



POLITECNICO
DI MILANO

INFORMATICA

Passaggio di
parametri di tipo
array

```
if (numC != 0)
{cout << endl << setw(20)
 << "rilevazioni zona "
 << setw(10) << "centro" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaC / numC;
for (i = 0; i < numC; i++)
{ deviazione = prezziC[i] - media;
  cout << setw(20) << prezziC[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione
    << endl;
}
}

if (numP != 0)
{cout << endl << setw(20) << "rilevazioni zona "
 << setw(10) << "periferia" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaP / numP;
for (i = 0; i < numP; i++)
{ deviazione = prezziP[i] - media;
  cout << setw(20) << prezziP[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione << endl;
}
```

```
if (numC != 0)
{cout << endl << setw(20)
 << "rilevazioni zona "
 << setw(10) << "centro" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaC / numC;
for (i = 0; i < numC; i++)
{ deviazione = prezziC[i] - media;
  cout << setw(20) << prezziC[i]
  << endl << setprecision(2)
  << setiosflags (ios::fixed ! ios::showpoint)
  << deviazione << endl;
  sommaC += prezziC[i];
}
} prezziC[]
```

numC ←→ **numP**
"centro" ←→ **"periferia"**
sommaC ←→ **sommaP**
stesso tipo

prezziC[] ←→ **prezziP[]**

```
if (numP != 0)
{cout << endl << setw(20) << "rilevazioni zona "
 << setw(10) << "periferia" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaP / numP;
for (i = 0; i < numP; i++)
{ deviazione = prezziP[i] - media;
  cout << setw(20) << prezziP[i]
  << endl << setprecision(2)
  << setiosflags (ios::fixed ! ios::showpoint)
  << deviazione << endl;
}
```

Fattorizzazione

```
if (numC != 0)
{cout << endl << setw(20)
 << "rilevazioni zona "
 << setw(10) << "centro" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaC / numC;
for (i = 0; i < numC; i++)
{ deviazione = prezziC[i] - media;
  cout << setw(20) << prezziC[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione
    << endl;
}
}

if (numP != 0)
{cout << endl << setw(20) << "rilevazioni zona "
 << setw(10) << "periferia" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaP / numP;
for (i = 0; i < numP; i++)
{ deviazione = prezziP[i] - media;
  cout << setw(20) << prezziP[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione << endl;
}
```

```
if (numC != 0)
{cout << endl << setw(20)
 << "rilevazioni zona "
 << setw(10) << "centro" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaC / numC;
for (i = 0; i < numC; i++)
{ deviazione = prezziC[i] - media;
  cout << setw(20) << prezziC[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione
    << endl; ...
}
}

if (numP != 0)
{cout << endl << setw(20) << "rilevazioni zona "
 << setw(10) << "periferia" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaP / numP;
for (i = 0; i < numP; i++)
{ deviazione = prezziP[i] - media;
  cout << setw(20) << prezziP[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione << endl;
}
```

```
if (numC != 0)
{cout << endl << setw(20)
 << "rilevazioni zona "
 << setw(10) << "centro" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaC / numC;
for (i = 0; i < numC; i++)
{ deviazione = prezziC[i] - media;
  cout << setw(20) << prezziC[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione
    << endl;
}
}

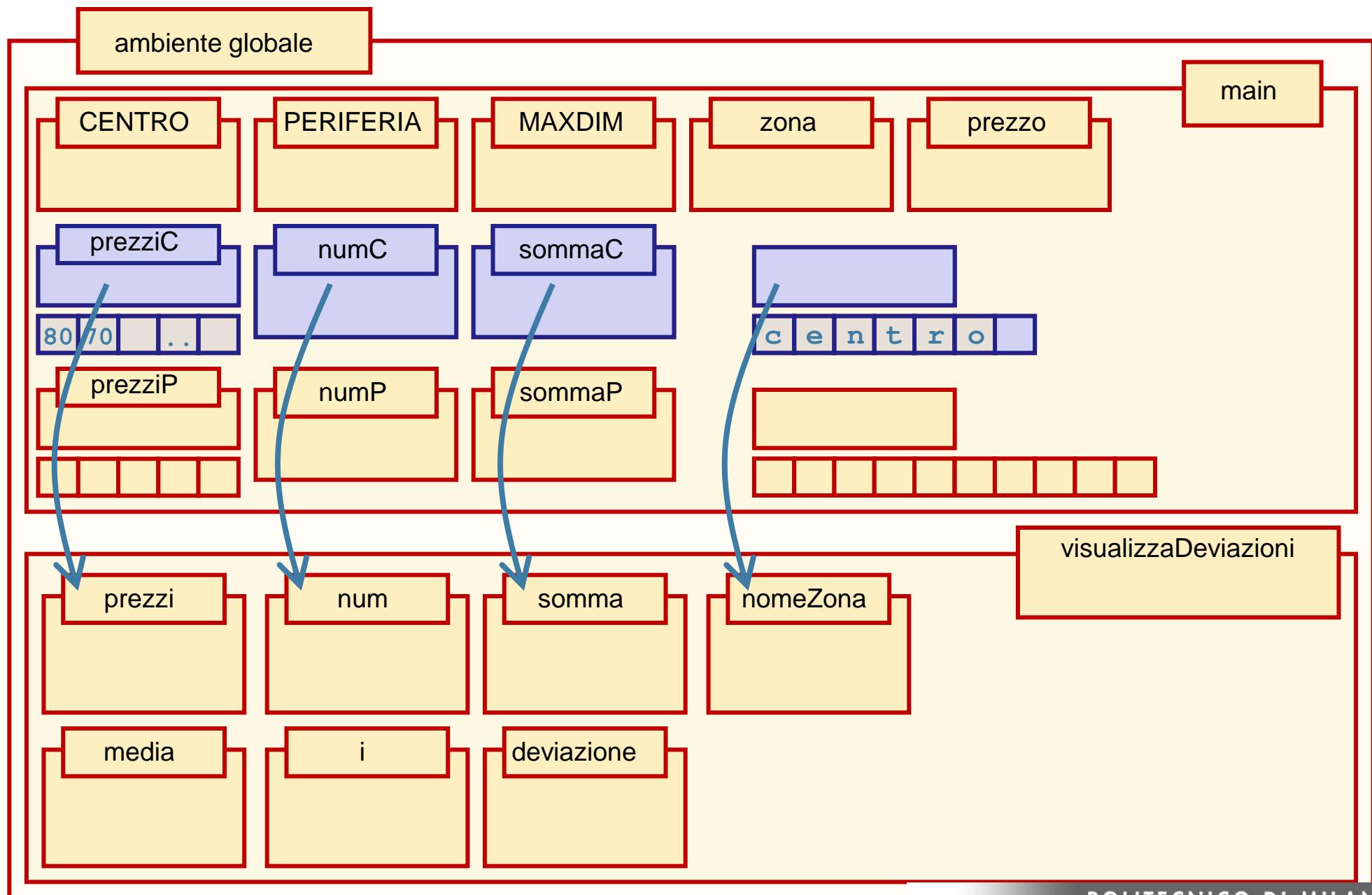
if (numP != 0)
{cout << endl << setw(20) << "rilevazioni zona "
 << setw(10) << "periferia" << endl << endl
 << setw(20) << "prezzi rilevati"
 << setw(40) << "deviazione rispetto al prezzo medio" << endl;
media = (float)sommaP / numP;
for (i = 0; i < numP; i++)
{ deviazione = prezziP[i] - media;
  cout << setw(20) << prezziP[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed ! ios::showpoint)
    << deviazione << endl;
}
```

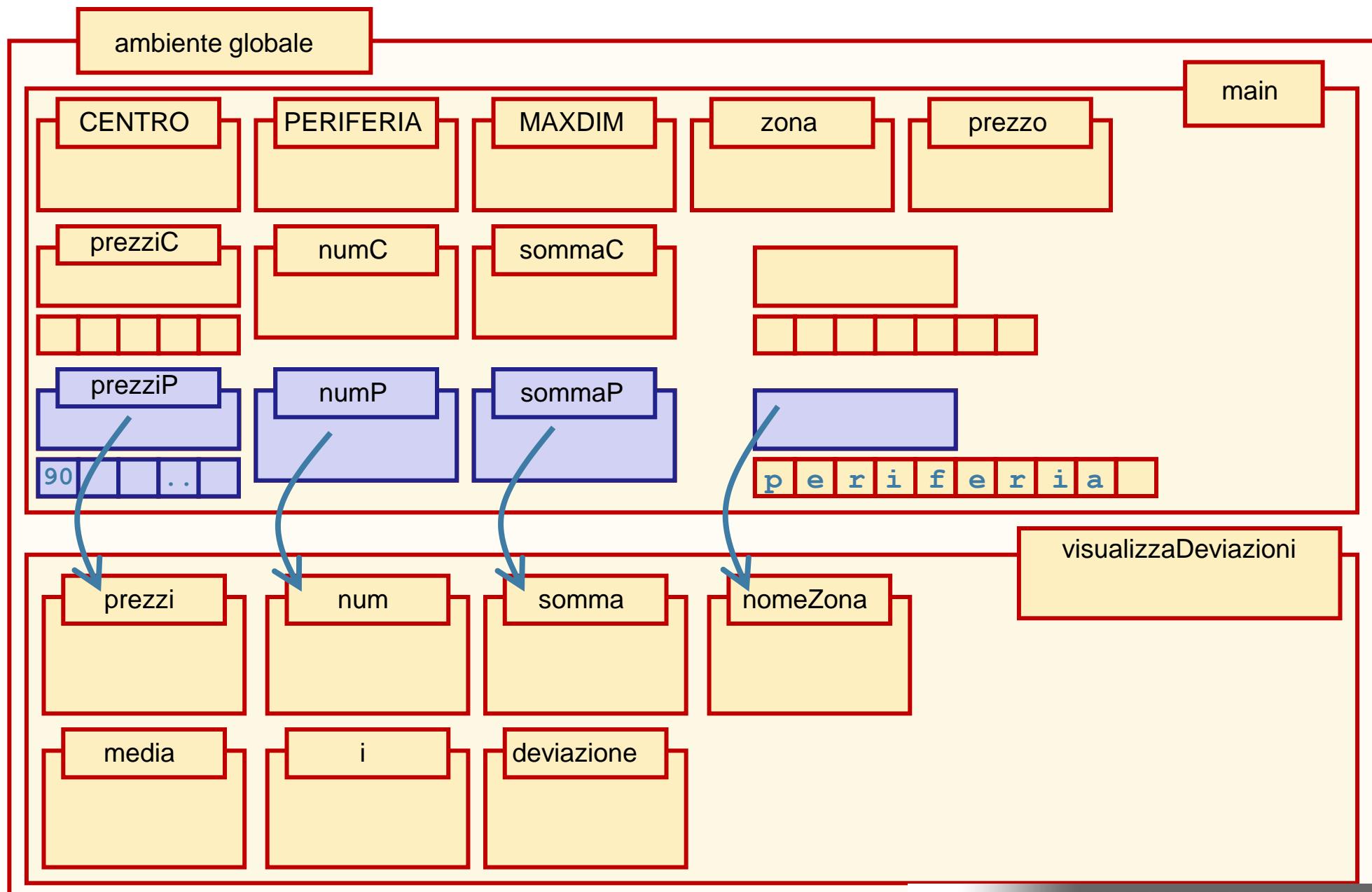
Array come parametri

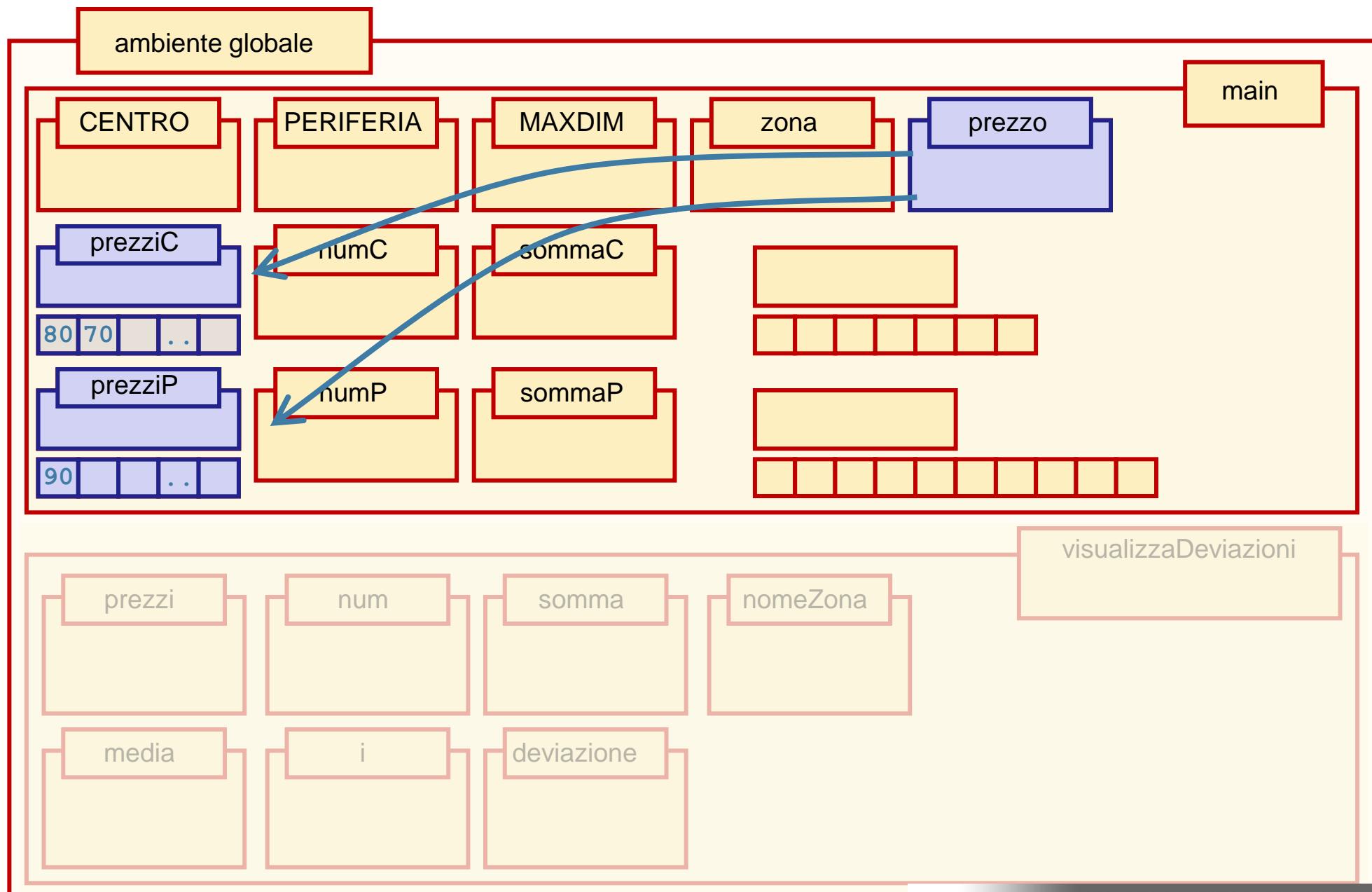
```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
    cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

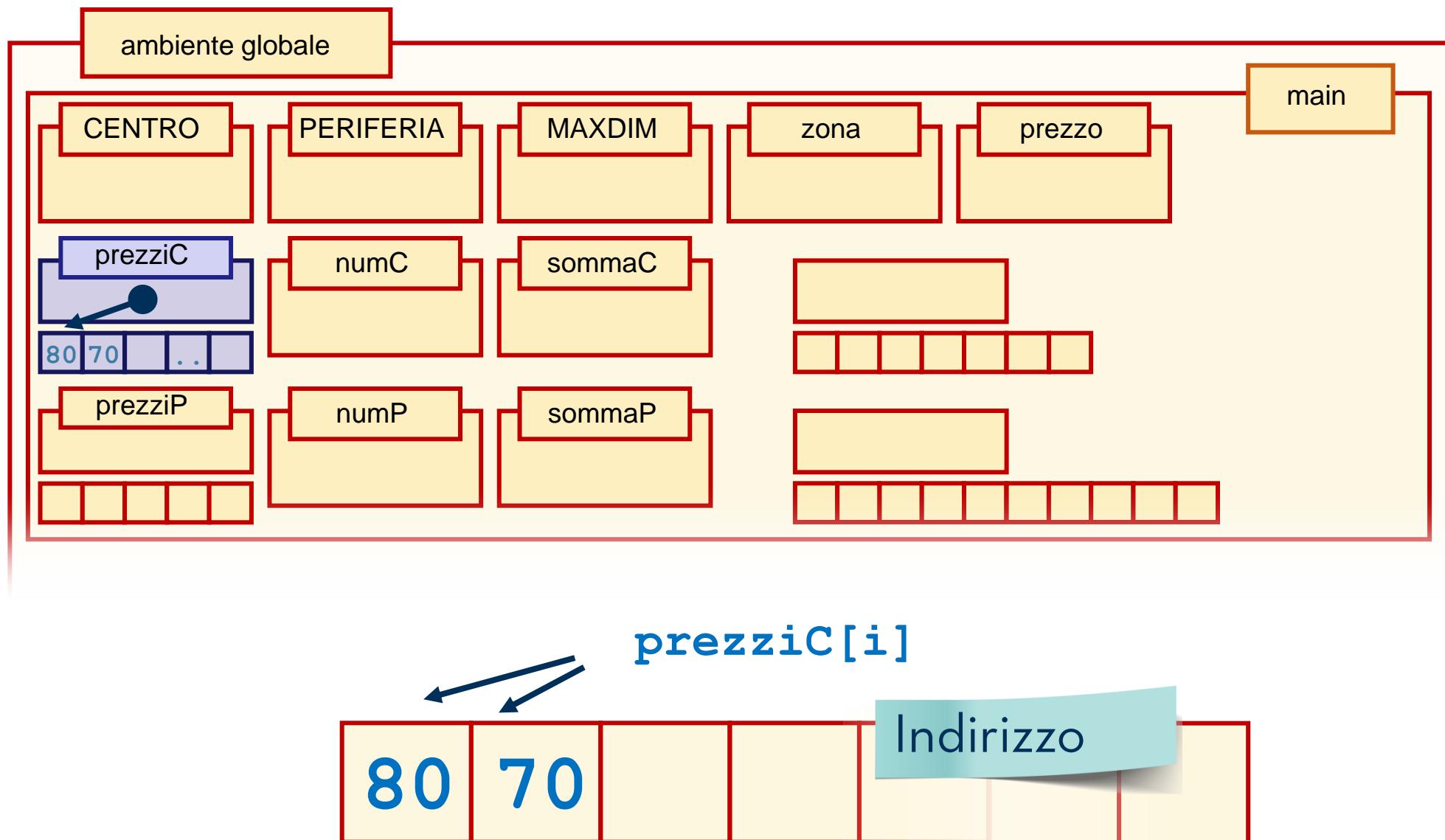
```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

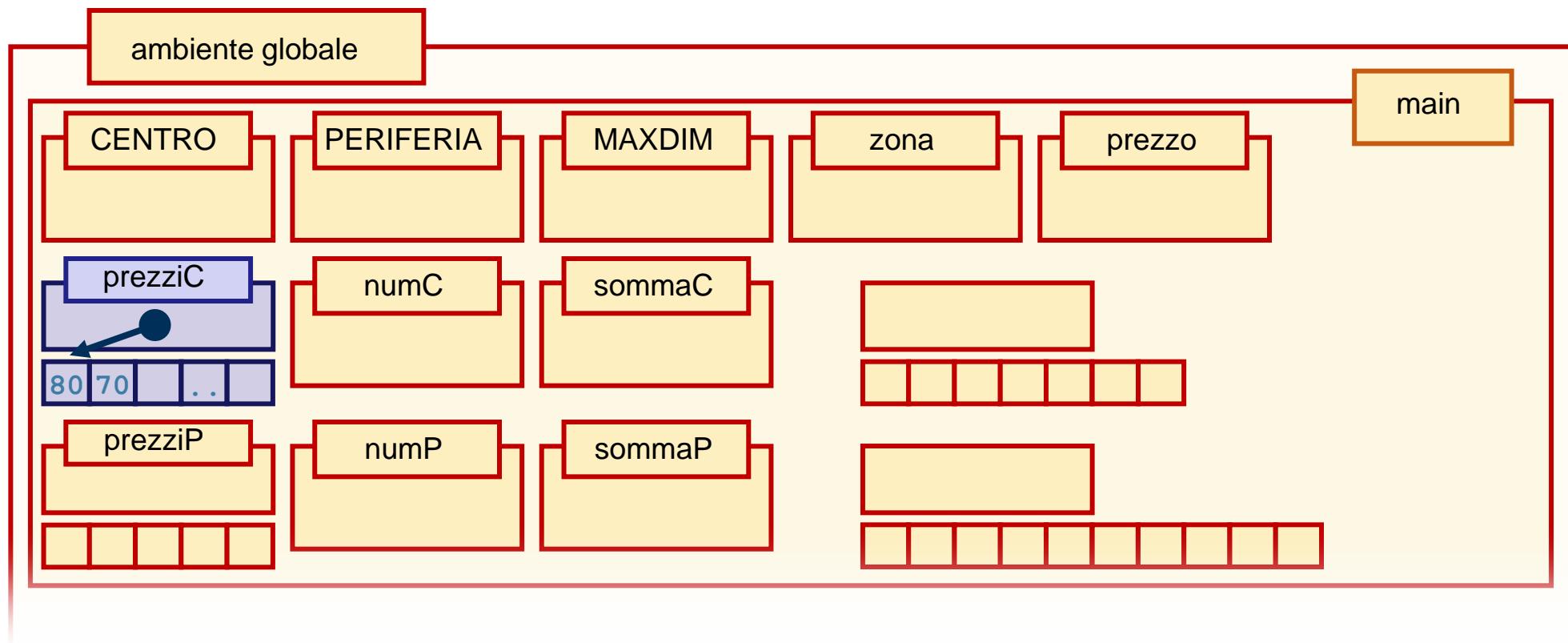
Che significato ha
il nome di un array
in C++



