

Esercitazione 26/02

Puntatori, array e funzioni, aritmetica dei puntatori

Riferimento

[Esercizio C006] – DIFICILE... Si chiede di scrivere un programma in C++,

- Il programma dovrà richiedere all'utente di inserire tre numeri interi. L'obiettivo del programma è determinare quale tra i numeri inseriti è il massimo. A tal fine, utilizza il riferimento ``max`` e una condizione con `(? :)`.
- Dopo aver trovato il massimo, il programma ne incrementa il valore utilizzando il riferimento.
- Alla fine, il programma stampa i valori delle tre variabili per verificare se l'incremento è stato effettuato correttamente.
- Perché non si può usare una sequenza delle istruzioni «if» al posto di ?

espressione_test ? azione_true : azione_false;

```
#include <iostream>
using namespace std;
```

```
int main() {
    int valore;
    cin >>valore;
    bool positivo;
    if(valore >= 0) {
        positivo = true;
    }
    else {
        positivo = false;
    }
    cout<<positivo;
    return 0;
}
```

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int valore;
    cin >>valore;
    bool positivo;

    positivo = (valore >= 0) ? true : false;

    cout<<positivo;

    return 0;
}
```

Soluzione C006

```
int main() {  
    int a, b, c;  
    cout << "Inserisci tre numeri: ";  
    cin >> a >> b >> c;  
    cout << a << b << c;  
  
    int& max = (a > b) ? (a > c ? a : c) : (b > c ? b : c);  
  
    cout << "Il numero massimo è: " << max << endl;  
    max++;  
    cout << a << b << c;  
    return 0;  
}
```

Puntatori

Un po' di teoria [prova intermedia del 4 luglio 2018]

Si scriva il valore assunto dalle variabili *s* e *b* dopo l'esecuzione delle seguenti righe di codice:

```
2 1. char b = 'R'; char s = 'S';  
   2. char& q = b;  
   3. char* ps = &s;  
   4. (*ps)++;  
   5. b = 'A' + (*ps - q);
```

Cerchiamo di fare il disegno... la soluzione è nelle slide successive

Puntatori

Un po' di teoria [prova intermedia del 4 luglio 2018]

