna.varni@unitn.it

Anno Accademico 2023/2024

Nelle puntate precedenti

- **Allocazione dinamica della memoria**

```
char * carattere = new char;  
int * array = new int[10];
```

- L'operatore `new` ci permette di allocare della memoria nello heap. Questo ci da maggiore flessibilità rispetto a programmi in cui la memoria viene allocata staticamente.

Nelle puntate precedenti

- **Deallocazione della memoria**

```
delete carattere;  
delete[] array;
```

- L'operatore `delete` ci permette di deallocare la memoria precedentemente riservata con `new`. Questo passaggio è importante poiché previene la creazione di `memory leak`;

Nelle puntate precedenti

- Valgrind (<https://valgrind.org/>)
 - E' un programma open-source che vi permette di eseguire dei controlli sull'utilizzo di memoria di programmi in C++. E' in grado di rilevare automaticamente memory leaks o altri problemi causati da un'errata gestione della memoria;

```
g++ -Wall vostro_codice.cc -o eseguibile  
valgrind ./eseguibile
```

1 - Random Array

```
int * generate_numbers (int N);
```

Scrivere un programma che, dato in input dall'utente da riga di comando un numero intero N, ritorni a video una lista di N numeri interi generati randomicamente nell'intervallo 0-99. Questi numeri devono essere salvati in un array. Questo array deve essere generato **dinamicamente** da una funzione con la firma uguale a quella mostrata sopra.

2 - Random Matrix

```
int ** generate_matrix(int R, int C);
```

Scrivere un programma che, dati in input dall'utente da riga di comando due numeri interi R e C, crei **dinamicamente** una matrice con R righe ed C colonne. Popolate poi questa matrice con dei numeri random.

Infine, stampate questa matrice a video.

3 - Temperatura

Scrivere un programma che, dato un file in input da linea di comando contenente diverse temperature. Il file è costituito da una sola colonna e le temperature sono rappresentate come numeri reali. Il programma salvi queste temperature in un array **dinamico** e stampi a video le prime N temperature più alte. Il valore N viene specificato dall'utente da linea di comando. Il programma accetta anche un intero M che indica quante temperature al massimo sono presenti nel file.

27.5

34.3

13.2

. . .

4 - Concatena

Si scriva la funzione *append* che concateni due array di numeri interi ordinati in senso crescente. La funzione riceve come parametri i puntatori a due array di numeri interi *pa* e *pb* e le loro dimensioni *na* e *nb* (due numeri interi) e restituisce come valore di ritorno il puntatore a un array di numeri interi. Si suppone che gli array puntati da *pa* e *pb* siano già ordinati in senso crescente. La funzione allocherà dinamicamente un array di $(na + nb)$ elementi e vi copierà gli elementi degli array puntati da *pa* e *pb* in modo tale che l'array risultante sia a sua volta ordinato in senso crescente. La funzione restituirà infine il puntatore all'array risultante. Sarà responsabilità del programma chiamante deallocare la memoria allocata dalla funzione.

Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione. Il programma chiederà all'utente di immettere da tastiera le dimensioni *nx* e *ny* di due array di numeri interi, allocherà dinamicamente gli array e chiederà all'utente di inserire i valori per entrambi gli array. Il programma utilizzerà quindi l'algoritmo BubbleSort per ordinare entrambi gli array, chiamerà la funzione *append* e stamperà a video l'array concatenato risultante.