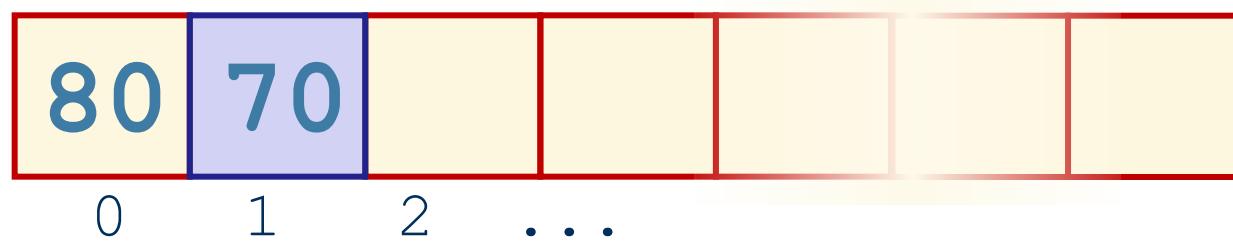


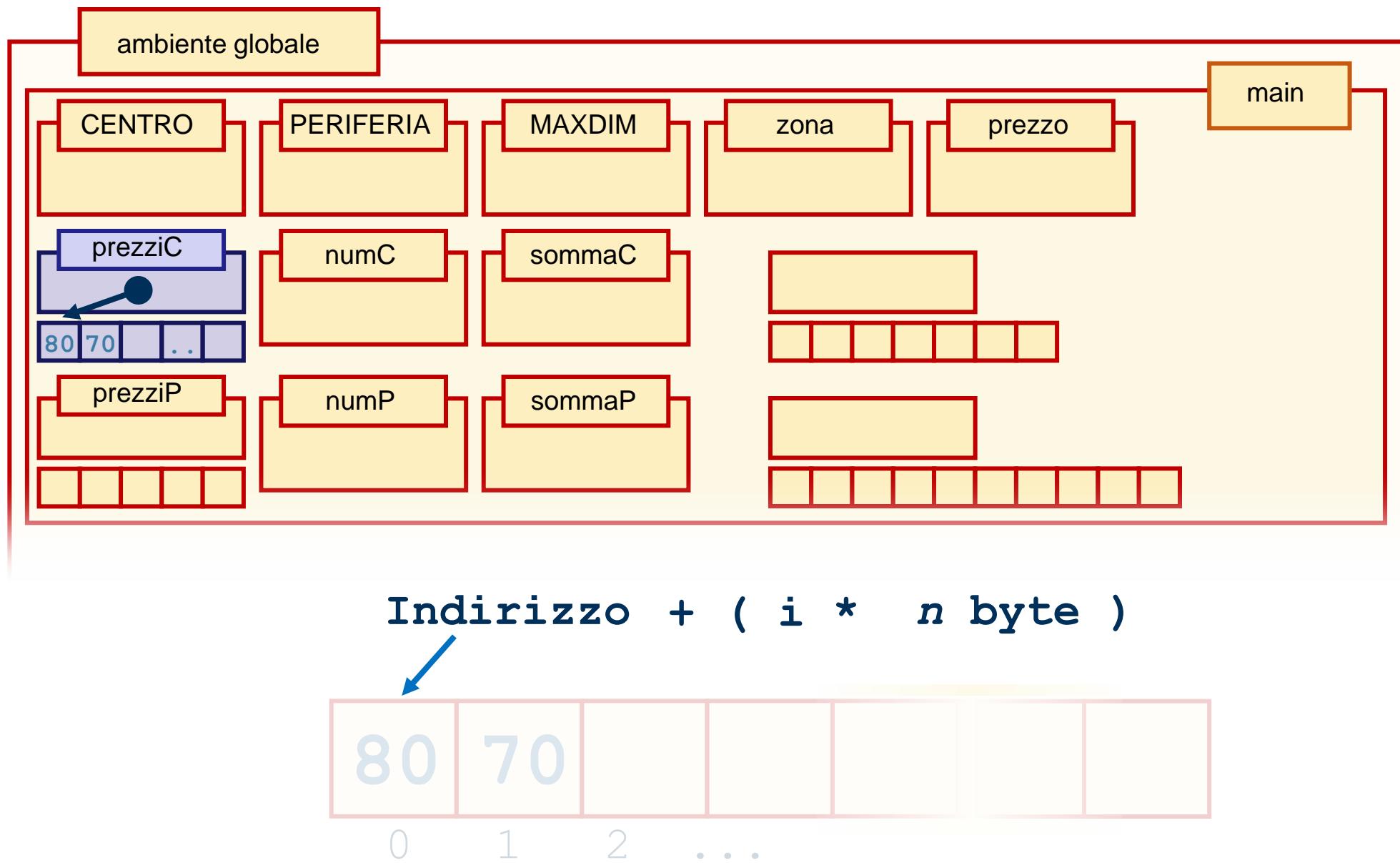


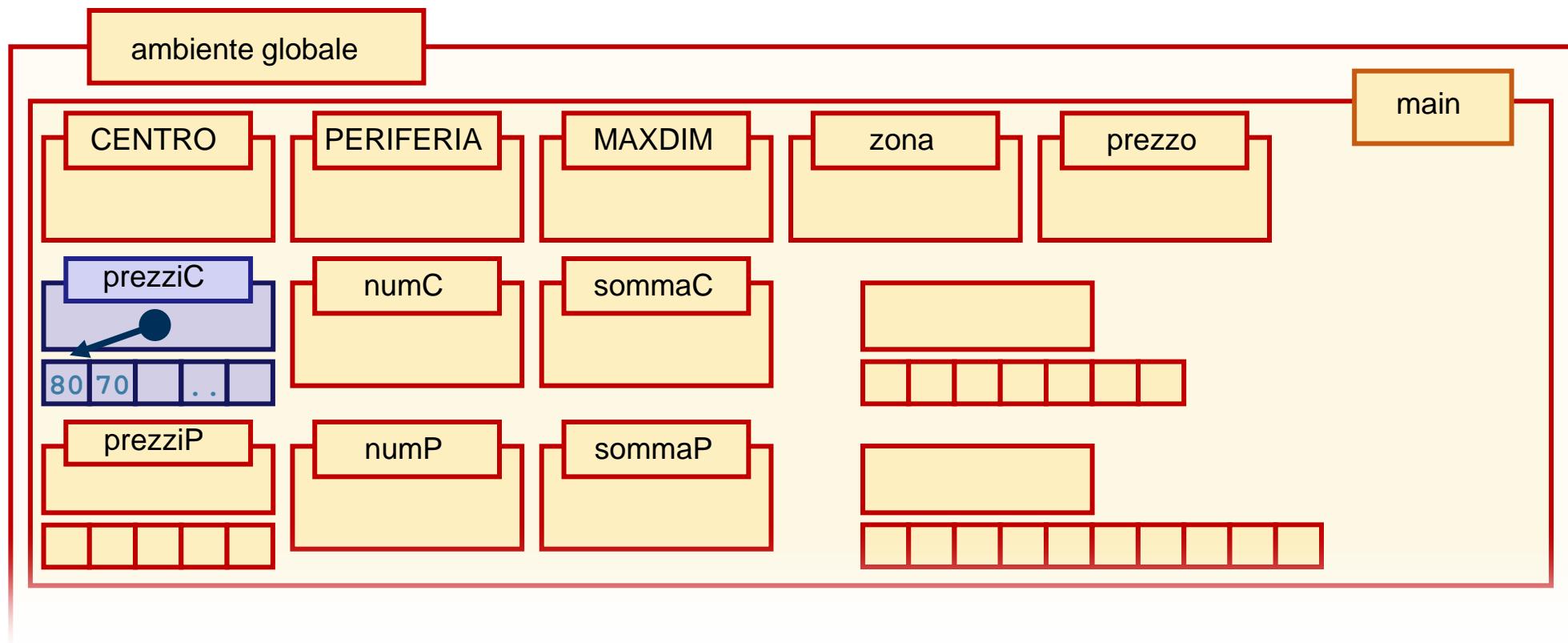




prezziC[1]

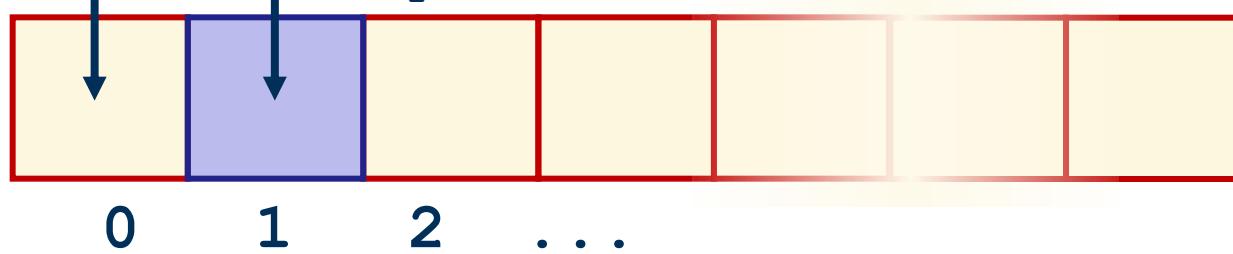


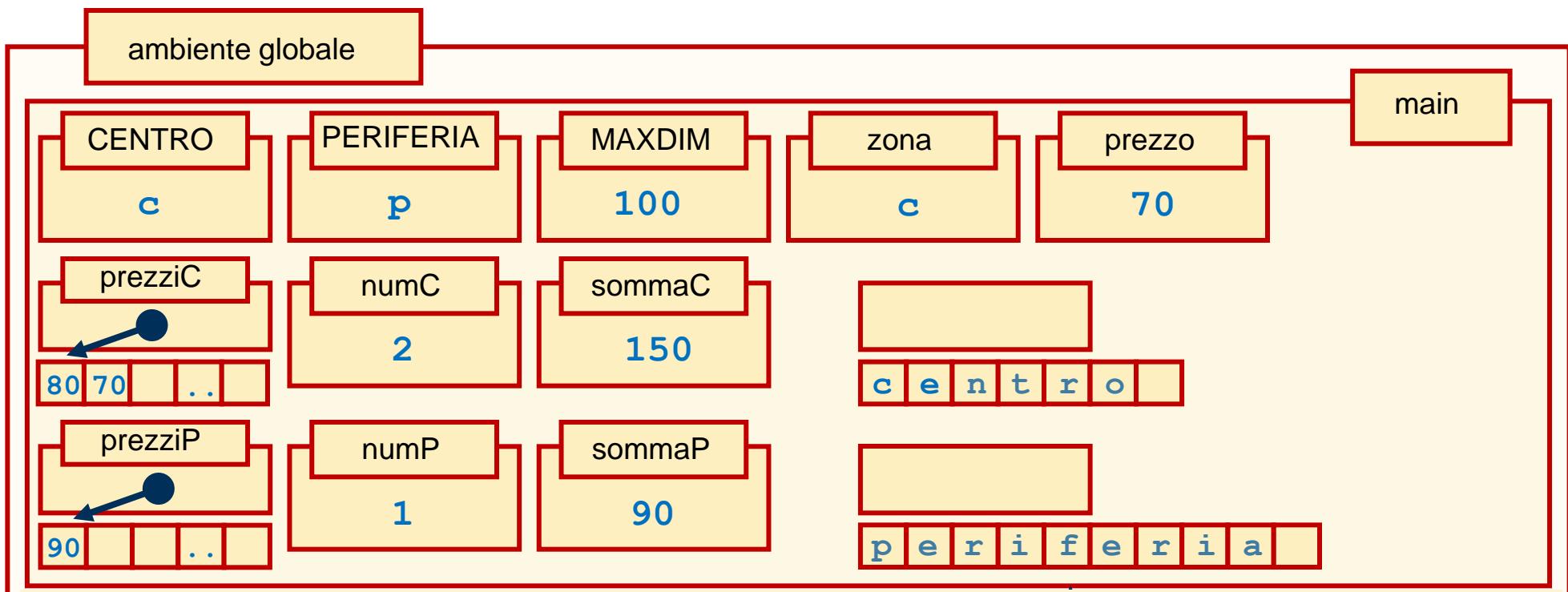


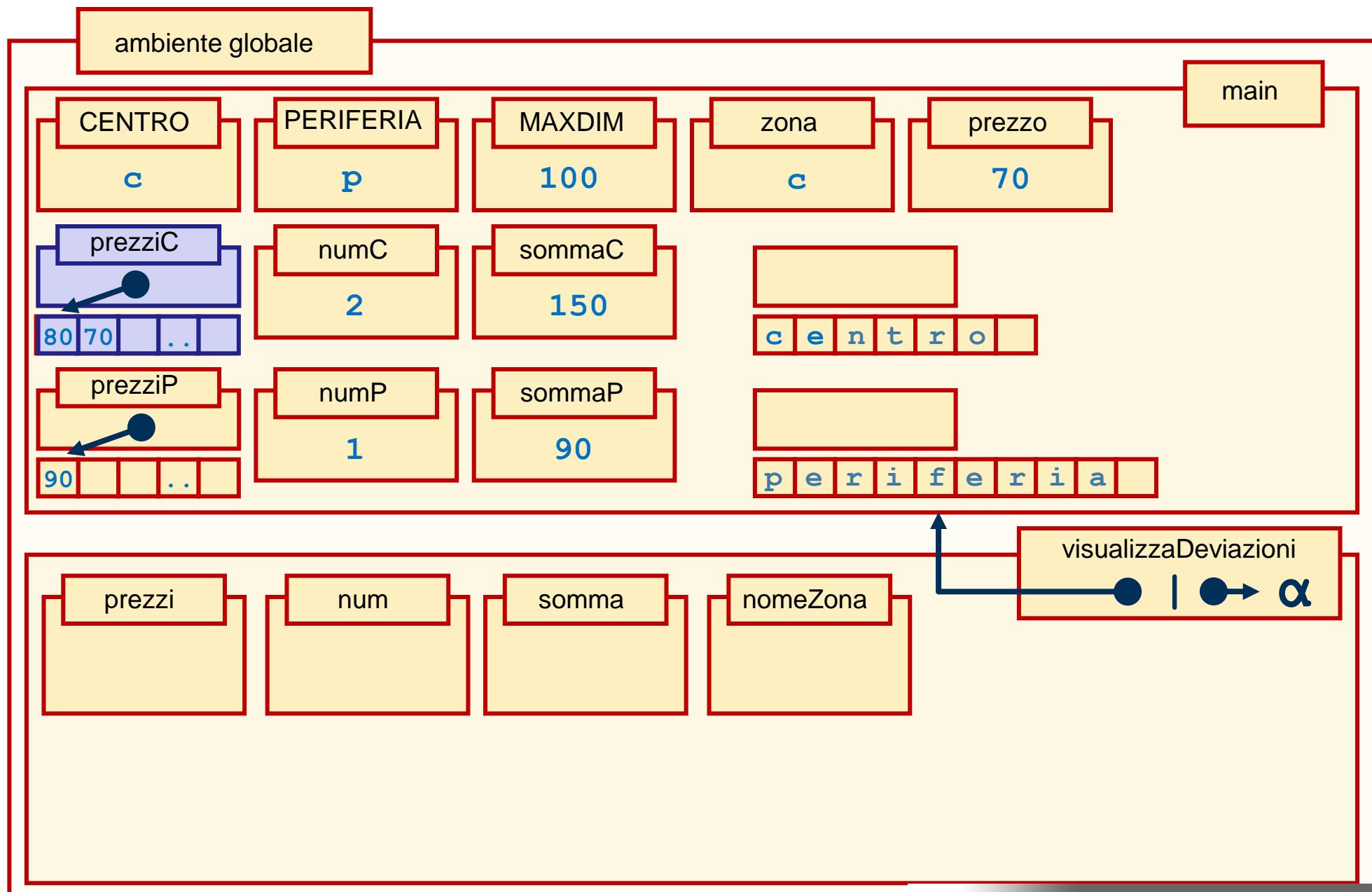


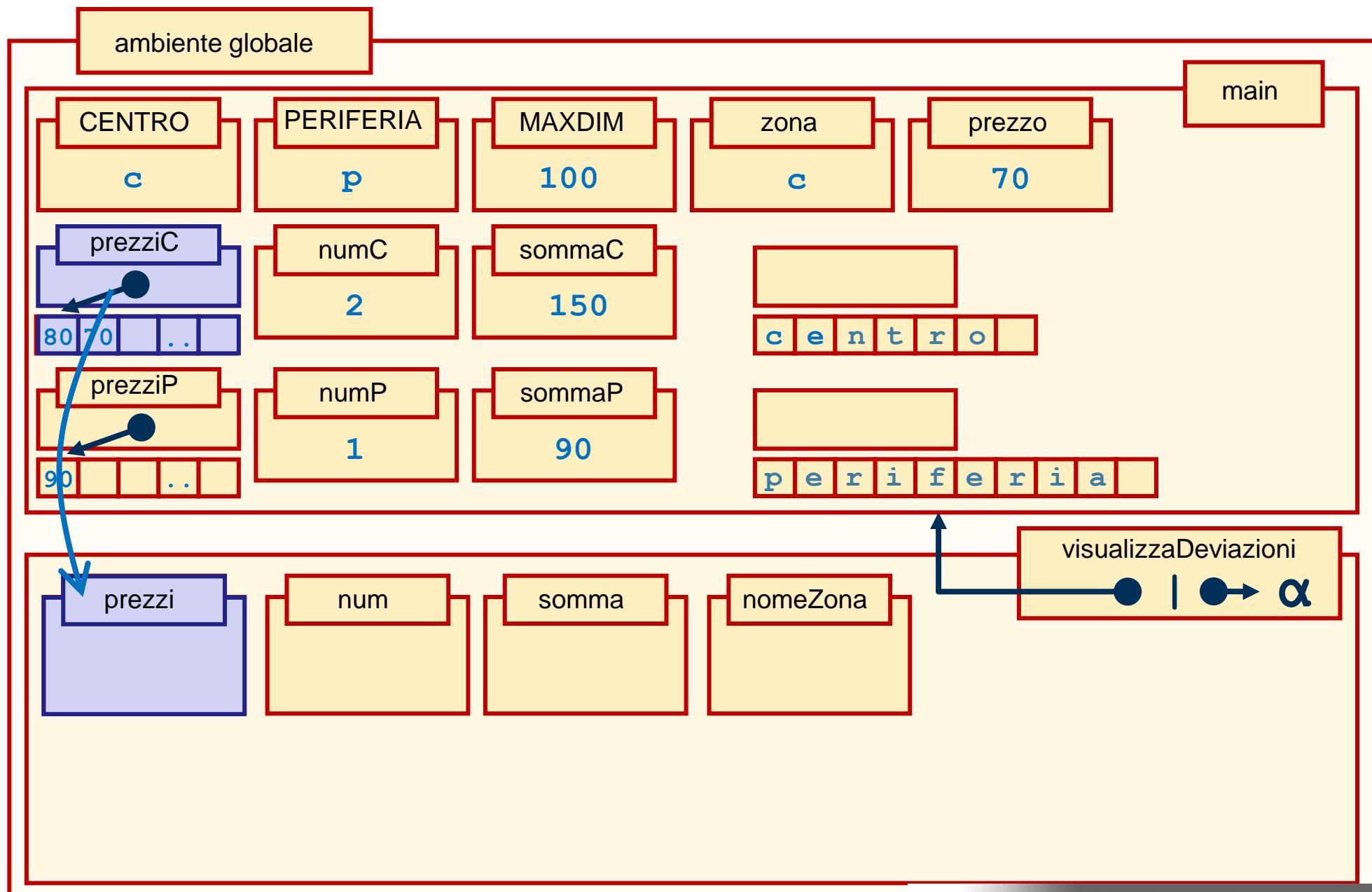
prezziC[1]

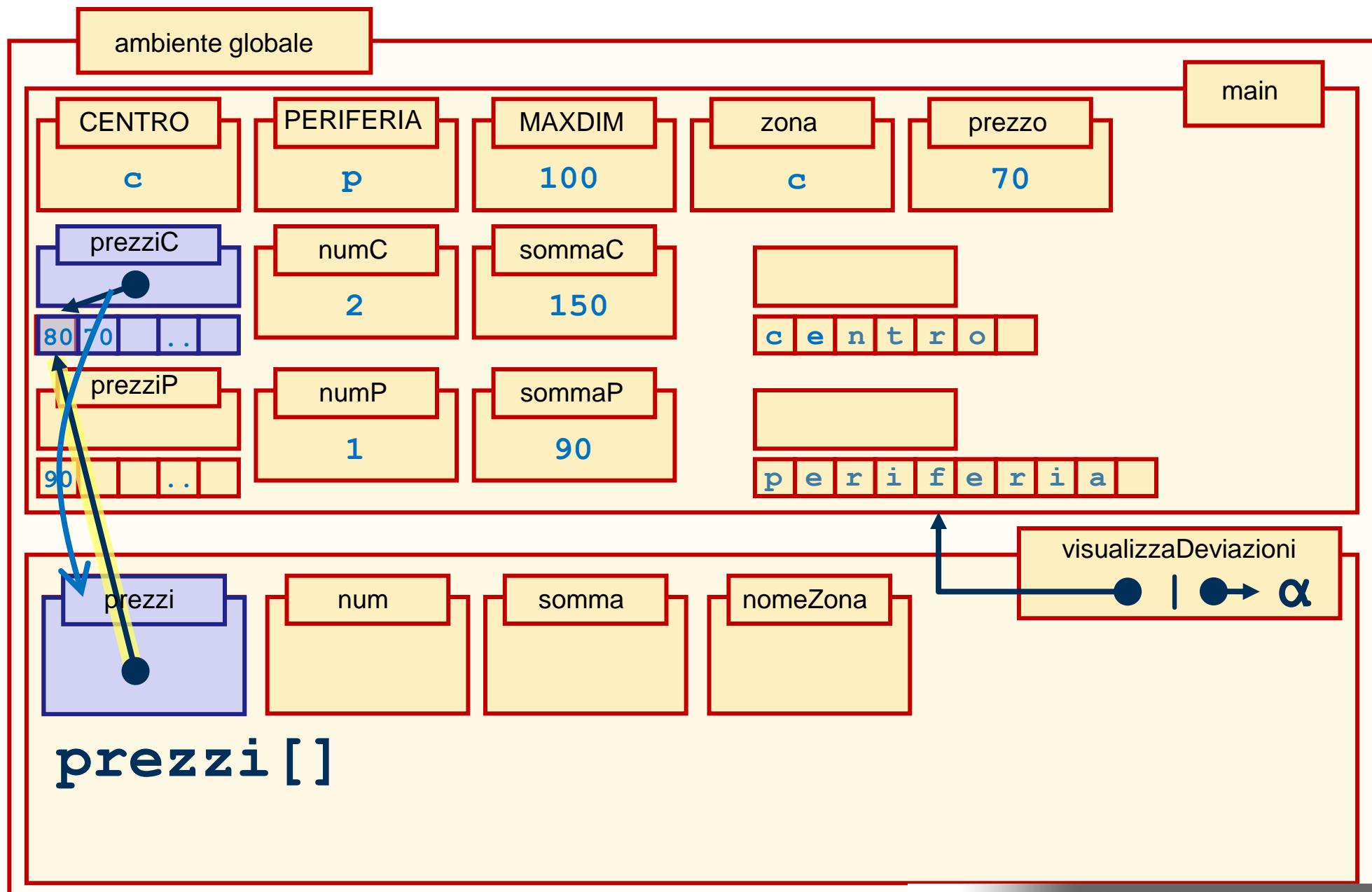
Indirizzo + (n nyte * 1)