```
2 1. char b = 'R'; char s = 'S';  
   2. char& q = b;  
   3. char* ps = &s;  
   4. (*ps)++;  
   5. b = 'A' + (*ps - q);
```

- 1) L'istruzione alla riga 2 definisce q come un riferimento a b . Il nome q è, quindi, **un secondo nome per denotare la variabile b** .
- 2) La riga successiva definisce ps come un puntatore a carattere e lo inizializza in modo tale che punti alla variabile s .
- 3) L'istruzione alla riga 4 incrementa di uno il valore della variabile puntata da ps (cioè s). Dopo l'esecuzione di tale istruzione s vale quindi 'T'.
- 4) Infine, l'ultima istruzione assegna alla variabile b il valore di 'A' incrementato della differenza tra il valore della variabile puntata da ps (cioè s che vale 'T') e il valore di q (che è un altro nome per la variabile b il cui valore è 'R'). Tale differenza vale 2 e pertanto la variabile b assume il valore 'C'.

Puntatori

[Esercizio C013] – Si scriva un seguente programma C++:

- Si scriva la **funzione C++** *massimo* che riceva come parametri i puntatori a tre numeri interi a , b e c e **restituisca come valore di ritorno il puntatore al più grande** dei tre numeri (un puntatore a numero intero).
- Nel caso in cui vi siano due numeri uguali, entrambi più grandi del terzo numero, la funzione restituisce il puntatore al primo dei due.
- Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione.
- Il programma chiederà all'utente di immettere da tastiera tre numeri interi, chiamerà la funzione *massimo* e stamperà a video il valore puntato dal puntatore che la funzione restituisce come valore di ritorno.
 - *Esempio:* se a vale 3, b vale 10 e c vale 5, la funzione restituirà come valore di ritorno il puntatore a b e il programma stamperà a video 10.

Aritmetica dei puntatori

[Esercizio C011] – Aritmetica dei puntatori e le stringhe

- Dichiarare due stringhe di uguale dimensione (p.e., 20 caratteri).
- Acquisire in una stringa una sequenza di caratteri (massimo 20),
- Copiare, usando i puntatori, tutti i caratteri dalla stringa acquisita all'altra, accedendo in modo indiretto a ciascuna delle locazioni delle due stringhe.
- Stampare il risultato.

Aritmetica dei puntatori

[Esercizio C014] – Aritmetica dei puntatori e l'array dei numeri reali

- Si scriva la funzione C++ *prodotto_scalare* che riceva come parametri il puntatore *px* al primo elemento di un array di numeri reali, il puntatore *py* al primo elemento di un array di numeri reali e la dimensione comune *n* dei due array (un numero intero).
- Utilizzando l'aritmetica dei puntatori, la funzione dovrà scandire i due array e calcolarne il prodotto scalare, restituito come valore di ritorno (un numero reale).
- Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione. Il programma chiederà all'utente di immettere da tastiera i valori per i due array, chiamerà la funzione *prodotto_scalare* e ne stamperà a video il valore di ritorno.
- *Esempio*: se l'array puntato da *px* vale {1.0, 3.0, 2.5, 0.0, 1.2} e l'array puntato da *py* vale {2.0, 1.0, 2.0, 3.8, 10.0} (si ha quindi $n = 5$),
- la funzione restituisce il valore del prodotto scalare dei due array, ovvero: $1.0 \times 2.0 + 3.0 \times 1.0 + 2.5 \times 2 + 0.0 \times 3.8 + 1.2 \times 10.0 = 22.0$.

Puntatori

[Esercizio C012] - Passaggio di puntatori a funzione:

- Si scriva la funzione C++ *swap* che riceva come parametri i puntatori a tre numeri interi *a*, *b* e *c* e ne ruoti i valori, ovvero: a *b* viene assegnato il valore di *a*, a *c* viene assegnato il valore di *b* e ad *a* viene assegnato il valore di *c*.
- Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione.
- Il programma chiederà all'utente di immettere da tastiera tre numeri interi, chiamerà la funzione *swap* per ruotarne i valori e stamperà a video il risultato.
 - *Esempio:* se *a* vale 3, *b* vale 5 e *c* vale 10, dopo la chiamata alla funzione si avrà che *a* varrà 10, *b* varrà 3 e *c* varrà 5.

Puntatori

[Esercizio C012B] - swap .. continuazione

- Se non si volessero utilizzare i puntatori come si potrebbe re-implementare la funzione?
- Per verificarlo, scrivete una funzione *swap_2* che scambi i valori senza usare i puntatori e utilizzatela nel programma sviluppato.
- Confrontate il codice dell'implementazione con i puntatori e di quella senza i puntatori. In che cosa si differenziano?

Aritmetica dei puntatori

[Esercizio C015] – Aritmetica dei puntatori, funzioni e le stringhe

- Si scriva la funzione C++ *cerca* che riceva come parametri il puntatore al primo elemento di una stringa *str1* e il puntatore al primo elemento di una seconda stringa *str2*.
- Utilizzando l'aritmetica dei puntatori, la funzione dovrà scandire la stringa *str1* e restituire, come valore di ritorno, il puntatore alla prima occorrenza della sottostringa *str2* in *str1*. Nel caso in cui la sottostringa *str2* non occorra in *str1*, la funzione restituirà NULL.
- Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione. Il programma chiederà all'utente di immettere da tastiera due stringhe, chiamerà la funzione *cerca* per verificare se la seconda stringa occorre all'interno della prima e stamperà a video, in caso di esito positivo, la stringa *str1* a partire dal carattere puntato dal puntatore restituito dalla funzione *cerca* e fino al suo termine.
- In caso di esito negativo della verifica, il programma stamperà a video un opportuno messaggio.

Puntatori

[Esercizio C017] - Scrivere un programma che carichi un vettore di caratteri e conti quante consonanti e vocali sono presenti, visualizzando i risultati a video. Si utilizzino i puntatori.

- In dettaglio:
 - Scrivi una funzione che prenda come argomenti un puntatore a carattere e un intero, e carichi il vettore di lunghezza specificata con caratteri inseriti dall'utente. La funzione deve quindi stampare il vettore.
 - Scrivi una funzione conta che prenda come argomenti un puntatore a carattere e un intero, e conti il numero di vocali e consonanti presenti nel vettore. La funzione deve quindi stampare il numero di vocali e consonanti.
 - Scrivi il programma principale che controlli il funzionamento corretto delle funzioni. Il programma definisce un vettore di caratteri di lunghezza massima 100, e chiede all'utente di inserire il numero di caratteri da caricare nel vettore.

Puntatori

[Esercizio C018] – Puntatori e Matrici:

- Si scriva la funzione C++ *righe_negative* che riceva come parametri una matrice A di n righe e n colonne (n è dichiarato come una costante intera all'inizio del programma) e un array di puntatori a numeri interi b .
- L'array b è costituito anche esso dallo stesso numero costante n di elementi e la funzione lo inizializza in modo tale che tutti gli elementi siano NULL. La funzione scandirà la matrice A riga per riga ed inserirà nell'array b i puntatori alle righe per le quali la somma degli elementi risulta essere un numero negativo.
- Si scriva quindi un programma C++ per verificare il corretto funzionamento della funzione: il programma chiederà all'utente di immettere da tastiera gli elementi della matrice A, chiamerà la funzione *righe_negative* e stamperà a video le righe la cui somma degli elementi è un numero negativo.

Esempio: se la matrice A (di dimensione 3 x 3) fosse la seguente:

$$A = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 3 & -1 \\ -5 & -6 & -2 \end{bmatrix}$$

si otterrebbe l'array $b = \{\text{puntatore alla prima riga, puntatore alla terza riga, NULL}\}$ e il programma stamperebbe a video la prima e la terza riga della matrice A.