Lezione 2 II semestre

visibilità e gestione delle variabili

sezione 3.6 del testo

Visibilità delle variabili

In un programma ci sono tante variabili

sarebbe inefficiente allocarle sempre tutte

in ogni momento vengono allocate solo quelle che servono in quel momento per l'esecuzione del programma

le variabili esistono solo nel blocco in cui sono dichiarate
blocco = {.....}

in un blocco non ci possono essere più di una dichiarazione di una variabile con un certo nome

```
{
  int x =1;
  char x=‘a’;  //ERRORE
 .............
}
```

ma,

```
{
  int x =1;
     {char x=‘a’;.....}  //OK
}
```

quando l’esecuzione entra in un blocco (=> {) ,
le variabili dichiarate in quel blocco sono
allocate

quando l’esecuzione esce dal blocco (}=>)
quelle variabili sono deallocate

tutto questo avviene automaticamente

variabili automatiche

blocchi annidati

```
//(*)
{ int x=0; //(0)
   {
      int y=1; //(1)
     {
        int z=2; //(2)
      }
      cout<<x<<y<<endl; // (1')
    }
   cout<<x<<endl; // (0')
}//(*')
```

La pila dei dati cresce e diminuisce dinamicamente

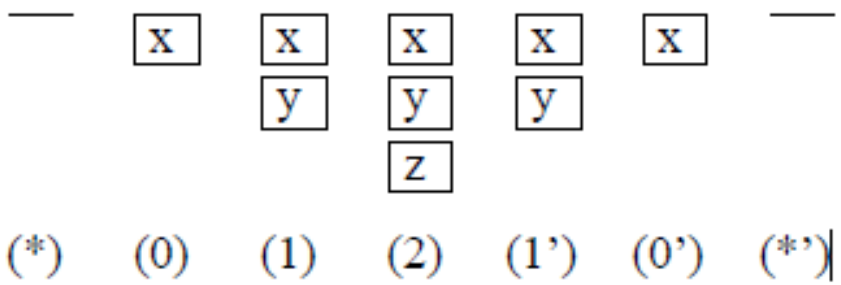

Figura 3.3: Allocazione e deallocazione della memoria durante l'esecuzione di un programma

i dati sono gestiti con una pila
su cui si fanno 2 operazioni: push (aggiungi) e pop (togli) sempre sulla cima della pila

si tratta degli R-valori delle variabili

```
{ int x=0; //(0)
   {
      int y=1; //(1)
      {
         int z=2, x = 4; cout<<x+z;//(2)
      }
      cout<<x<<y<<endl; // (1')
   }
  cout<<x<<endl; // (0')
}//(*')
```

x=0
y=1
z=2 x=4

(2)

la x di (0) è oscurata da quella di (2)