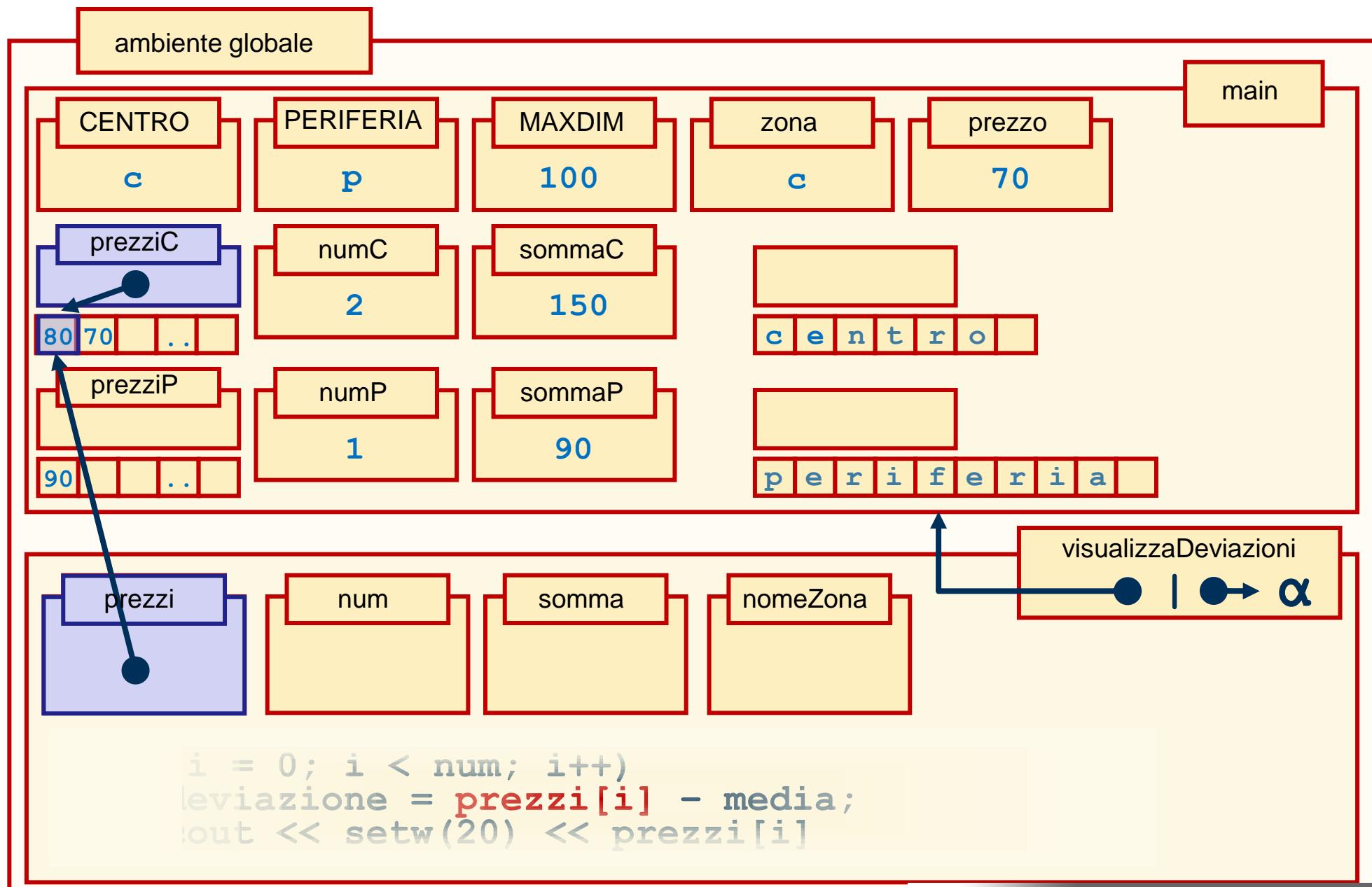


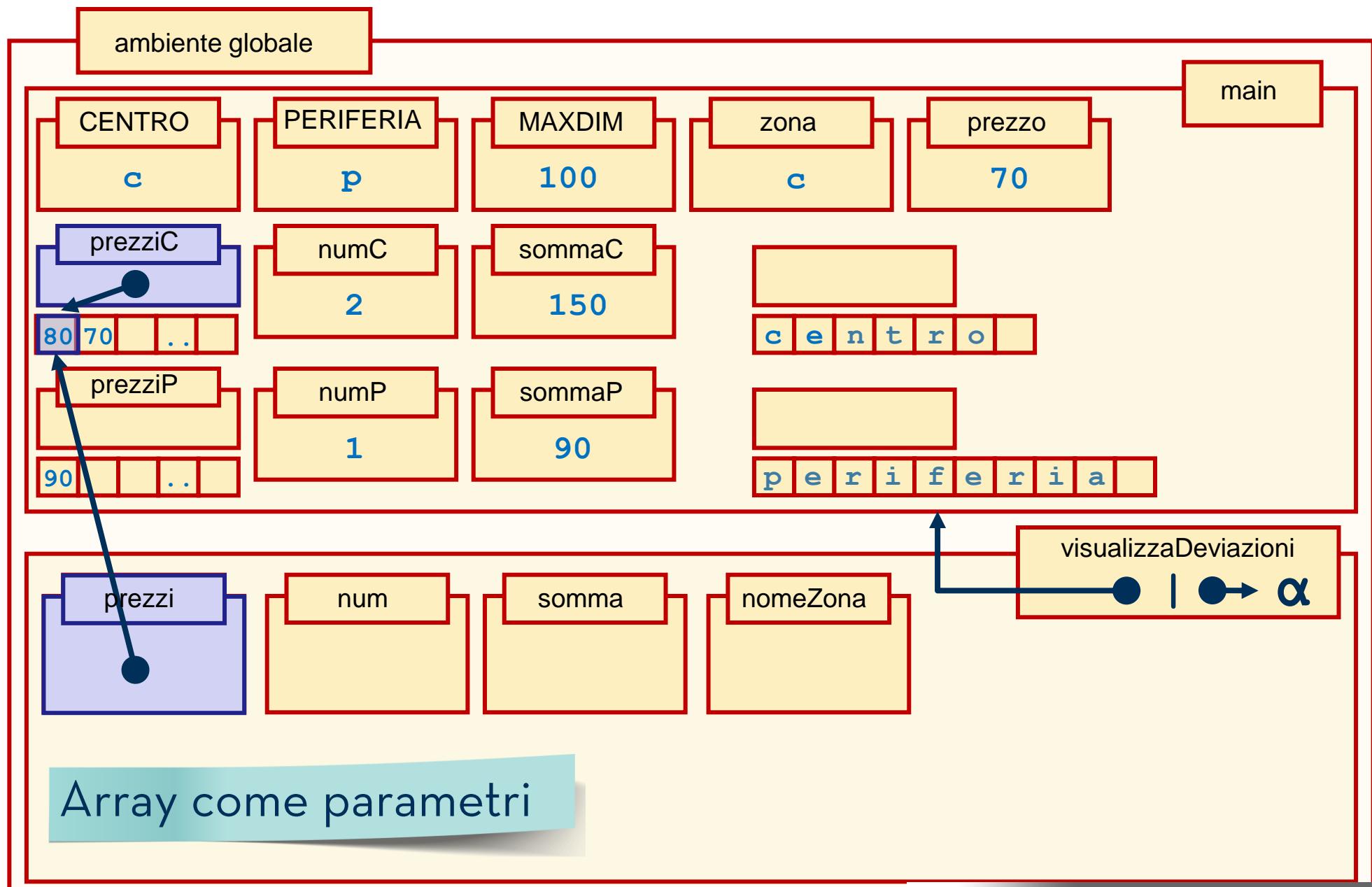












ambiente globale

main

```
void visualizzaDeviazioni(...)  
    (char nomeZona[], int somma, int num, int prezzi[])  
{ float media, deviazione;  
int i;  
if (num != 0)  
{ cout << endl << setw(20)  
    << "rilevazioni zona "  
    << setw(10) << nomeZona << endl << endl  
    << setw(20) << "prezzi rilevati"  
    << setw(40) << "deviazione rispetto al prezzo medio"  
    << endl;  
media = (float) somma / num;  
for (i = 0; i < num; i++)  
{ deviazione = prezzi[i] - media;  
cout << setw(20) << prezzi[i]  
    << setw(40) << setprecision(2)  
    << setiosflags (ios::fixed | ios::showpoint)  
    << deviazione  
    << endl;  
}  
}  
}
```

visualizzaDeviazioni

array

```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

array

```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

array []

```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

Verifica di corrispondenza

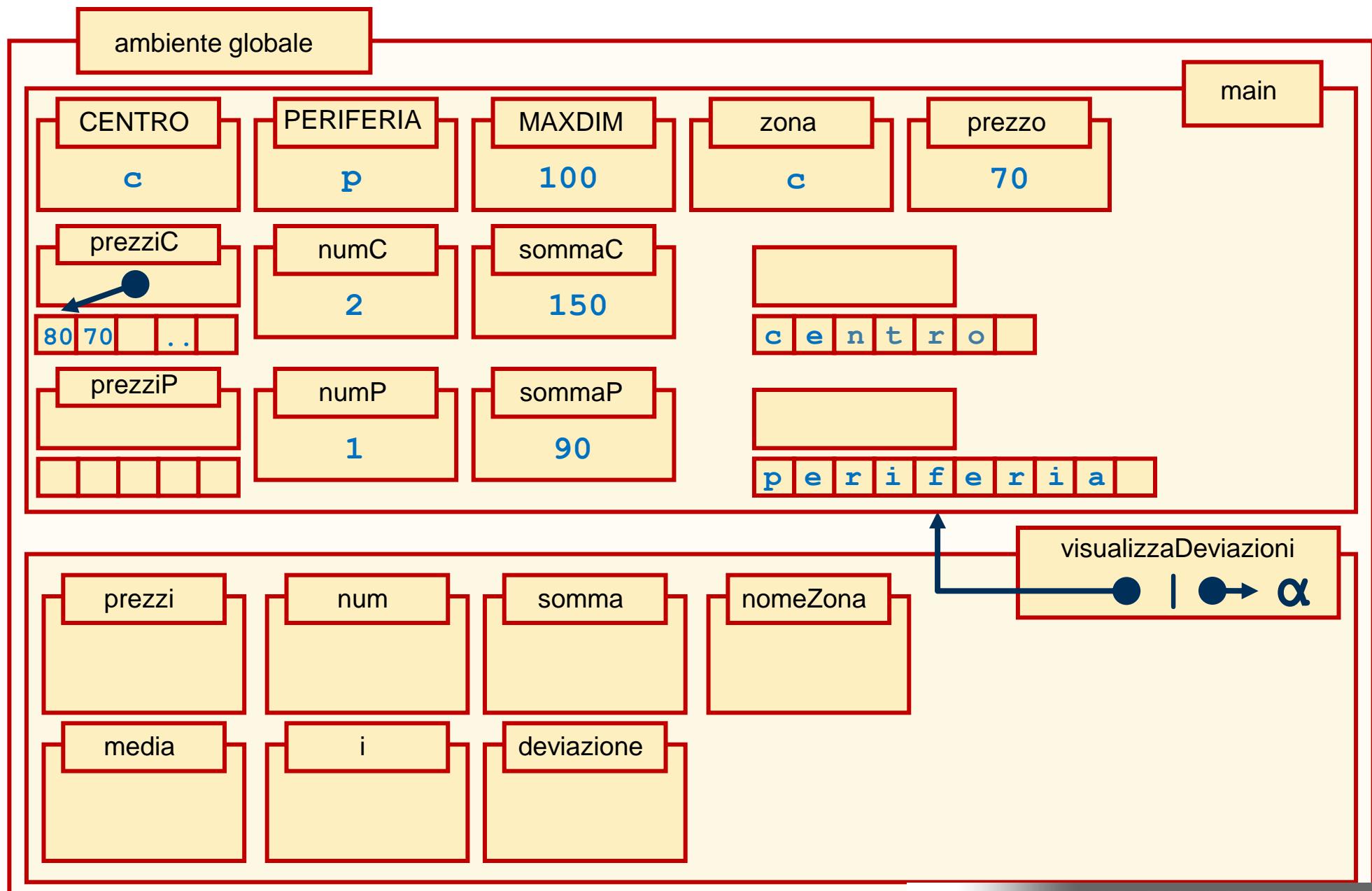
```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

