**4° lezione prog II**

**Puntatoti e riferimenti**

**Variabile puntatore**

Una **variabile puntatore** contiene un indirizzo di memoria.Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Si dice che è una variabile puntatore punta ad un certo dato perché contiene il suo indirizzo in memoria.

**Come si dichiara una variabile puntatore**

Una variabile puntatore si dichiara nel seguente modo:

**int \*contPtr;**

* **int** tipo
* **Il simbolo\*** indica che si tratta di una variabile puntatore ed è chiamato operatore di de referenziazione.

**Operatore di indirizzo**

Si indica con il simbolo &  e restituisce l'indirizzo del suo operando.

 ES **int y=5; int \*yPtr=&y;**

**In questo modo si sta definendo una variabile y  = 5 e una variabile puntatore yPtr che ha l'indirizzo della variabile y.**

Se si volesse **modificare il valore del dato puntato da Yptr** Si dovrebbe procedere nel seguente modo:  \*yPtr=20;

In questo modo Y avrà il valore 20.

Se si volesse stampare **l’indirizzo della cella di memoria**  si dovrebbe procedere nel seguente modo:

* **cout<<p; è un valore esadecimale**

Se invece si volesse **stampare il valore contenuto nella variabile y(con l'utilizzo della variabile puntatore)** si dovrebbe procedere nel seguente modo:

* **cout<<\*p);**

**Array e puntatori**

**Il nome  dell’ array è un puntatore  costante al primo elemento dell’ array pertanto  per visualizzare il primo elemento di questo (indice 0 si procede) si procede  nei seguenti modi:**

Supponendo di aver dichiarato un’ array di double di 3 elementi(da 0 a 2) {1.2, 10.7,9.8}

Immagine che contiene testo, Carattere, bianco, tipografia

Descrizione generata automaticamente

Se il puntatore punta al vettore si può utilizzare il puntatore come un array e pertanto si può  accedere ai vari elementi nel seguente modo: Immagine che contiene testo, Carattere, bianco, schermata

Descrizione generata automaticamente

**Aritmetica dei puntatori**

**Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente**

Con l'espressione **\*(ptr+x)**  il puntatore si sposta di x posizioni.Il programmatore non si deve preoccupare della quantità di byte della cella di memoria  e ne   qual'è l’indirizzo di memoria perché ci pensa il compilatore.

**Riassumendo**

**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente**

si potrebbe fare anche in un ciclo for() nel seguente modo:

**for(...){printf(“%d\t”,\*(ptr+i));}**

**Quando si utilizza l'aritmetica dei puntatori bisogna prestare attenzione ad utilizzare  l'offset in quanto si potrebbe superare la dimensione del vettore e in questo caso avremo un errore di Run Time. ES V[10] \*(V+20) (errato).**

**è buona norma che quando non si utilizza più il puntatore si pone a NULL e prima utilizzare un qualsiasi puntatore testare che non sia NULL(!=NULL)Oppure quando non  lo si inizializza e quindi non si da un indirizzo lo si pone a NULL.**

**ATTENZIONE  Supponendo di aver definito una variabile x=10; e xPtr=&x;**

**importante con \*xPtr stampo il dato puntato da ptr(10 %d) e con xPtr stampo l’indirizzo(%p)**

**Operatori consentiti per l'aritmetica dei puntatori**

**Immagine che contiene testo, Carattere, schermata, algebra

Descrizione generata automaticamente**

* **Con l'espressione ptr++/ptr+=1/ptr=ptr+1  faccio puntare il puntatore alla posizione successiva (intero e puntatore)**
* **Con l'espressione ptr–- faccio puntare il puntatore alla posizione precedente (intero e puntatore)**

**Puntatori costanti e puntatori a costanti**

Prima di definire che cos'è il puntatore a  costante definiamo Che cos'è una costante. Una costante è una variabile di un certo tipo **che non può cambiare durante il runtime del programma.**

**Puntatori a costanti**

**int num=10;**

**int\* const ptr=&num;**

**Mediante il puntatore ptr non si potrà modificare il valore puntato dal puntatore stesso** (come una costante). **Si deduce pertanto che va inizializzato contestualmente alla sua dichiarazione.**È irrilevante (in questo caso) se il dato è costante o variabile.

**Puntatori costanti**

**int num=20;**

**const int \*ptr=&num;**

Si usa quando attraversa il puntatore ptr non si vuole modificare il valore alla locazione di memoria puntata dal   puntatore.**Si deduce pertanto che va inizializzato contestualmente alla sua dichiarazione**

**Riassumendo**

**Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente**

**Attenzione ovviamente al posto del printf() si usa il cout senza gli specificatori di formato e apposto dello scanf() si usa il cin.**

**Riferimenti**

Una variabile tipo riferimento al tipo X è un ulteriore nome per una variabile X.

Ad esempio:

int n; int& r =n;

Una variabile di tipo riferimento al tipo deve essere stilizzata al momento della sua definizione.

**Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente**

**Passaggio di riferimenti a funzioni**

**Immagine che contiene testo, schermata, linea, numero

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, diagramma, Policromia

Descrizione generata automaticamente**

**Come posiamo far restituire ad una funzione un riferimento?**

**Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente**

* + Un riferimento è un puntatore costante il cui nome non è accessibile al programmatore, che ha per valore l’indirizzo dell’oggetto riferito: l’operazione che coinvolge l’oggetto riferito viene realizzata effettuando un indizione sul puntatore nascosto.
  + I riferimenti risultano utili per il programmatore ma non aggiungono alcuna potenzialità ai programmi rispetto all’uso dei puntatori.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

**L’opretore &(uderdend)**

L'operatore & ha tre usi in C++:

1. **Usato come prefisso di una variabile, restituisce l'indirizzo della variabile stessa.**
2. **Usato come suffisso di un tipo nella definizione di un riferimento, dichiara quest'ultimo come alias della variabile usata per la sua inizializzazione.**
3. **Usato come suffisso di un tipo nella dichiarazione dei parametri di una funzione, dichiara questi ultimi come riferimenti alle variabili passate alla funzione.**

**Puntatori a funzioni**

Non solo i dati, ma anche le istruzioni stanno in memoria a certi indirizzi; è possibile creare puntatori che puntino a funzioni, cioè al nome della funzione, che altro non è che un puntatore alla prima istruzione della funzione:

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente