PASSAGGIO DI VARIABILI

PER RIFERIMENTO IN C++

Classe 3 Sez.A inf.

FUNZIONI - C++

Partiamo dal problema dello scambio dei valori tra due variabili a, b

```
int a,b, temp;
a = 5;
b = 7;
temp = a;
a = b;
b = temp;
```

Questo semplice algoritmo abbiamo visto essere alla base di molti algoritmi di ordinamento

FUNZIONI - C++

Provate ora ad inserire questa porzione di codice in una funzione a parte !

L'intestazione sarà del tipo:

void scambia(int a, int b)

FUNZIONI - C++

E poi saremmo tentati di proseguire così...

```
void scambia(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

FUNZIONI - C++

```
void scambia(int a, int b){
      int temp;
      temp = a;
      a = b;
      b = temp;
}

int main() {
      int val1 = 5, val2=7;
      scambia(val1, val2);
      cout << "val1 = " << val1 << endl;
      cout << "val2 = " << val2 << endl;
}
```

Cosa produce in output ?

val1 = 5;

val2 = 7;

FUNZIONI - C++

Lo scambio, che è avvenuto all'interno della funzione scambia(...) non viene memorizzato dal main…

Perché ??? Le variabili sono passate per valore alla funzione scambia(....) che ne crea delle copie locali !

Lo scambio avviene tra variabili locali e quindi non riconosciute dal main….

FUNZIONI - C++

Come si può rimediare?

1) dichiarare due variabili globali. NO: abbiamo visto che il loro uso deve essere limitato ad esigenze particolari
2) abbandonare il progetto….. NO !
3) passaggio per riferimento. SI: specifichiamo all'esecutore che i parametri passati sono variabili del main e non copie locali

FUNZIONI - C++

Come avviene il passaggio per riferimento?

Passando alla funzione scambia(....) non i valori delle variabili, ma le variabili stesse ! (o meglio le locazioni di memoria occupate dalle variabili stesse)

FUNZIONI - C++

Come avviene il passaggio per riferimento?

Esempio:

```
int a = 5;
cout << "a =" << a << endl;
cout << "&a =" << &a << endl;
```

cosa produce in output? l'operatore & applicato ad una variabile restituisce il suo indirizzo di memoria. Provate!

FUNZIONI - C++

RISULTATO:

a = 5 (valore di a)

&a = 0x7ffeef4f7b4c (indirizzo di memoria della variabile a)

FUNZIONI - C++

Come avviene il passaggio per riferimento ?

```
void scambia(int &a, int &b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

In questo modo vengono passati gli indirizzi di memoria delle variabili e le modifiche sono valide anche nel main !

FUNZIONI - C++

```
#include <iostream>
using namespace std;
void scambia(int &a, int &b){
     int temp;
     temp = a;
     a = b;
     b = temp;
}
int main(){
    int a = 5, b = 7;
    cout << "prima dello scambio" <<endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    scambia(a,b);
    cout << "dopo lo scambio" <<endl;
    cout << "a = " << a<< endl;
    cout << "b = " << b<< endl;
    return 0;
}
```

FUNZIONI - C++

Output:

```
prima dello scambio
a = 5
b = 7

dopo lo scambio
a = 7
b = 5
```

FUNZIONI - C++

Il passaggio di ARRAY a funzioni avviene SEMPRE per riferimento

Le modifiche apportate nella funzione all'array rimangono anche nel main e nelle altre funzioni!