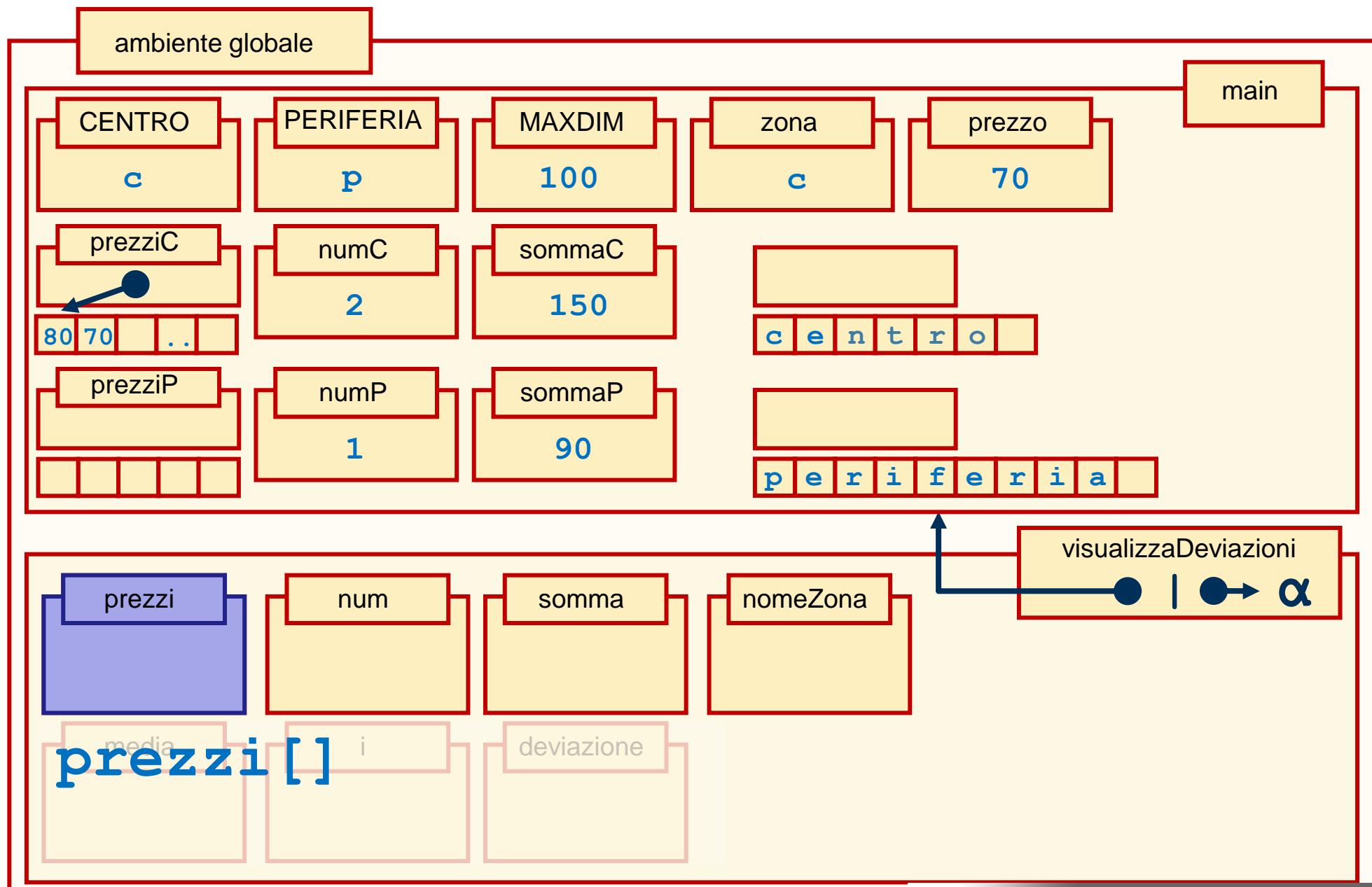
Verifica di corrispondenza

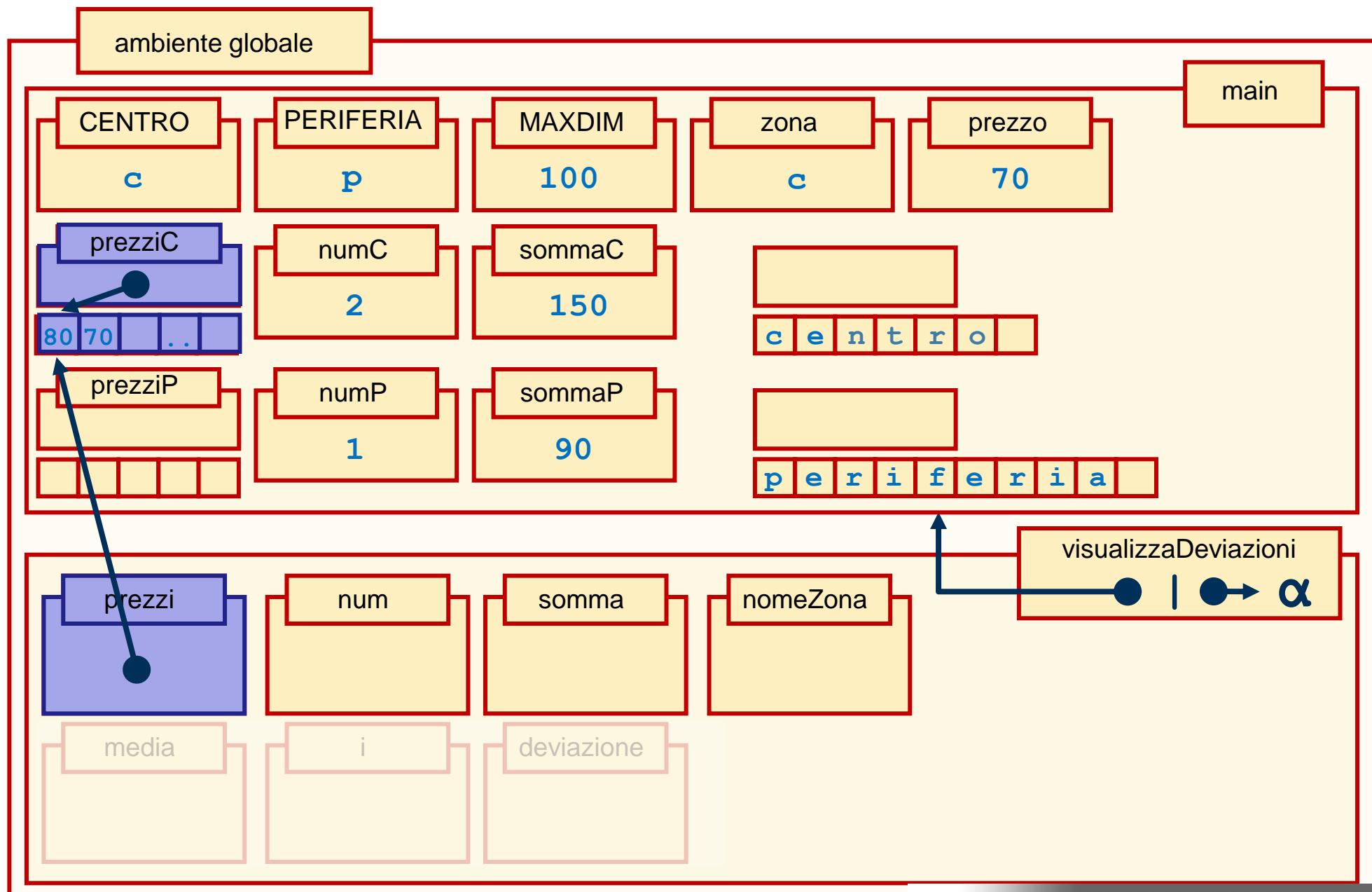
```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

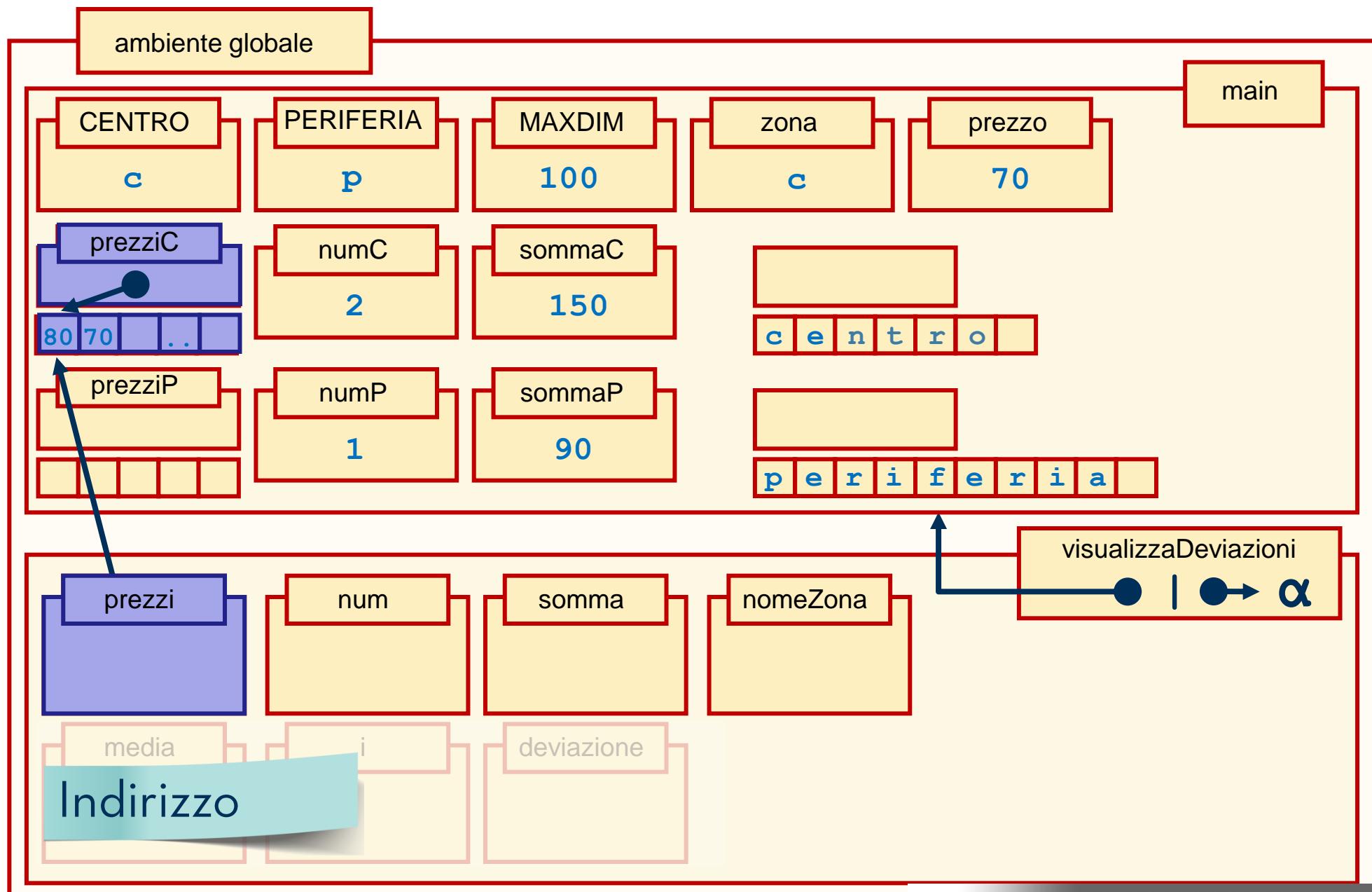
Verifica di corrispondenza

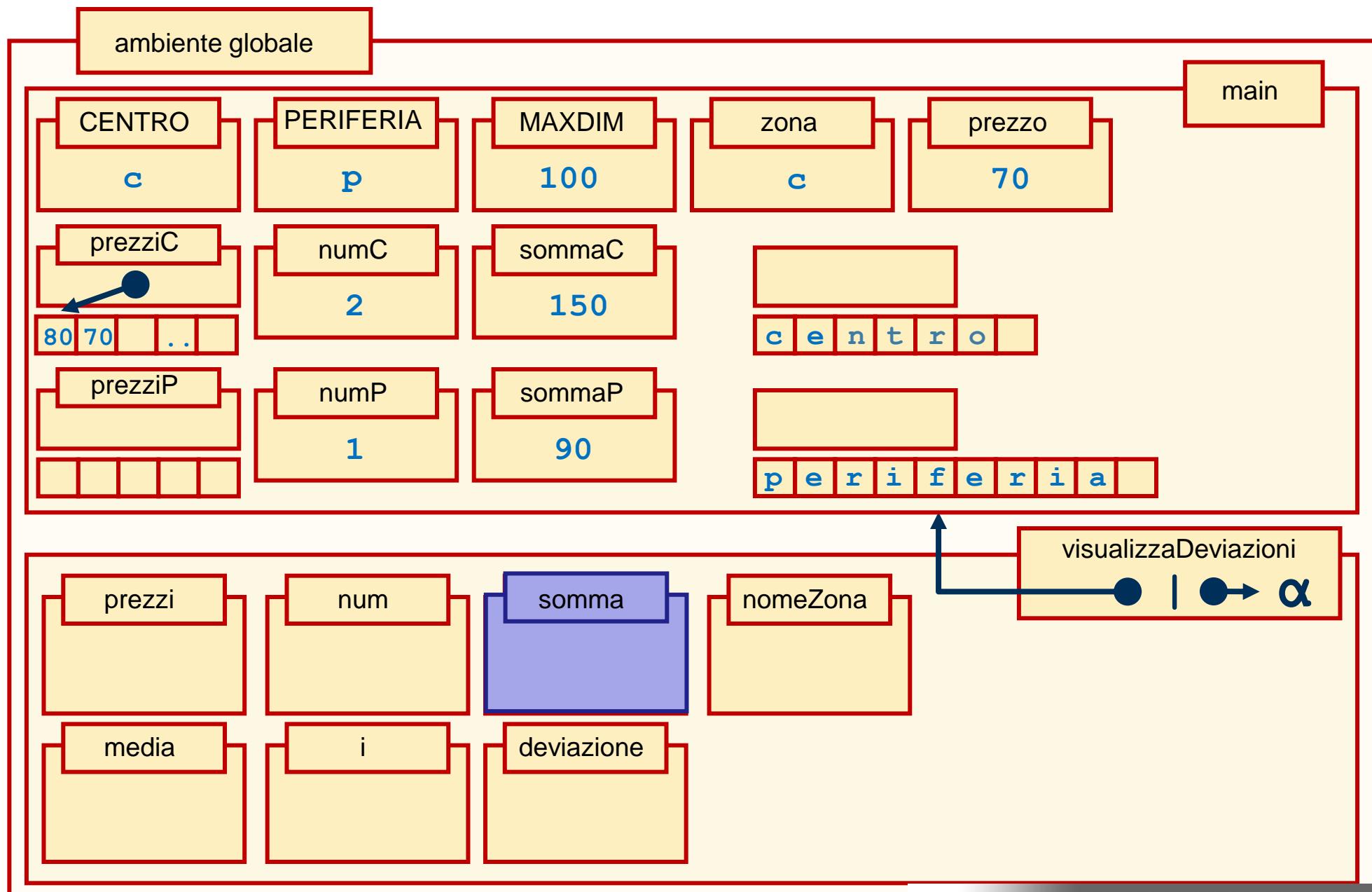
```
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi - media;
cout << setw(20) << prezzi
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

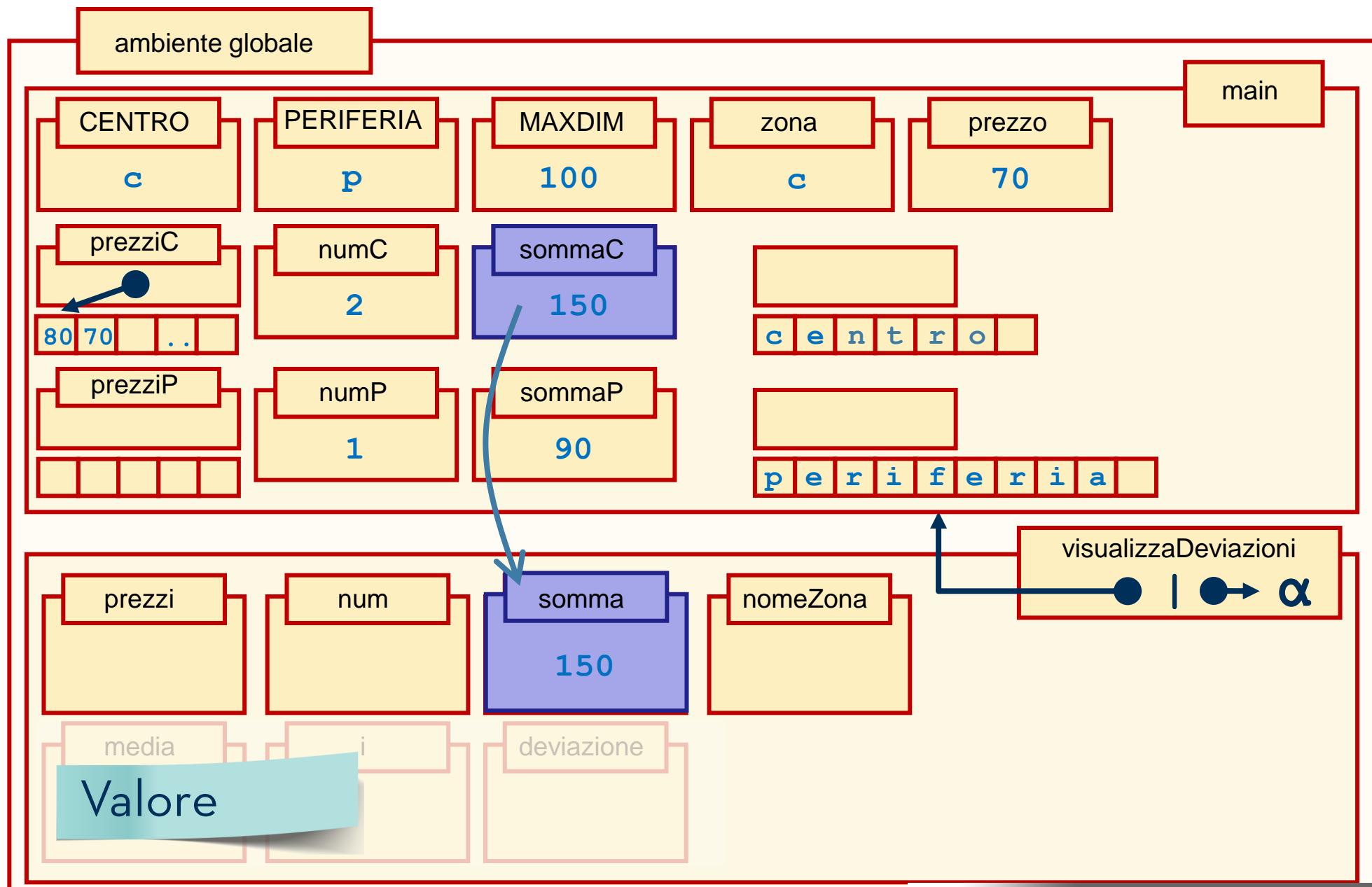


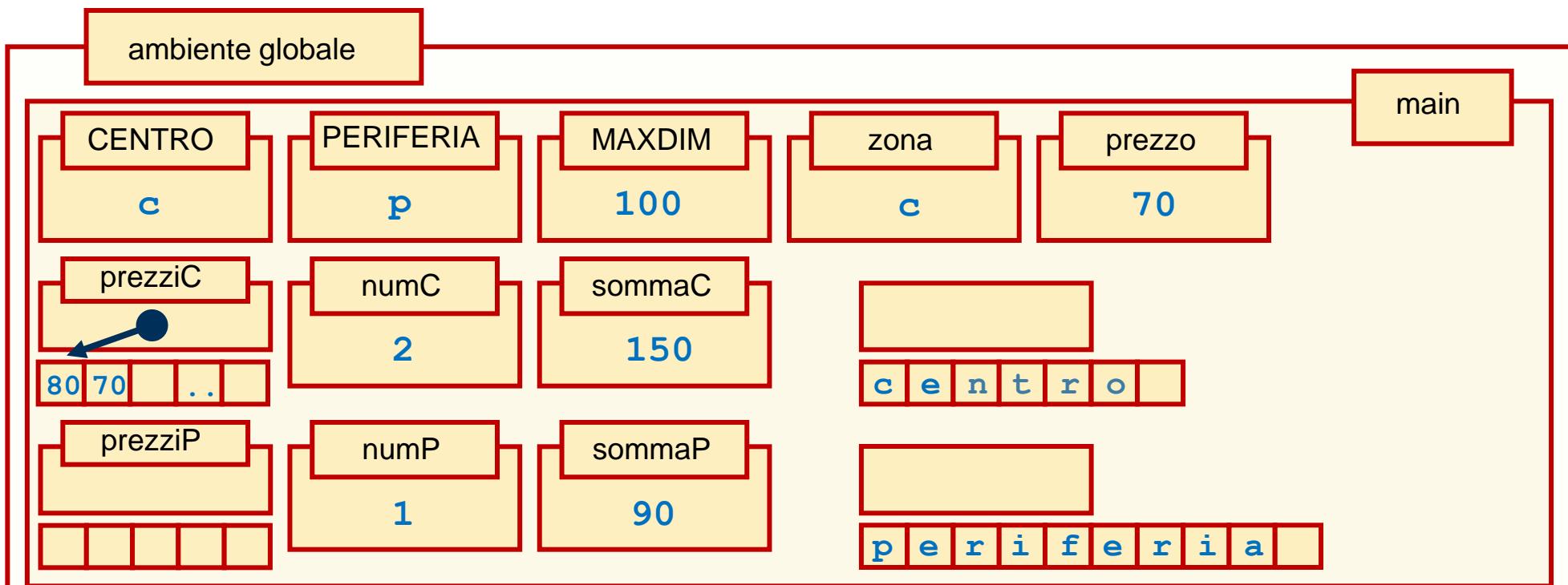




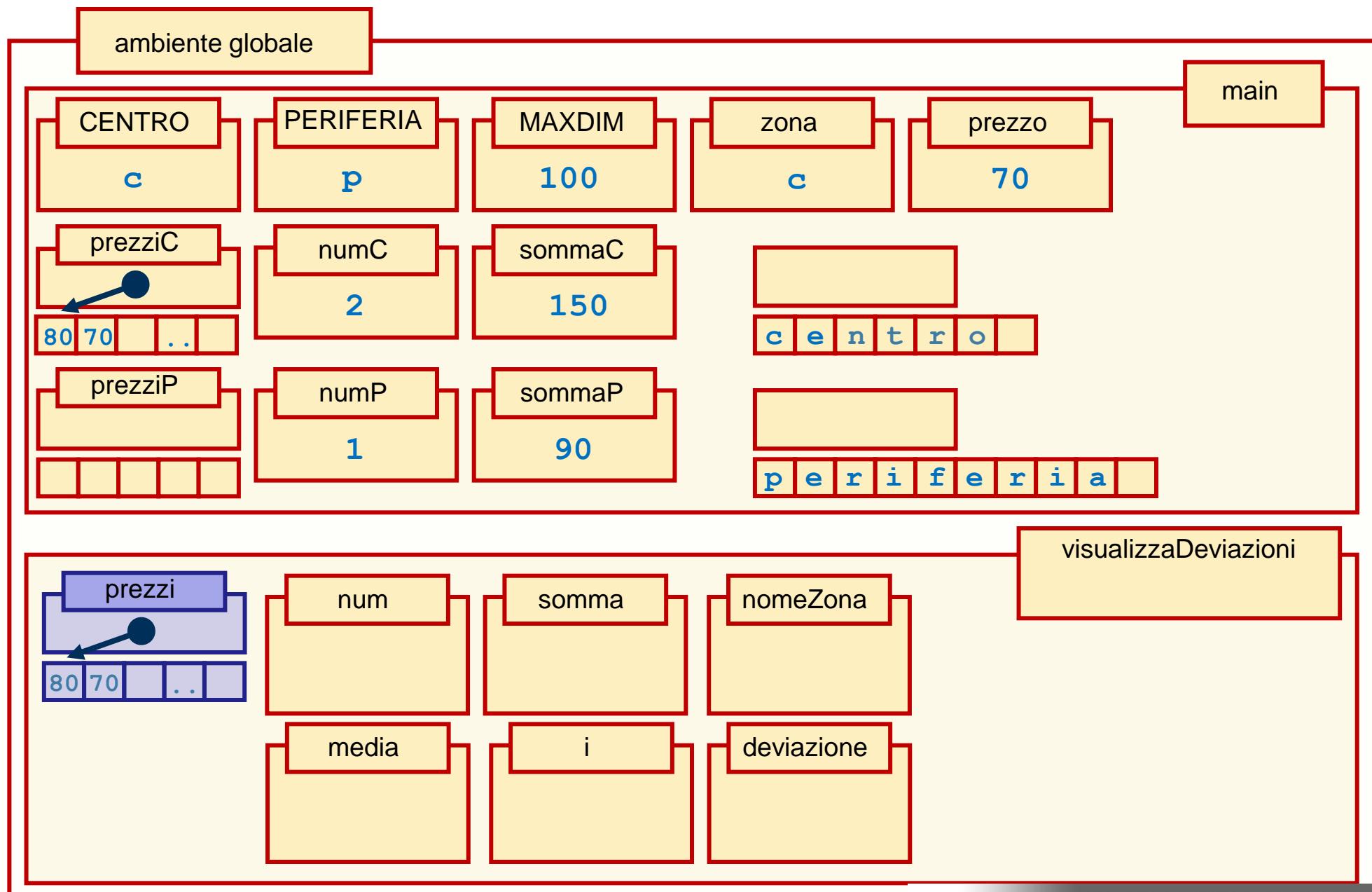


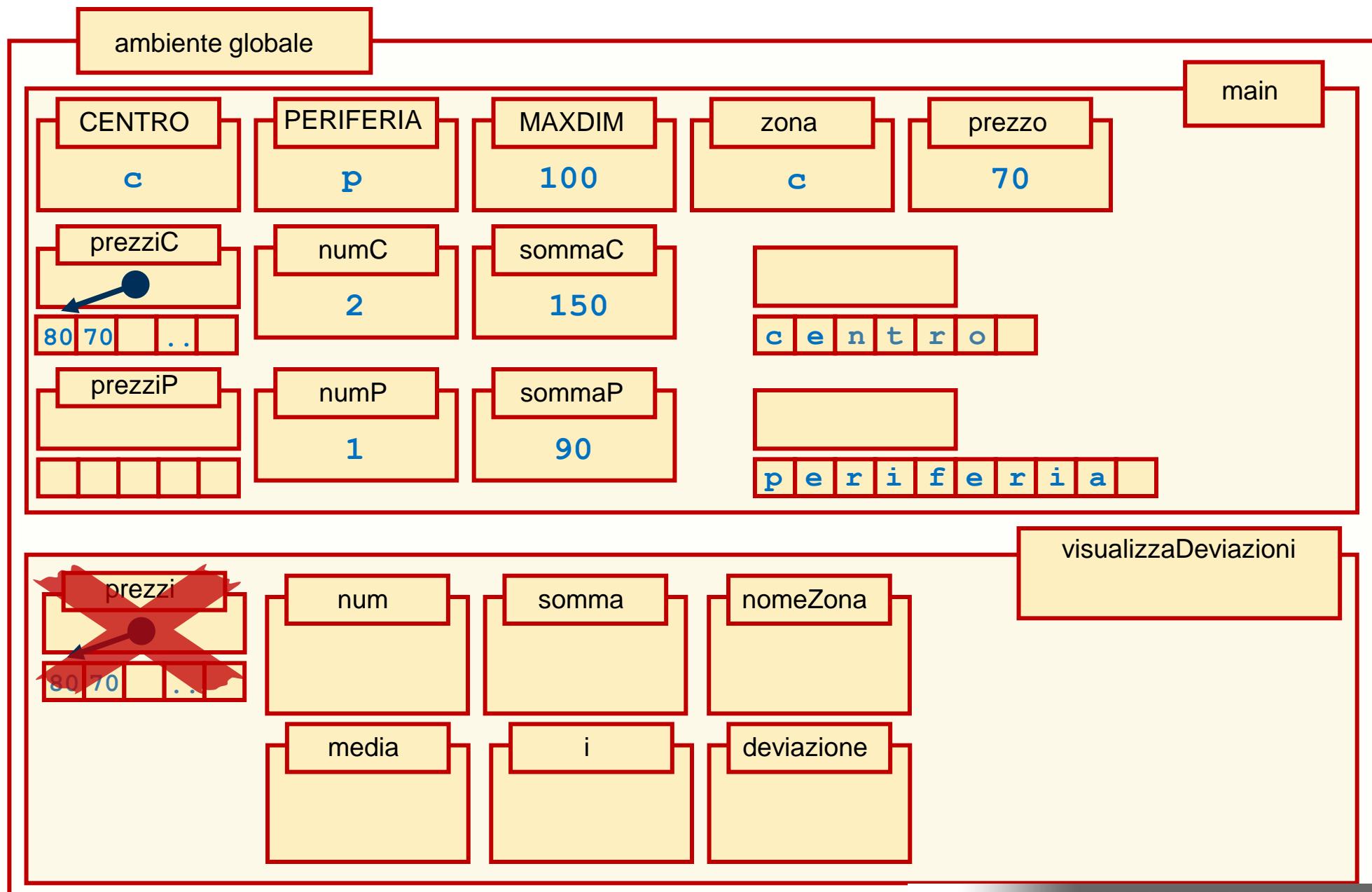


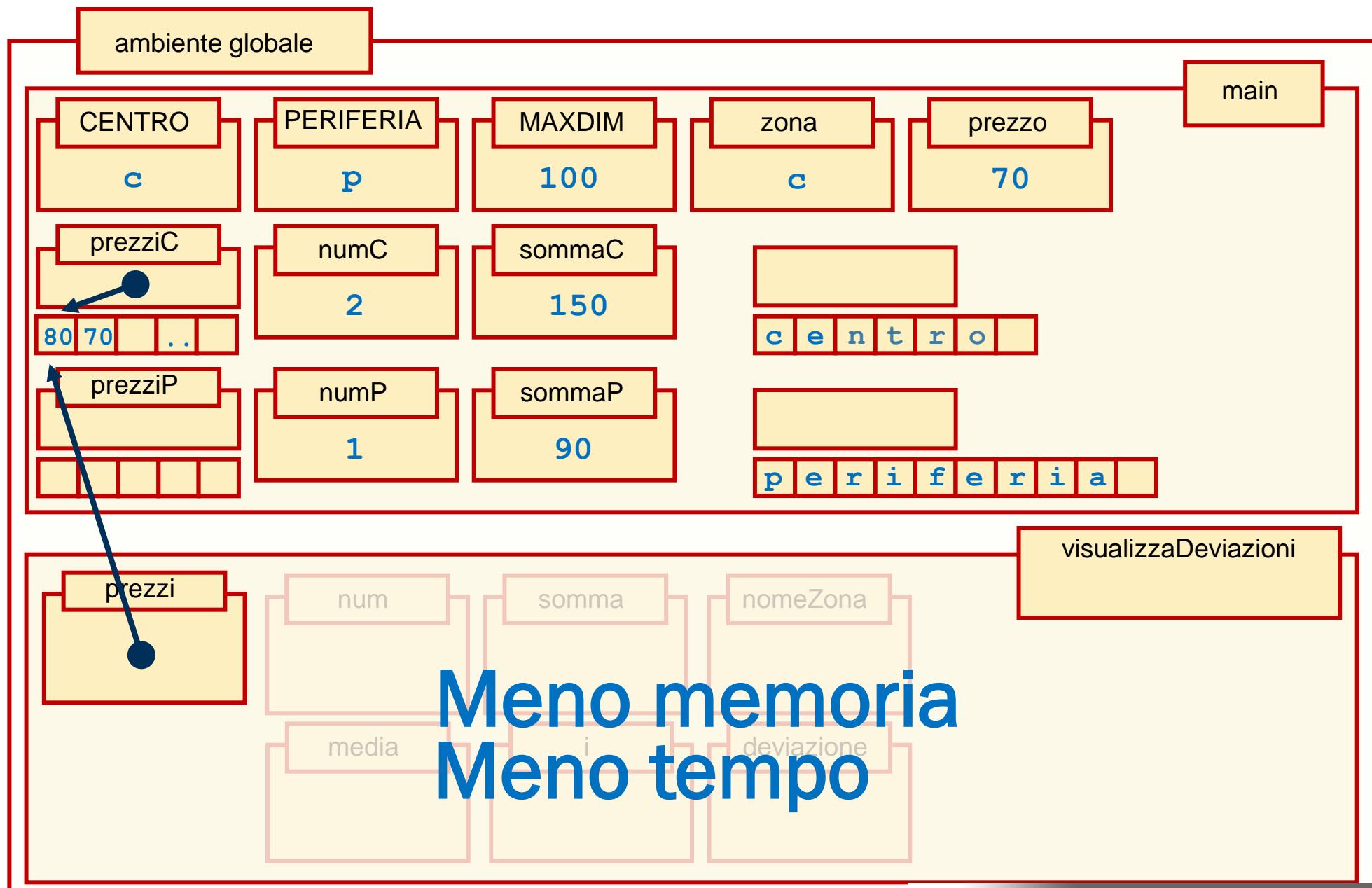




Efficienza







ambiente globale

main

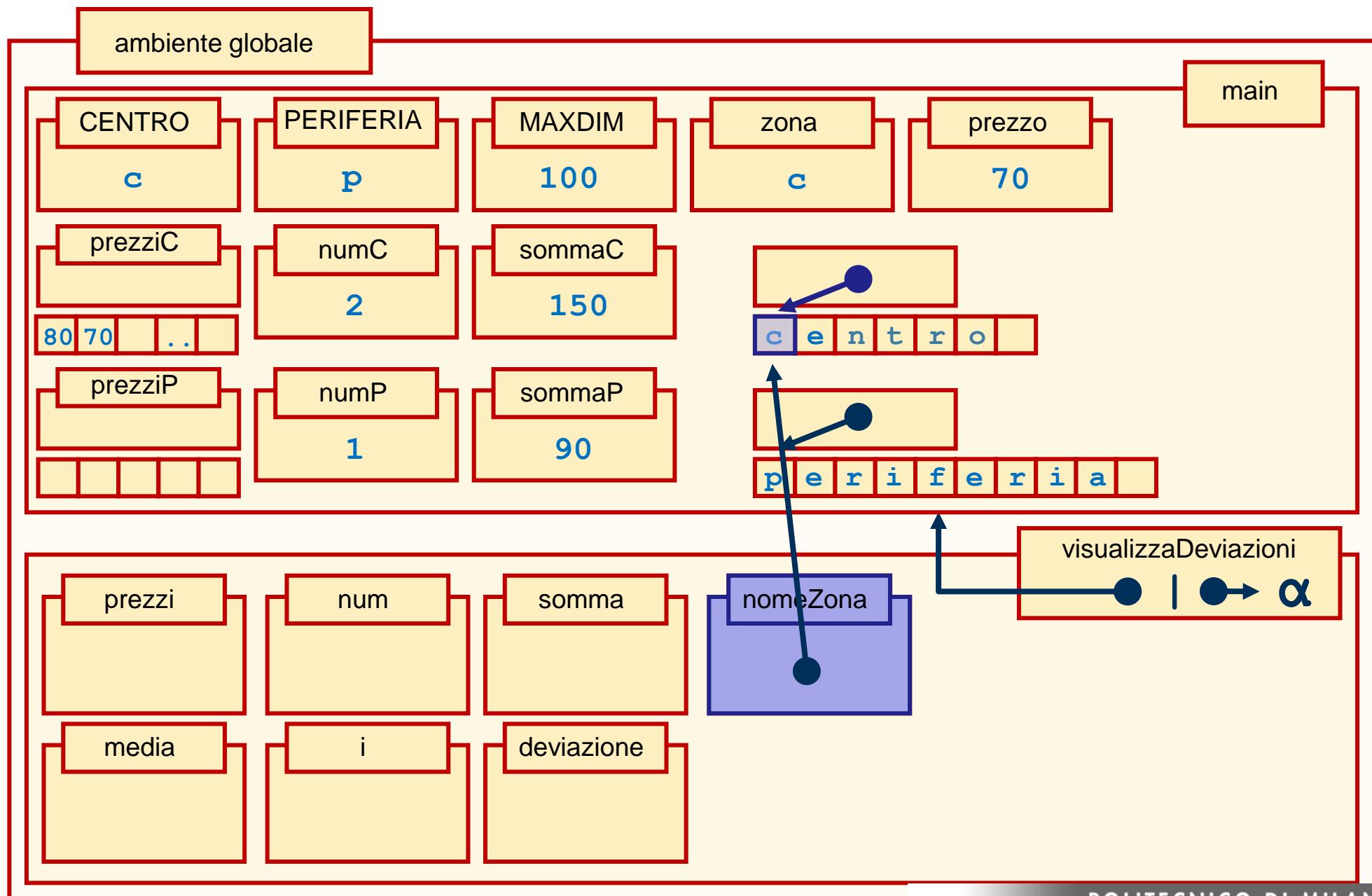
```
#include <iostream>
using namespace std;
#include <iomanip.h>

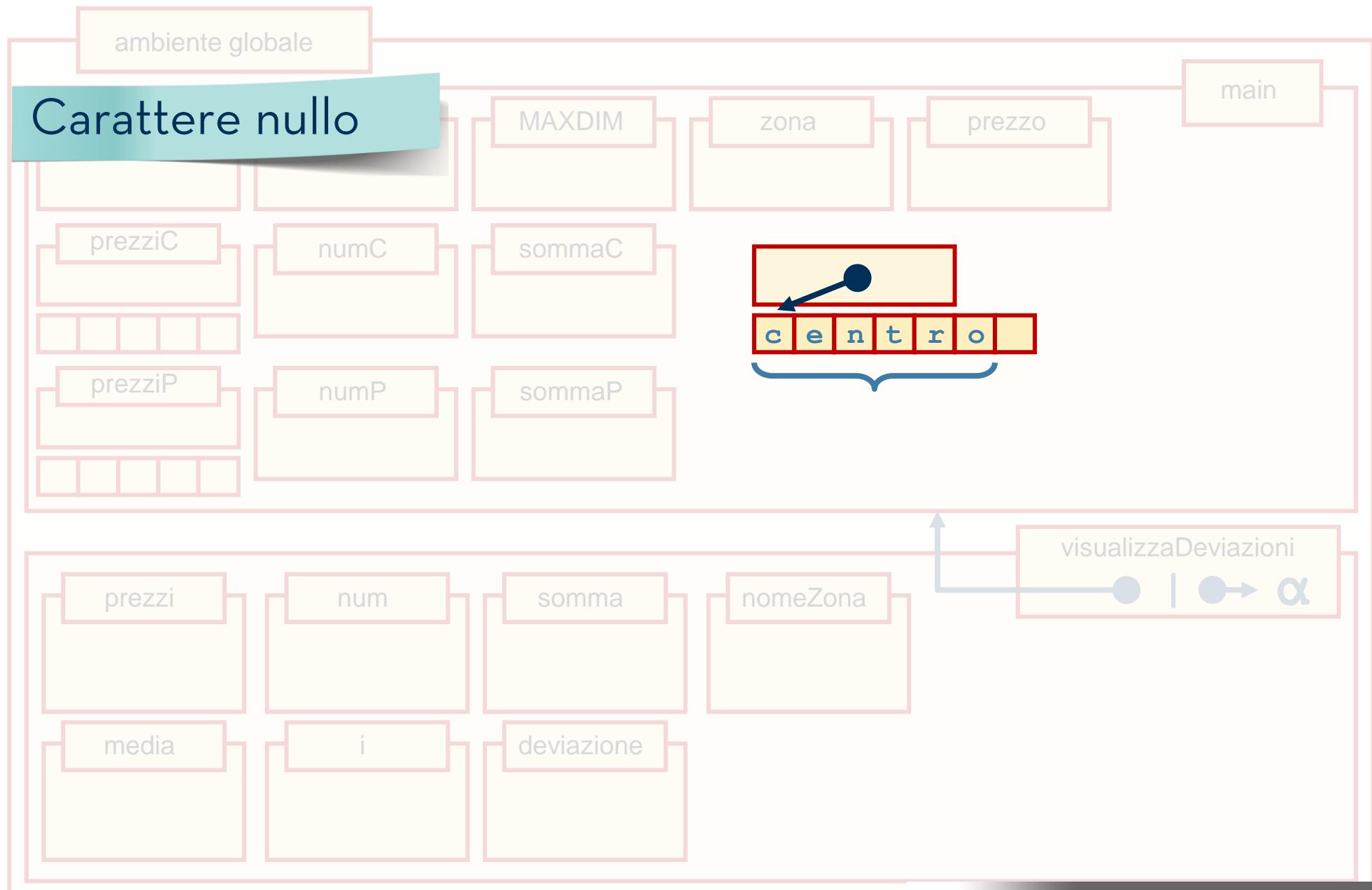
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
```

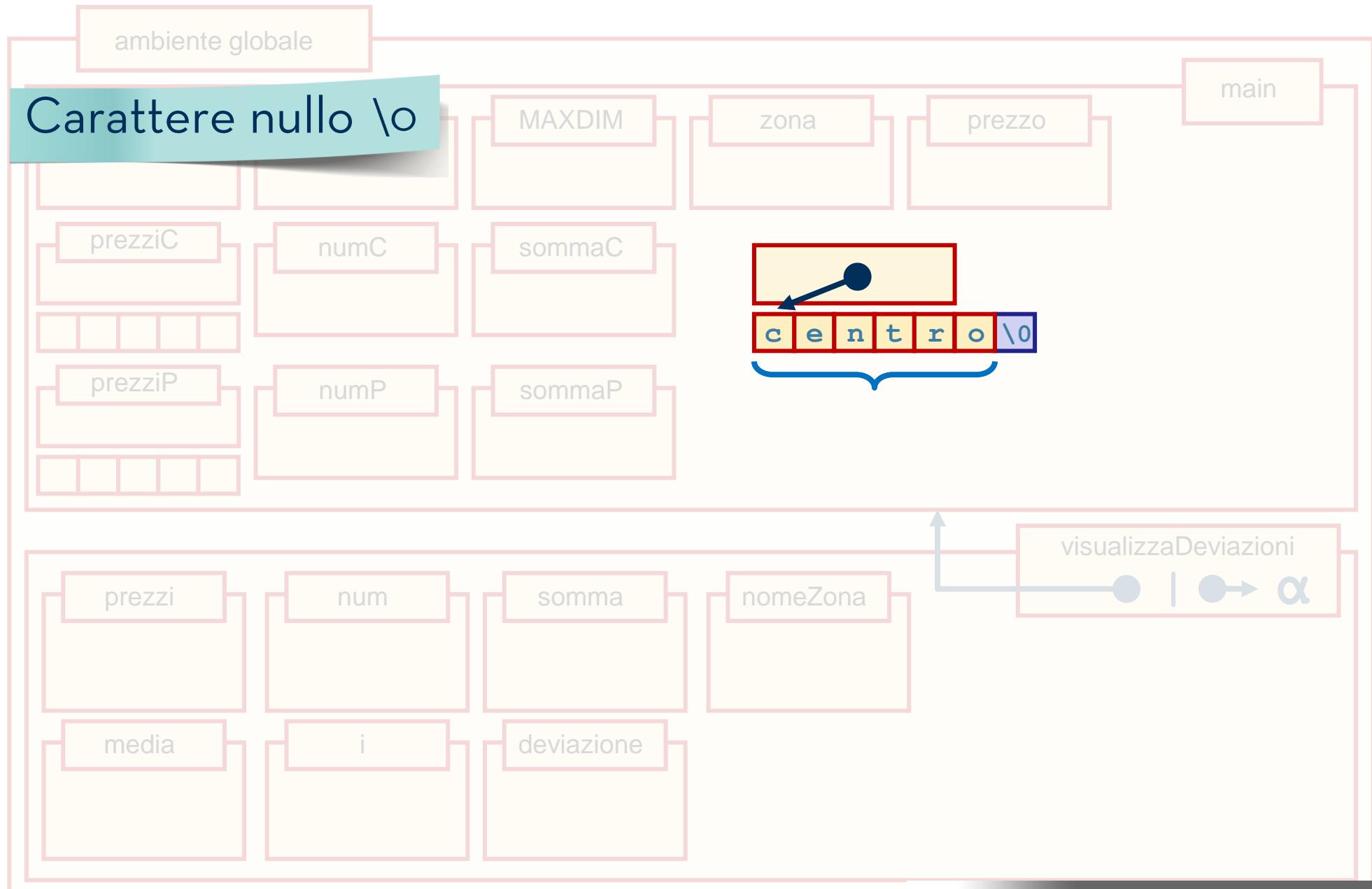
```
...
visualizzaDeviazioni ("centro", sommaC, numC, prezziC);
visualizzaDeviazioni ("periferia", sommaP, numP, prezziP);
}
```

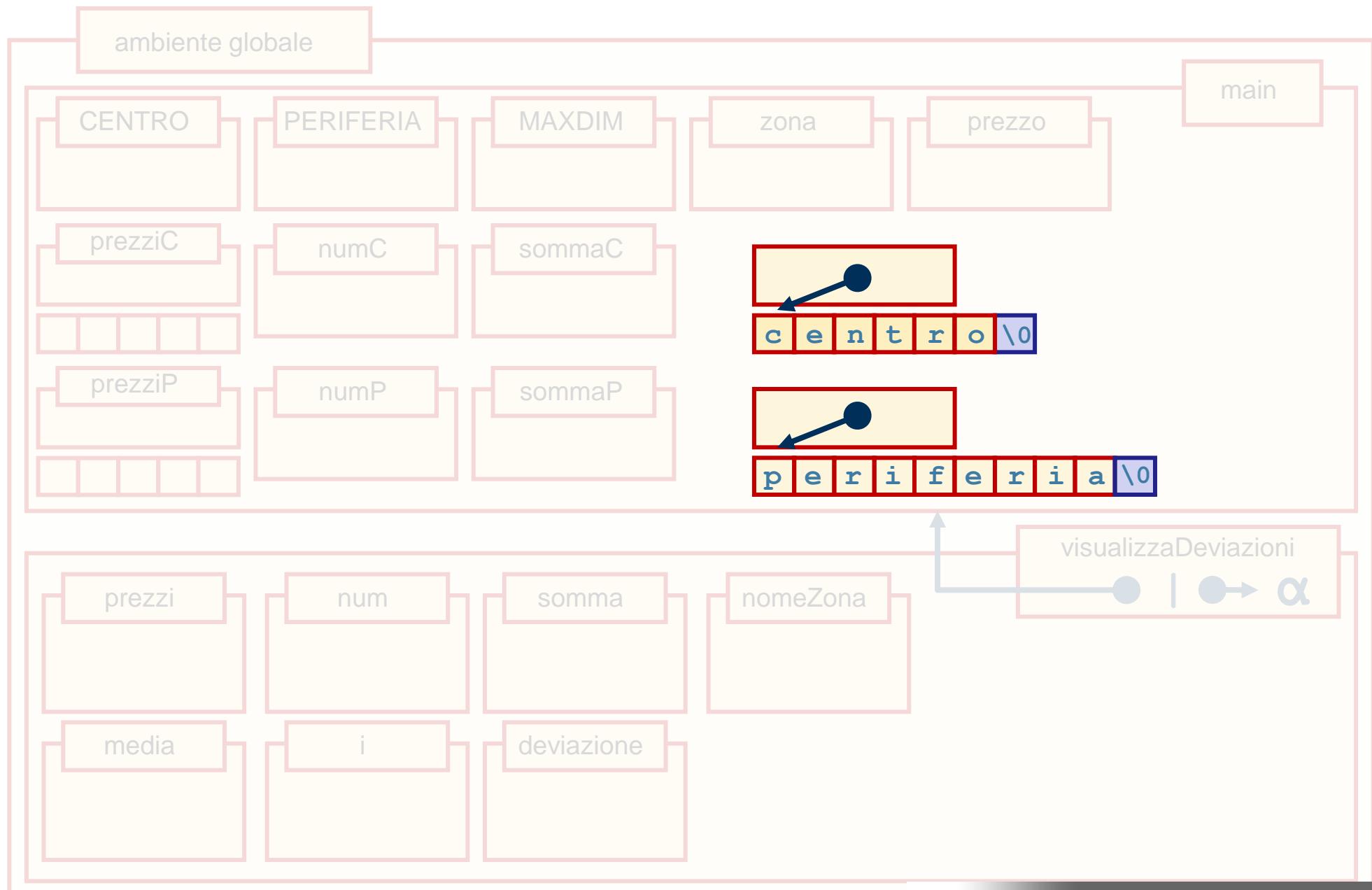
visualizzaDeviazioni

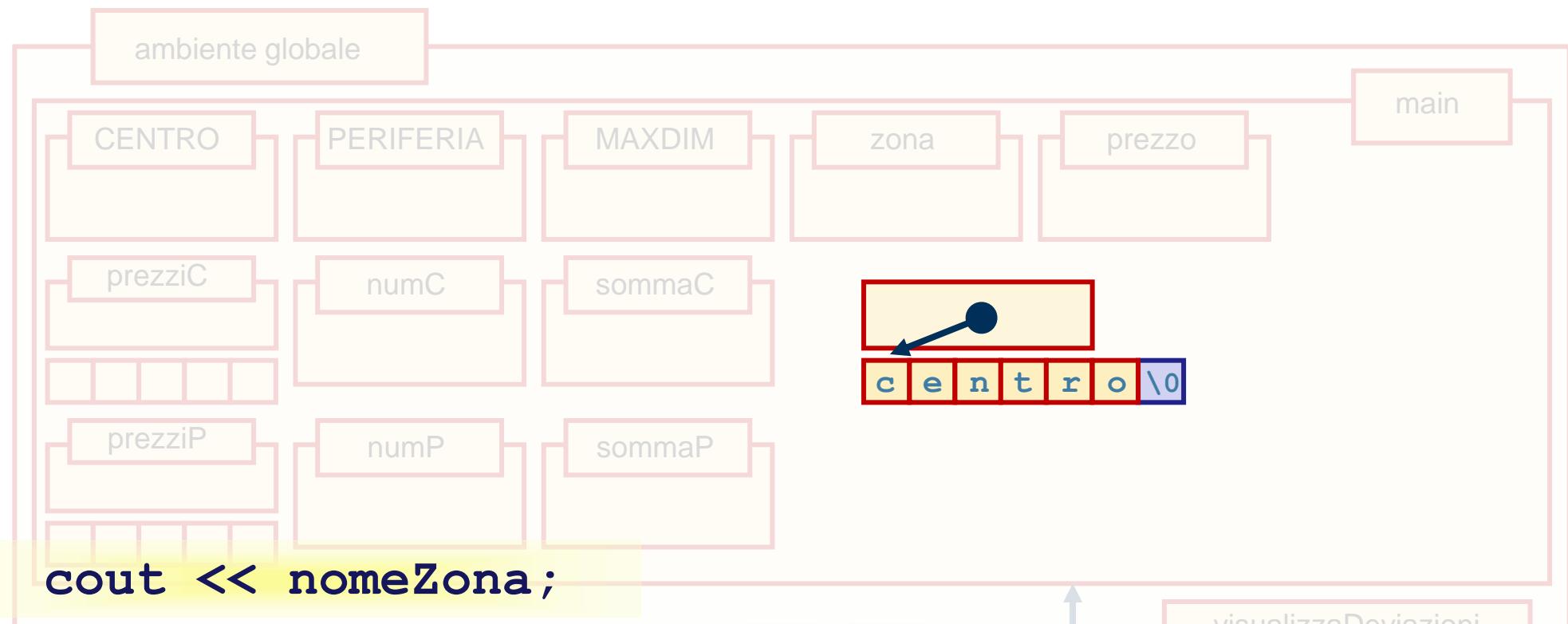
| ↗ α



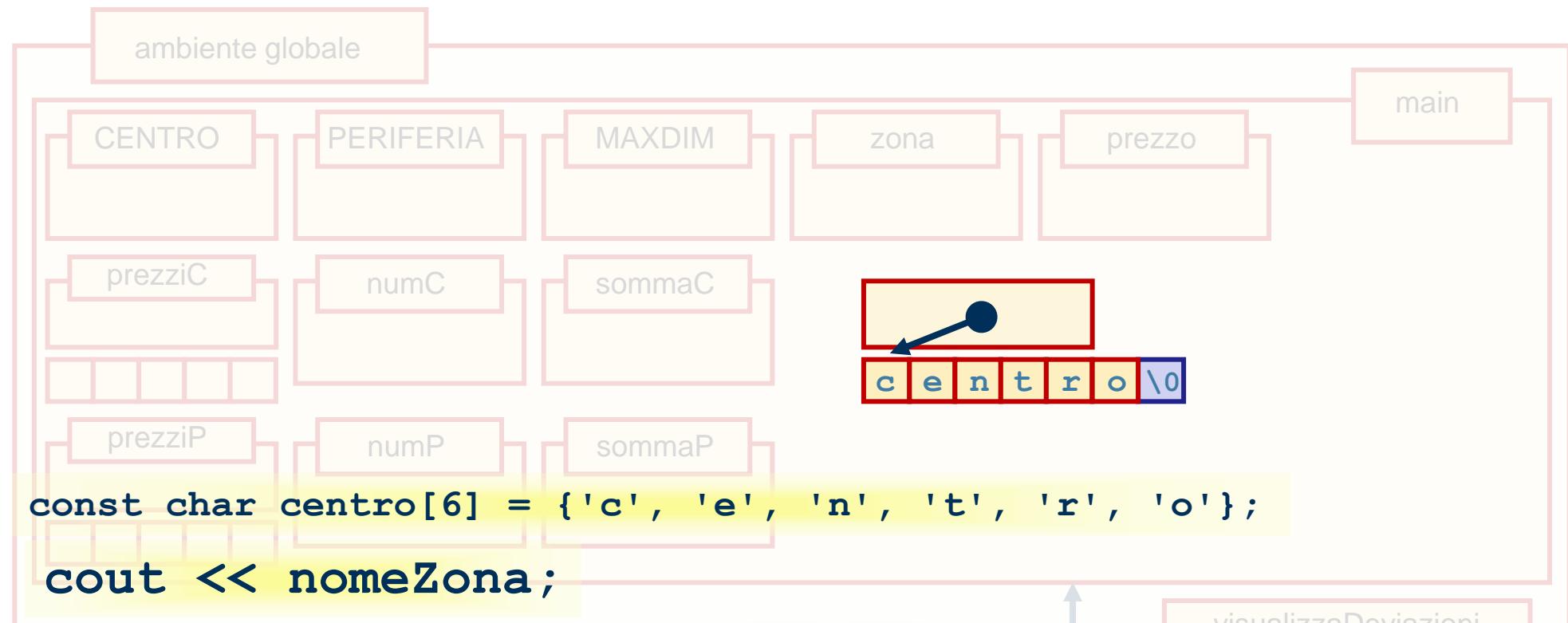


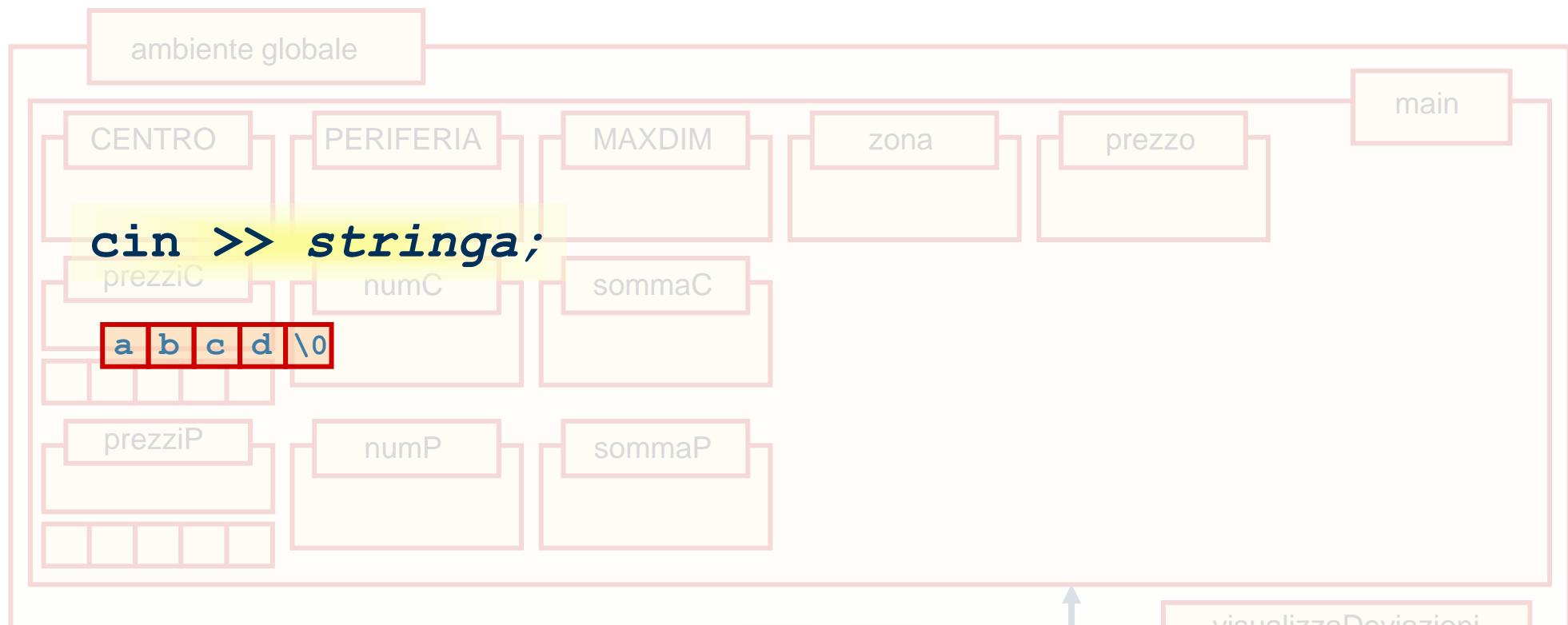




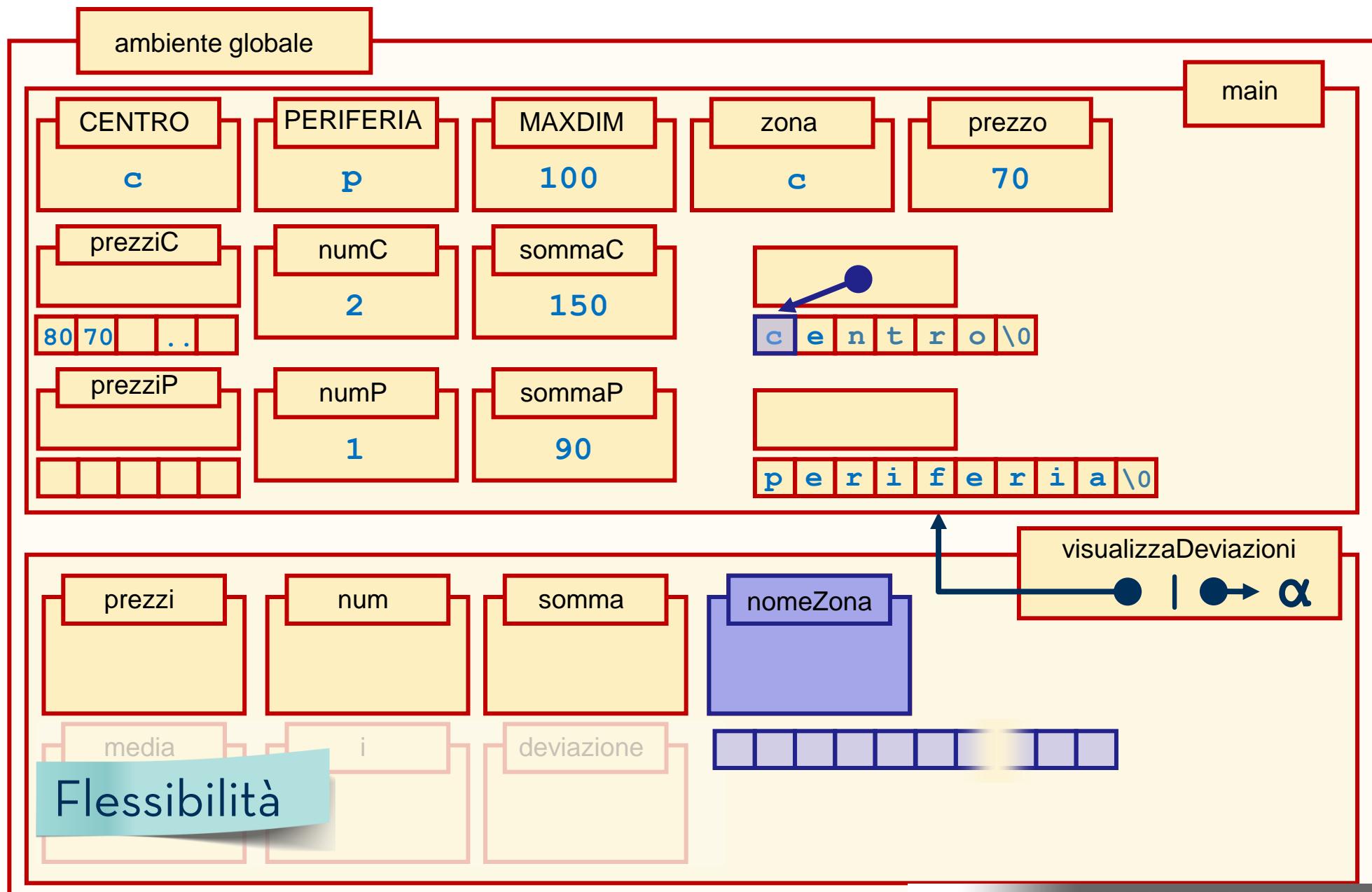


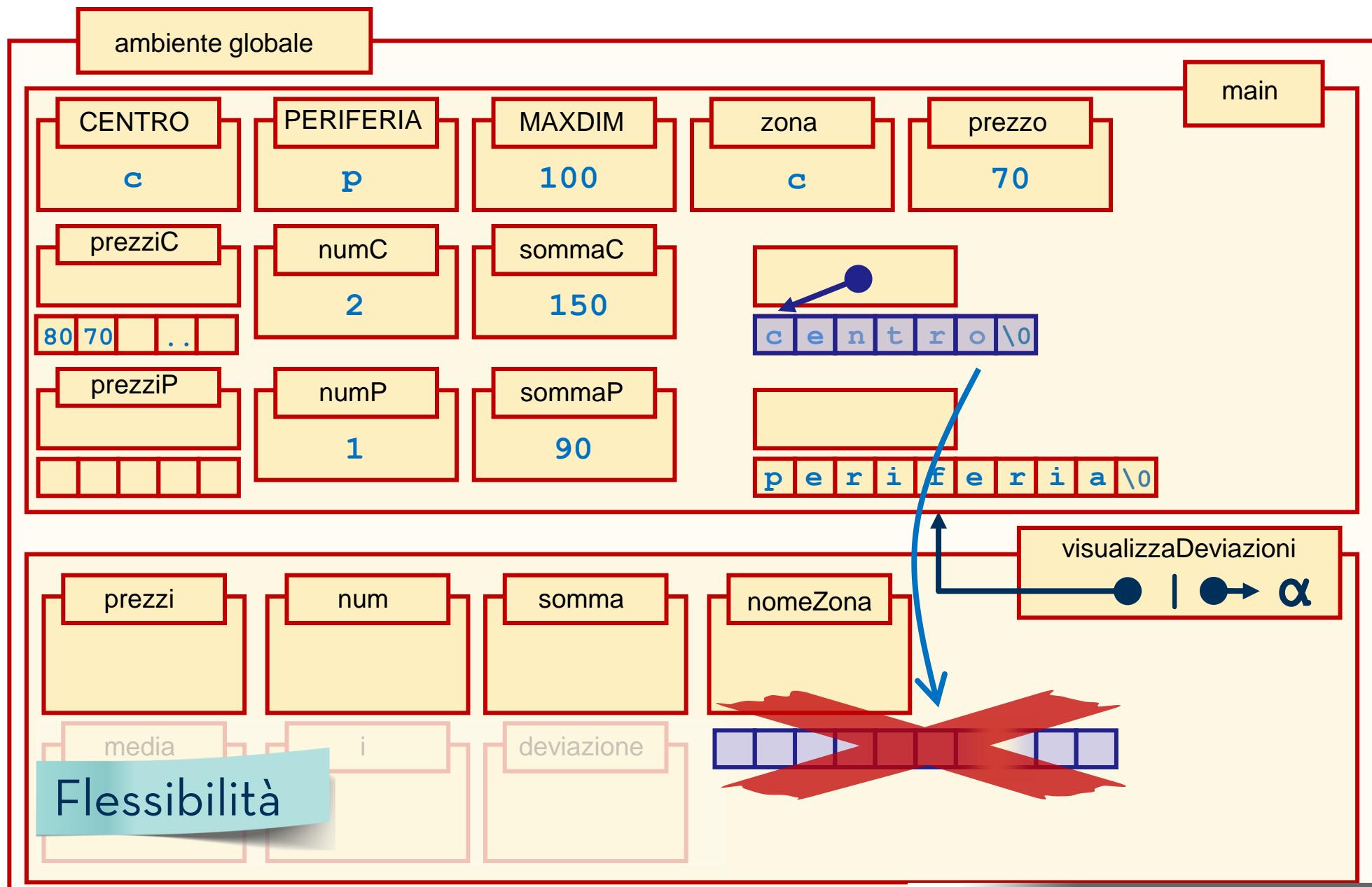
centro

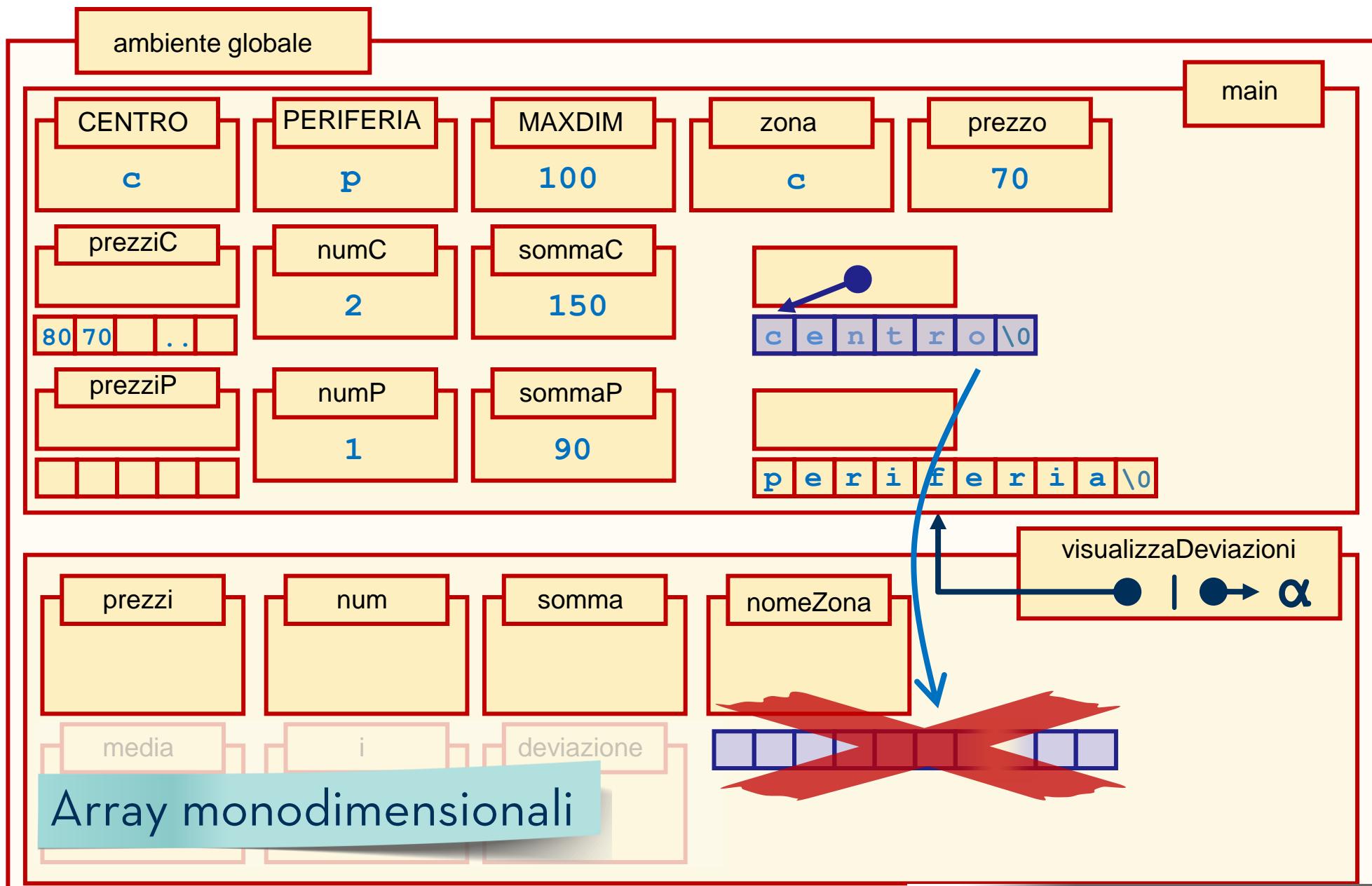




abcd ↲







ambiente globale

main

usare indici congruenti con
gli array su cui operano

visualizzaDeviazioni

 α

ambiente globale

main

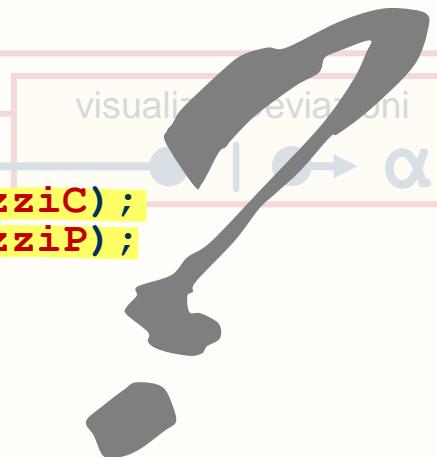
```
#include <iostream>
using namespace std;
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(10) << nomeZona << " : ";
cout << setprecision(2) << media << endl;
cout << setprecision(2) << deviazione << endl;
for (i = 0; i < num; i++)
cout << prezzi[i] << " ";
cout << endl;
}
```



All'interno
della dimensione
corretta

```
...
visualizzaDeviazioni ("centro", sommaC, numC, prezziC);
visualizzaDeviazioni ("periferia", sommaP, numP, prezziP);
}
```



ambiente globale

main

```
#include <iostream>
using namespace std;
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
```



```
...
```

```
visualizzaDeviazioni ("centro",     sommaC, numC, prezziC);
visualizzaDeviazioni ("periferia",   sommaP, numP, prezziP);
}
```



ambiente globale

Alternativa

```
#include <iostream>
using namespace std;
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
```

main

Usare un
elemento terminatore
nell'array



```
#include <iostream>
using namespace std;
#include <iomanip.h>

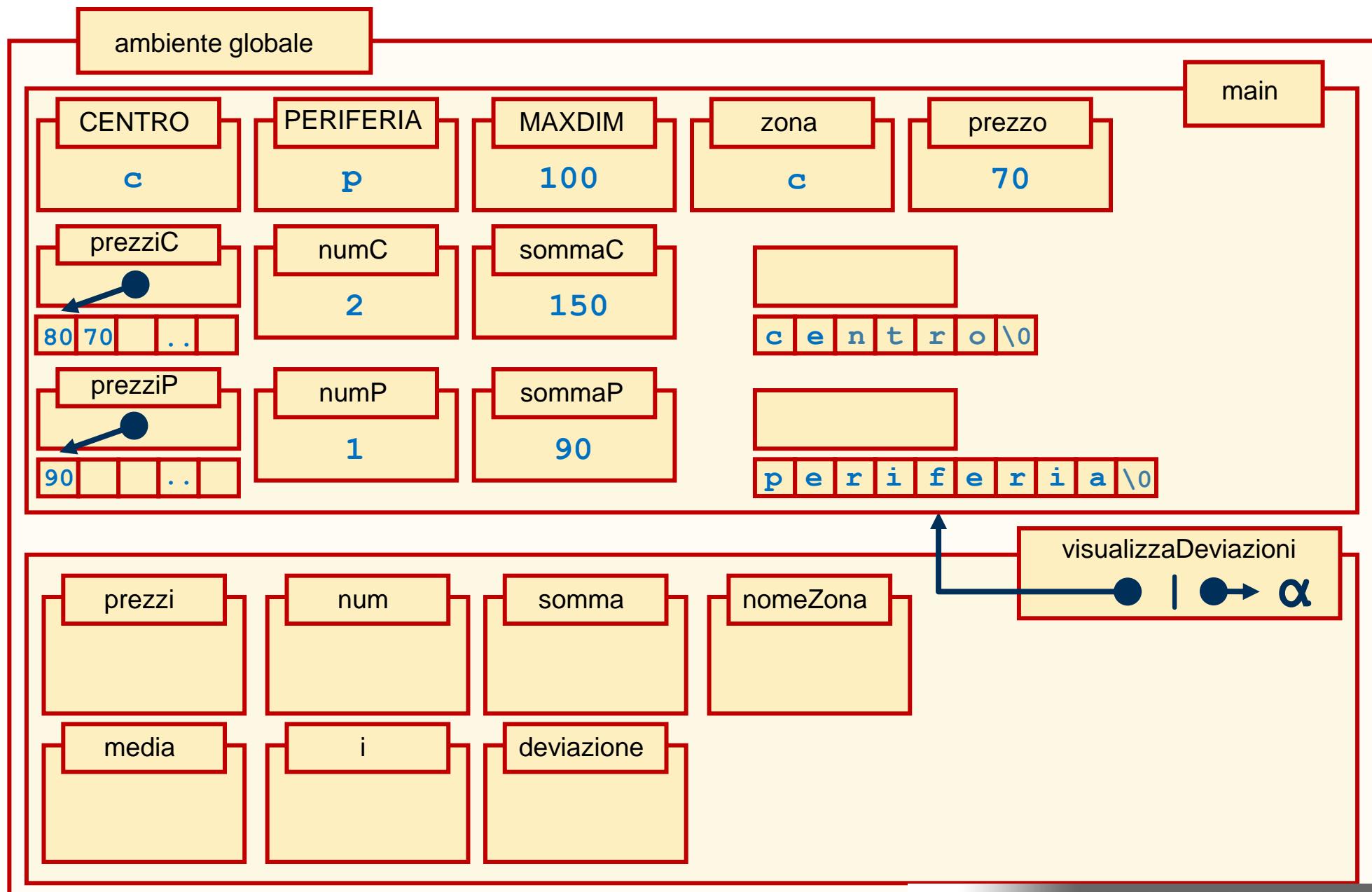
void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
}

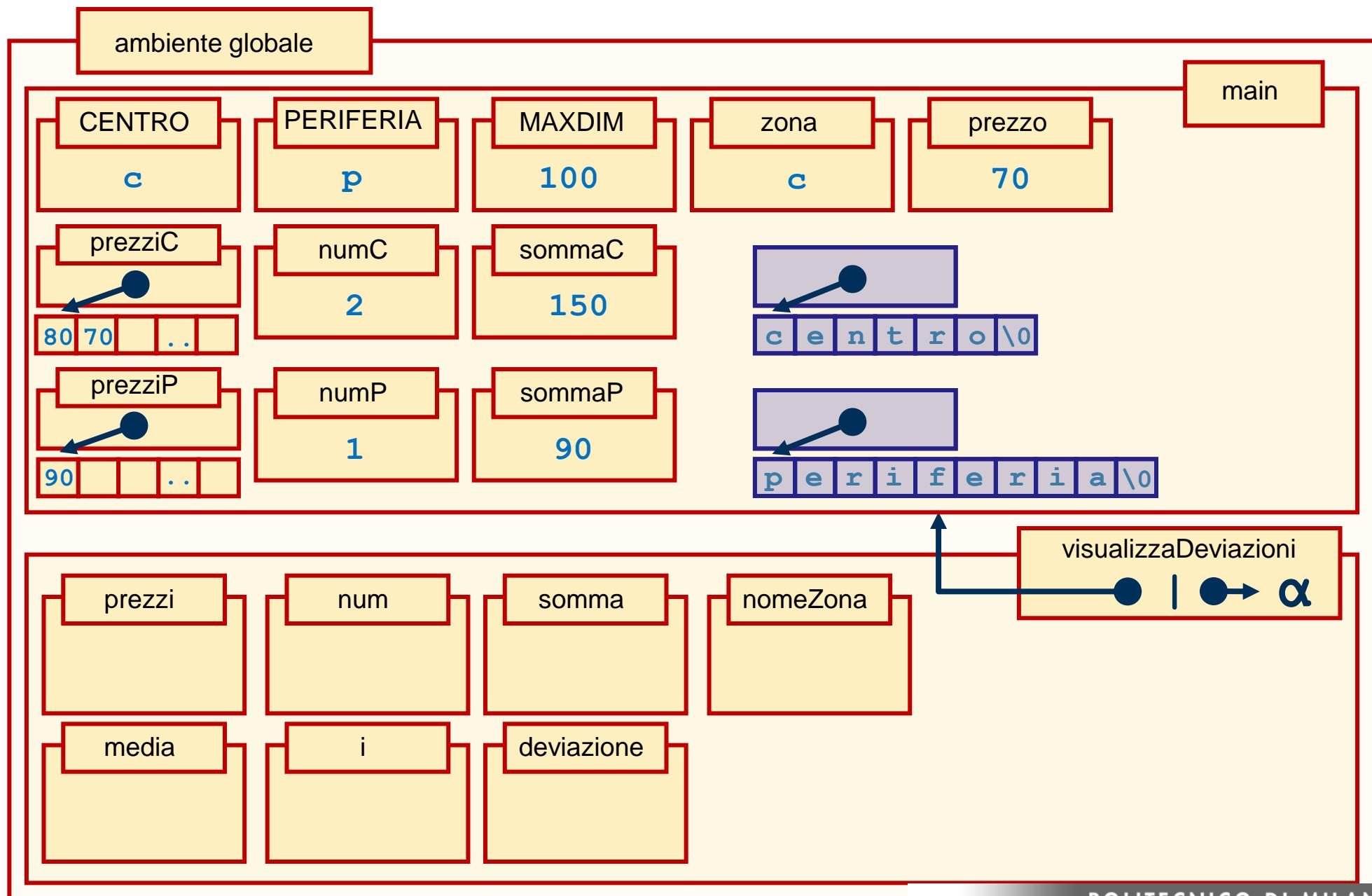
int main()
const int MAXDIM = 100;
```

```
#include <iostream>
using namespace std;
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float) somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}

int main()
{
const int MAXDIM = 100;
```





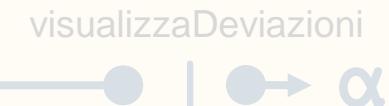
ambiente globale

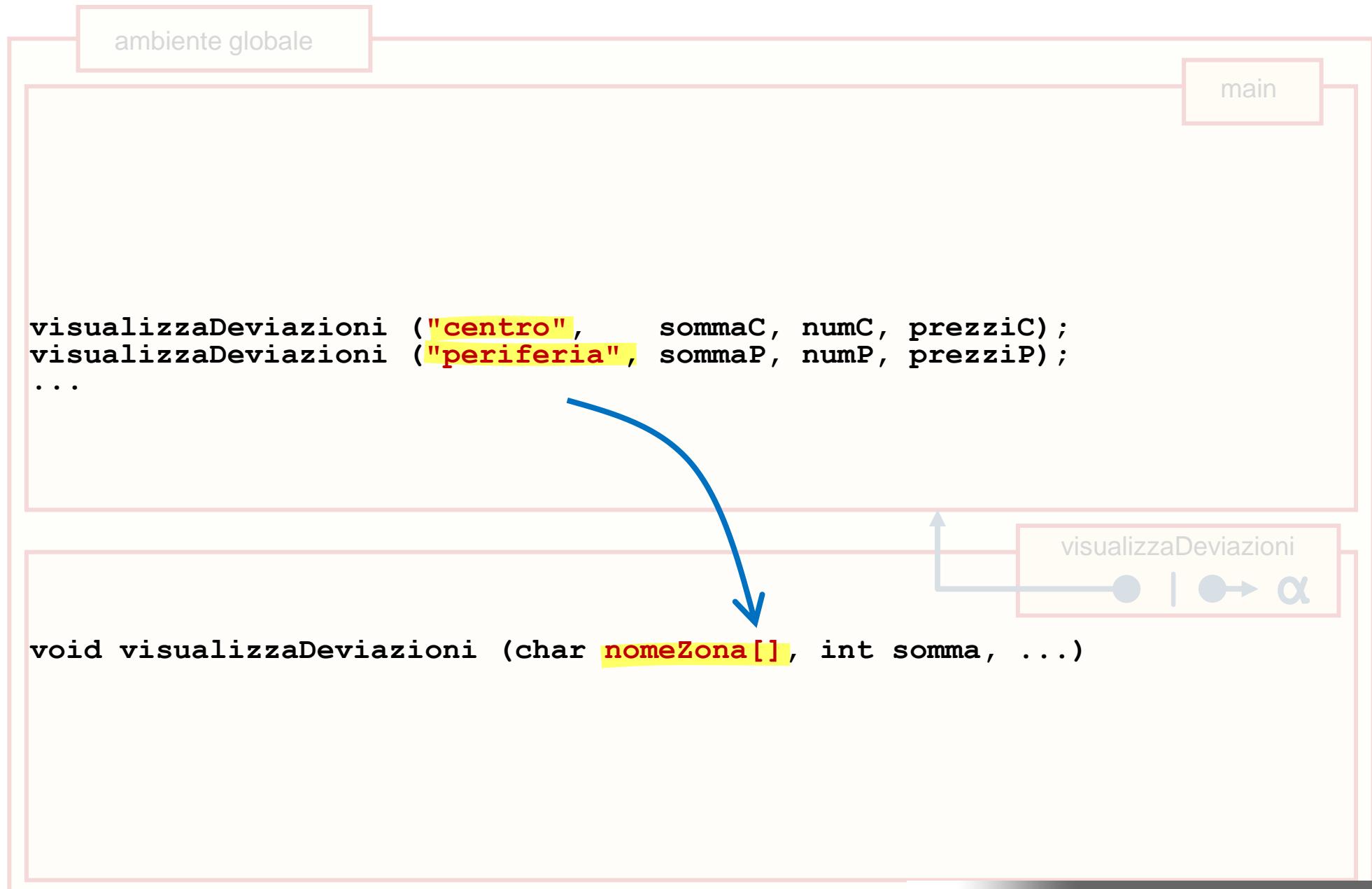
main

```
...
const char nomeCentro[7] = "centro";
const char nomePeriferia[10] = "periferia";
...
```

```
visualizzaDeviazioni (nomeCentro, sommaC, numC, prezziC);
visualizzaDeviazioni (nomePeriferia, sommaP, numP, prezziP);
...
```

```
void visualizzaDeviazioni (char nomeZona[], int somma, ...)
```





```
// Analisi del prezzo di un prodotto in una città divisa in due zone.  
// Versione con uso di sottoprogrammi.  
#include <iostream>  
using namespace std;  
#include <iomanip.h>  
  
void visualizzaDeviazioni  
    (char nomeZona[], int somma, int num, int prezzi[])  
{ float media, deviazione;  
  int i;  
  if (num != 0) // se c'è almeno una rilevazione  
  { cout << endl << setw(20)  
    << "rilevazioni zona " // crea intestazione  
    << setw(10) << nomeZona << endl << endl // tabella per zona  
    << setw(20) << "prezzi rilevati" // di nome nomeZona  
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;  
  media = (float) somma / num; // ne calcola il prezzo medio  
  for (i = 0; i < num; i++) // per ogni rilevazione, stampa;  
  { deviazione = prezzi[i] - media;  
    cout << setw(20) << prezzi[i] // una riga contenente il prezzo  
    << setw(40) << setprecision(2) // rilevato e la deviazione  
    << setiosflags (ios::fixed | ios::showpoint) // rispetto  
    << deviazione // alla media, in formato fisso, con il  
    << endl; // punto decimale e due cifre decimali  
  }  
}  
  
int main()  
{ const char CENTRO = 'c', PERIFERIA = 'p';  
  const int MAXDIM = 100;  
  char zona; // zona della rilevazione corrente
```



20 quartieri

```
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
else
    cout << "zona non corretta: inserire nuovo valore";
}
visualizzaDeviazioni("quartiere1", somma1, num1, prezzi1);
visualizzaDeviazioni("quartiere2", somma2, num2, prezzi2);
visualizzaDeviazioni("quartiere3", somma3, num3, prezzi3);
...
visualizzaDeviazioni("quartiere19", somma19, num19, prezzi19);
visualizzaDeviazioni("quartiere20", somma20, num20, prezzi20);
}
```

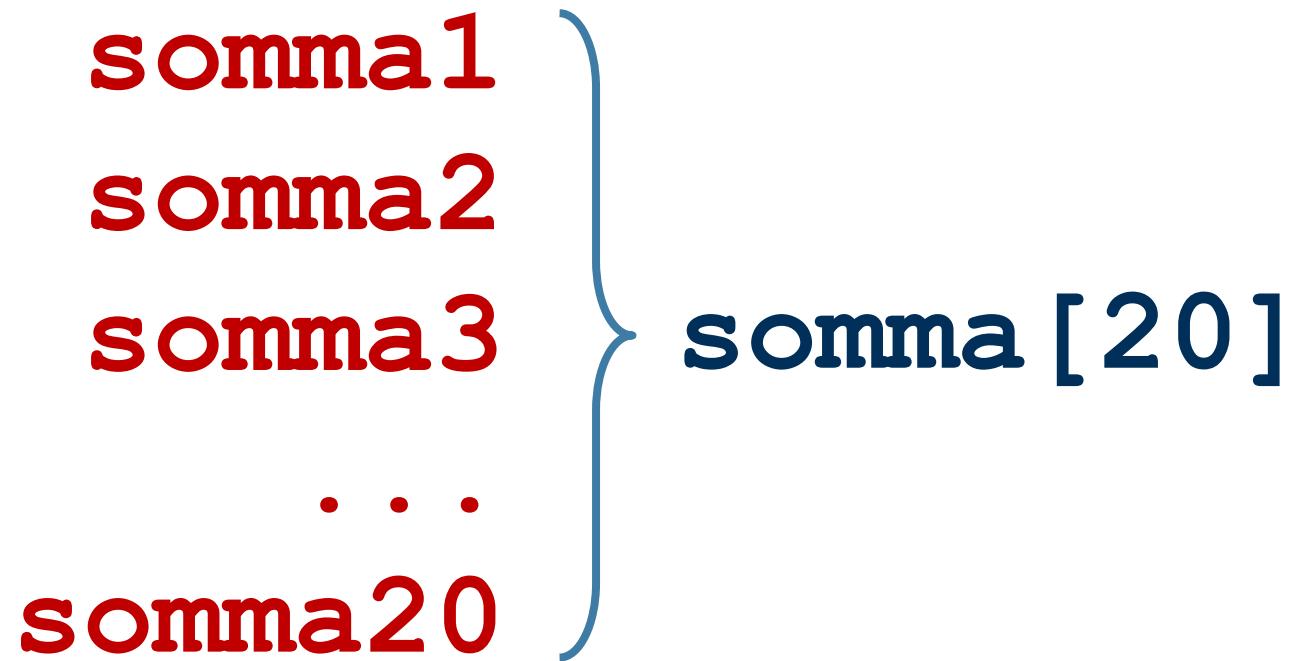
```
int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona; // zona della rilevazione corrente
int prezzo, // prezzo della rilevazione corrente
prezziC[MAXDIM], prezziP[MAXDIM], // prezzi rilevati nelle due zone
numC, numP, sommaC, sommaP; // numero è somma dei prezzi delle due zone

cout << setw(53) << "acquisizione rilevazioni" << endl;

numC = 0; numP = 0; sommaC = 0; sommaP = 0;
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "zona della nuova rilevazione: "; cin >> zona;
if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
|| (zona == PERIFERIA && numP >= MAXDIM))
break;
if (zona == CENTRO || zona == PERIFERIA)
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
if (zona == CENTRO)
{ sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
if (zona == PERIFERIA)
{ sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
}
else
cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("quartiere1", somma1, num1, prezzi1);
visualizzaDeviazioni("quartiere2", somma2, num2, prezzi2);
```

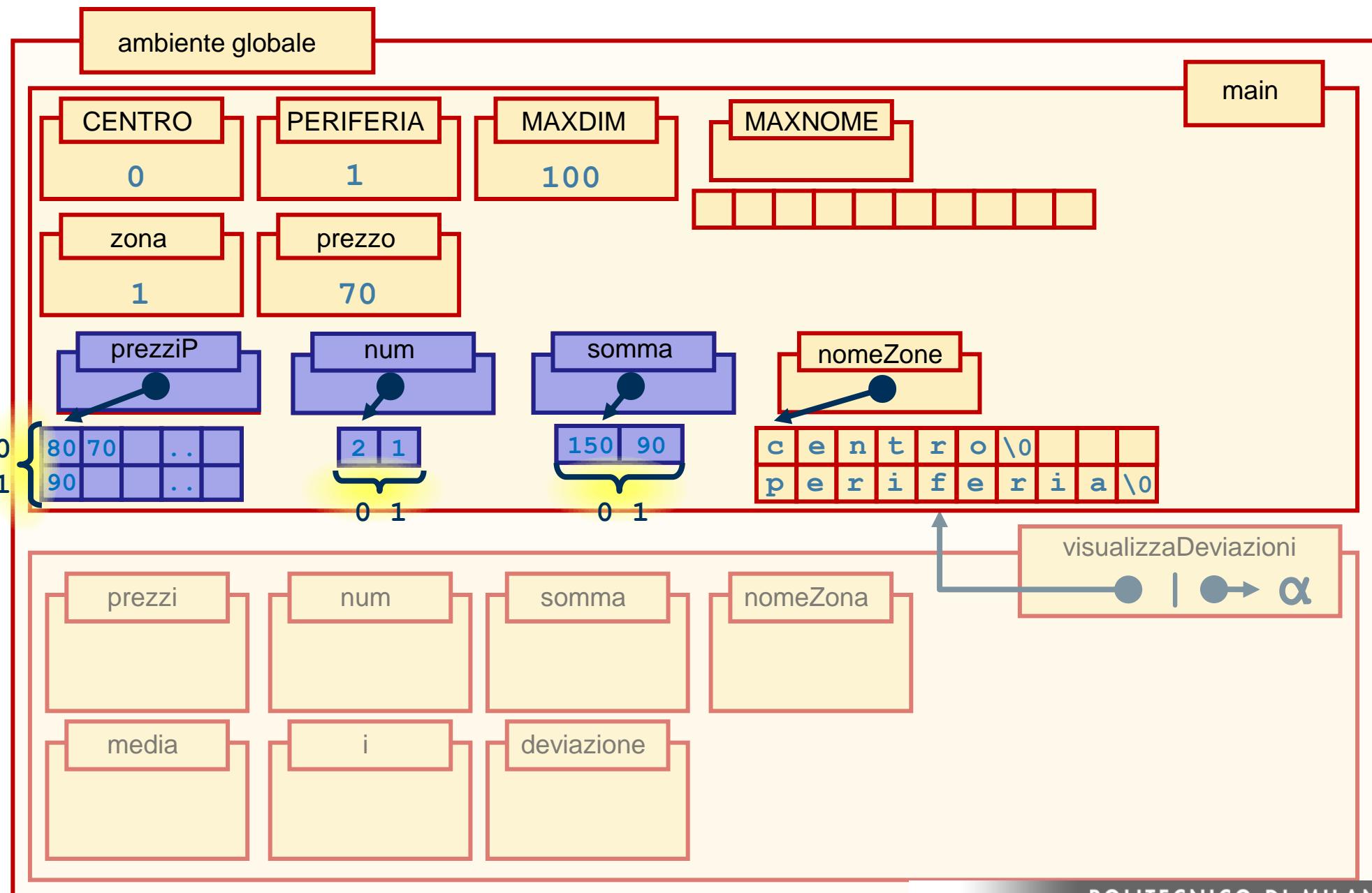
```
if (zona == quartiere1 || zona == quartiere2  
    || zona == quartiere3 ... || zona == quartiere20)  
  
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;  
  if (zona == quartiere1)  
    { somma1 += prezzo; prezzi1[num1] = prezzo; num1++; }  
  if (zona == quartiere2)  
    { somma2 += prezzo; prezzi2[num2] = prezzo; num2++; }  
  if (zona == quartiere3)  
    { somma3 += prezzo; prezzi3[num3] = prezzo; num3++; }  
  ...  
  if (zona == quartiere20)  
    { somma20 += prezzo; prezzi20[num20] = prezzo; num20++; }  
}  
else  
  cout << "zona non corretta: inserire nuovo valore";  
}
```

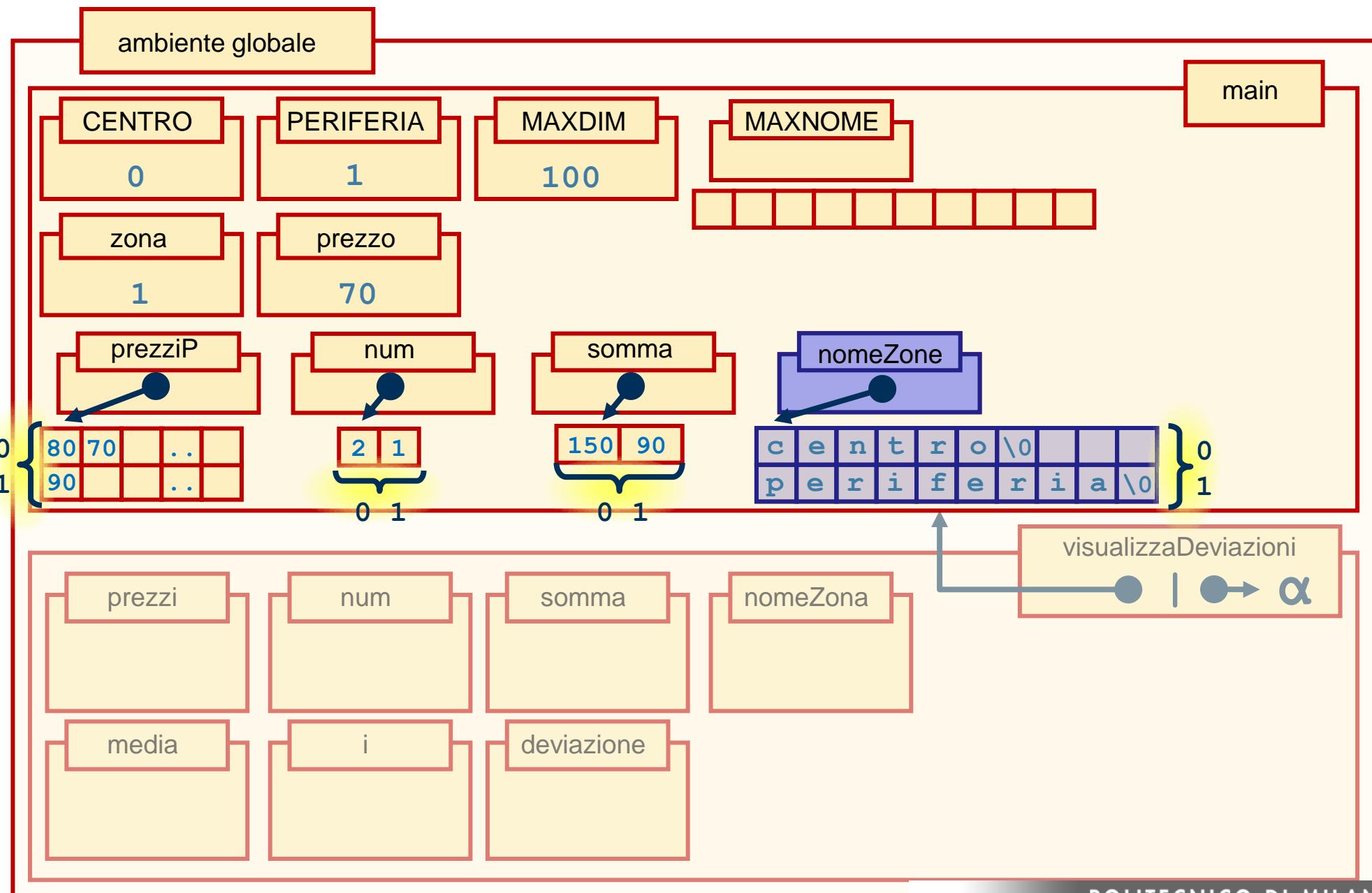


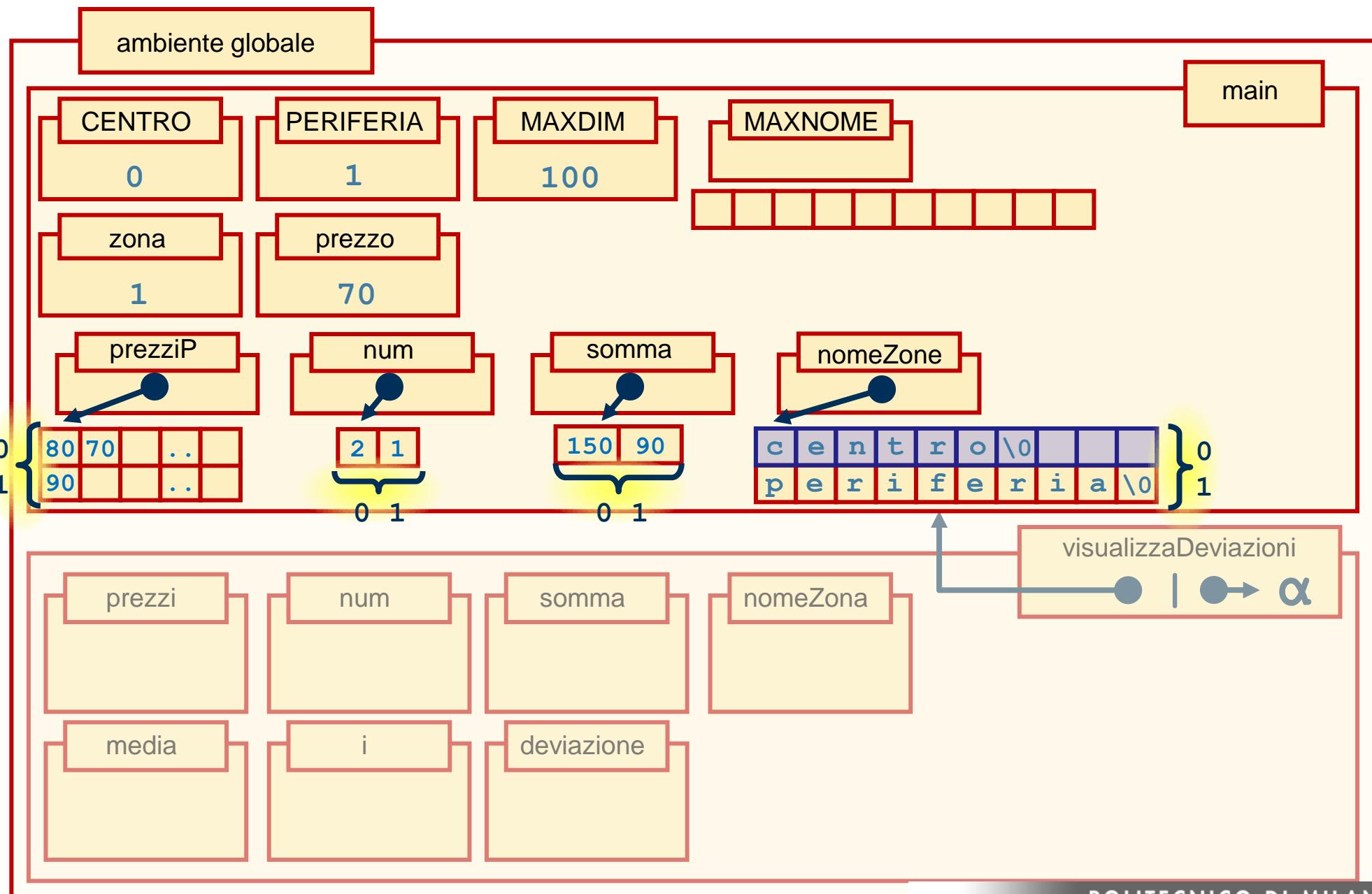
somma [numeroZone] sommaC, sommaP

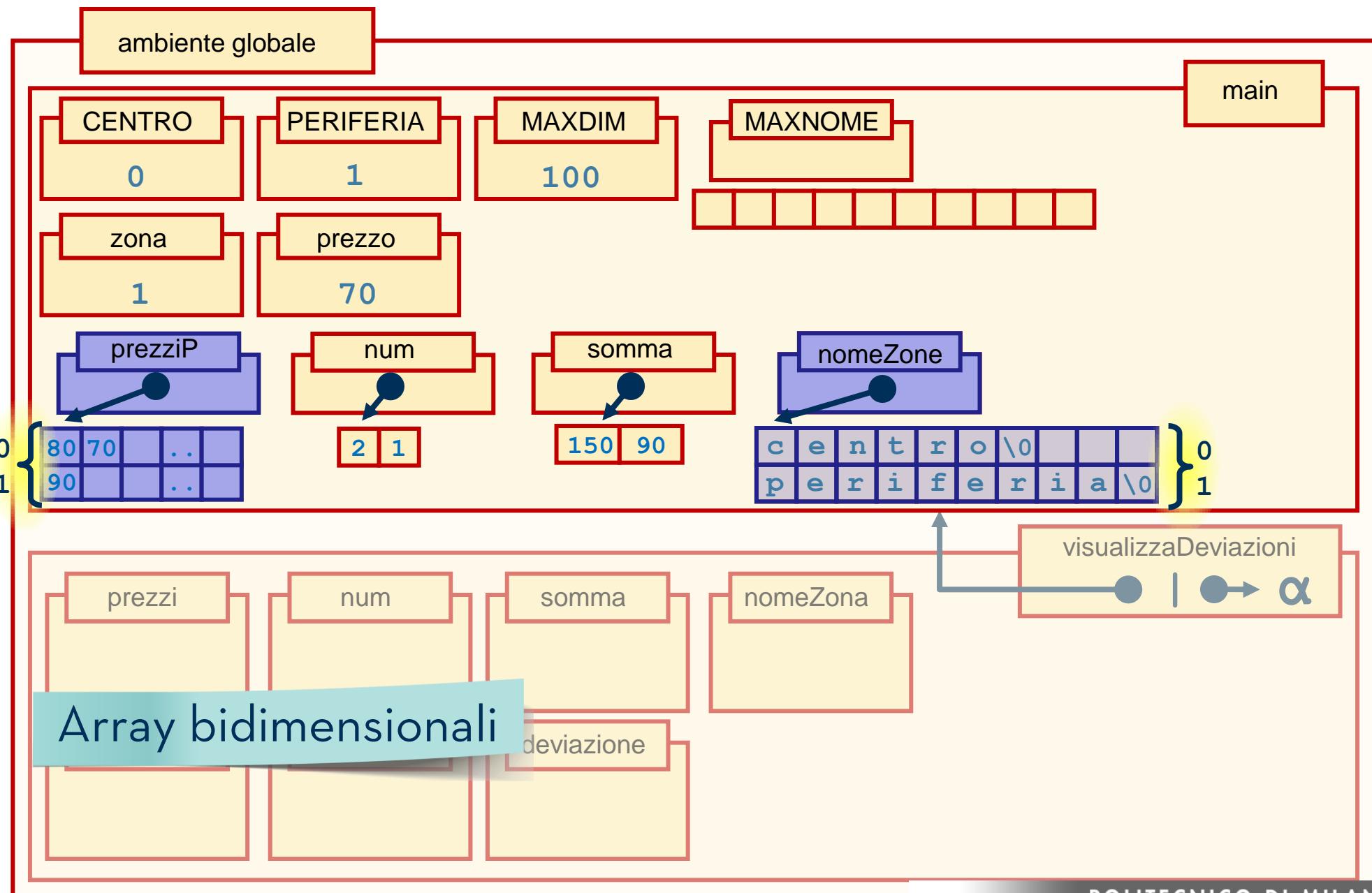
prezzi [numeroZone] prezziC, prezziP

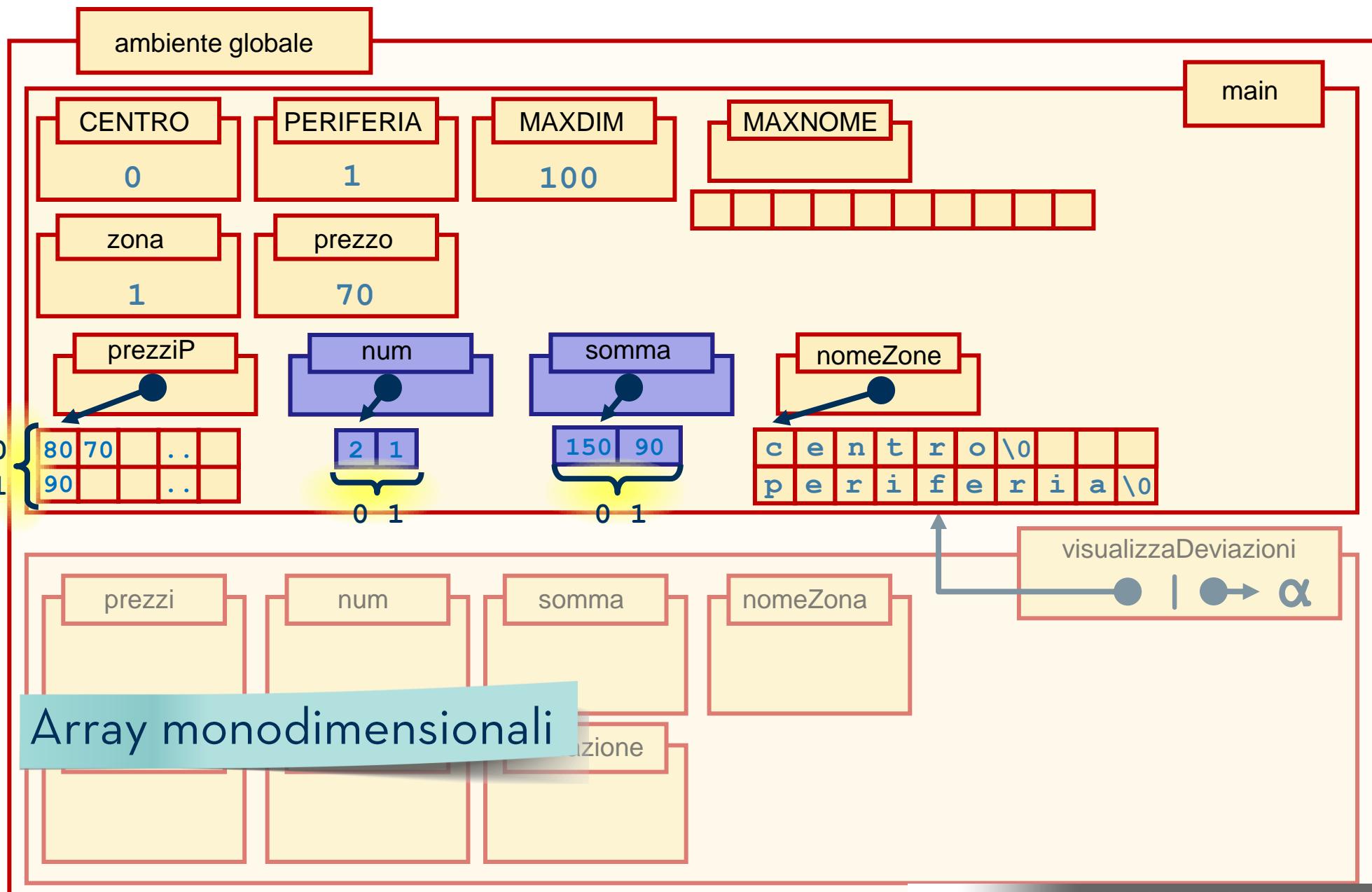
num [numeroZone] numC, numP

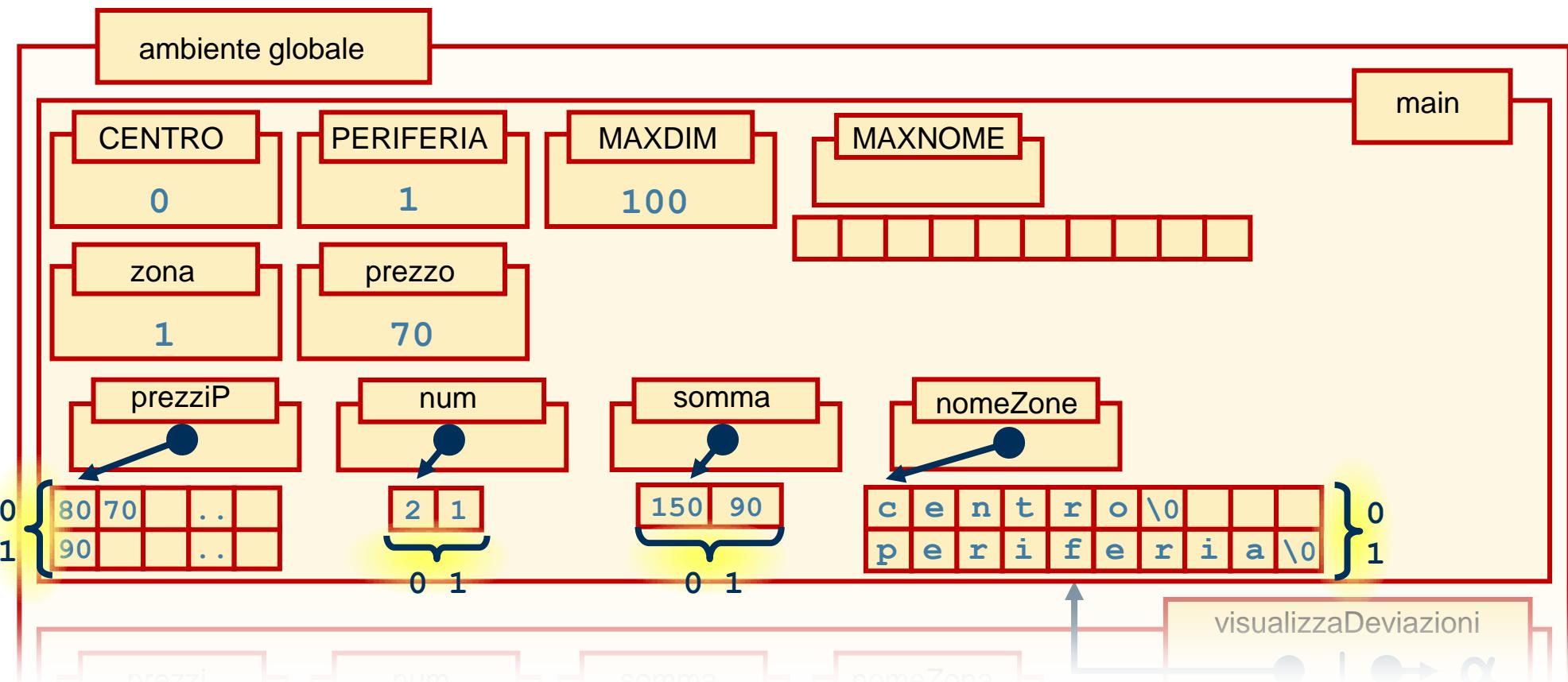




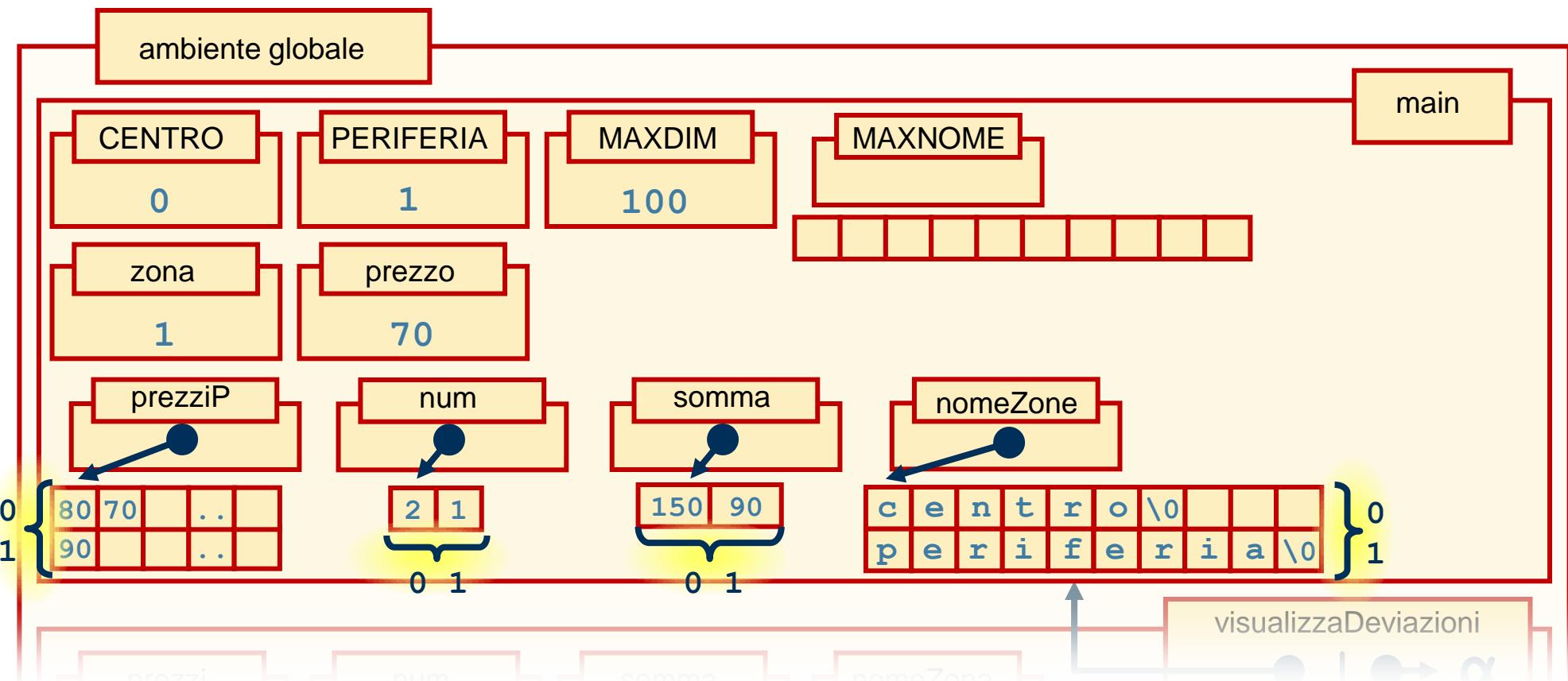








```
const char CENTRO = 'c', PERIFERIA = 'p';
```



```
const int CENTRO = 0, PERIFERIA = 1;
```

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{
    float media, deviazione;
    int i;
    if (num != 0)
        { cout << endl << setw(20)
          << "rilevazioni zona "
          << setw(10) << nomeZona << endl << endl
          << setw(20) << "prezzi rilevati"
          << setw(40) << "deviazione rispetto al prezzo medio"
          << endl;
        media = (float)somma/num;
        for (i = 0; i < num; i++)
            { deviazione = prezzi[i] - media;
              cout << setw(20)
                  << setw(40) << setprecision(2)
                  << setiosflags (ios::fixed | ios::showpoint)
                  << deviazione
                  << endl;
            }
        }

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
  const int MAXDIM = 100;
  char zona;
```

2zone

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float)somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
}

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona;
```

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
  int i;
  if (num != 0)
    { cout << endl << setw(20)
      << "rilevazioni zona "
      << setw(10) << nomeZona << endl << endl
      << setw(20) << "prezzi rilevati"
      << setw(40) << "deviazione rispetto al prezzo medio"
      << endl;
    media = (float)somma / num;
    for (i = 0; i < num; i++)
      { deviazione = prezzi[i] - media;
        cout << setw(20) << prezzi[i]
        << setw(40) << setprecision(2)
        << setiosflags (ios::fixed | ios::showpoint)
        << deviazione
        << endl;
      }
    }
}

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
  const int MAXDIM = 100;
  char zona;
```

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float)somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
}

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona;
```

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float)somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
}

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona;
```

```
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float)somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}

int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona;
```

```
int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
const int MAXDIM = 100;
char zona;
int prezzo,
prezziC[MAXDIM], prezziP[MAXDIM],
numC,numP,sommaC,sommaP;

cout << setw(53) << "acquisizione    rilevazioni" << endl;

numC = 0; numP = 0; sommaC = 0; sommaP = 0;
while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
|| (zona == PERIFERIA && numP >= MAXDIM) )
break;
if (zona == CENTRO || zona == PERIFERIA)
{ cout << "prezzo della nuova rilevazione: ";
cin >> prezzo;
if (zona == CENTRO)
{ sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
if (zona == PERIFERIA)
{ sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
}
else
cout << "zona non corretta: inserire nuovo valore";
}
```

```
int main()
{ const char CENTRO = 'c', PERIFERIA = 'p';
  const int MAXDIM = 100;
  char zona;
  int prezzo,
      prezziC[MAXDIM], prezziP[MAXDIM],
      numC, numP, sommaC, sommaP;

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
        break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  char zona;
  int prezzo,
      prezziC[MAXDIM], prezziP[MAXDIM],
      numC, numP, sommaC, sommaP;

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
        break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona,
    prezzo,
    prezziC[MAXDIM], prezziP[MAXDIM],
    numC, numP, sommaC, sommaP;

  cout << setw(53) << "acquisizione      rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
      break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM] ,
      numC,numP,sommaC,sommaP;

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
        break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM],
      num[2], somma[2];

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
        break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM],
      num[2], somma[2];
  const int MAXNOME = 10;
  const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
      break;
    if (zona == CENTRO || zona == PERIFERIA)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM],
      num[2], somma[2];
  const int MAXNOME = 10;
  const char nomeZone[2][MAXNOME] = {"centro", "periferia"};
  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  numC = 0; numP = 0; sommaC = 0; sommaP = 0;
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
        || (zona == PERIFERIA && numP >= MAXDIM) )
      break;
    if (zona == CENTRO)
    { cout << "prezzo: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
```



```
    num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

numC = 0; numP = 0; sommaC = 0; sommaP = 0;
while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
visualizzaDeviazione("periferia", sommaP, numP, prezziP);
{
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

numC = 0; numP = 0; sommaC = 0; sommaP = 0;
while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",     sommaC, numC, prezziC);
visualizzaDeviazione("periferia",  sommaP, numP, prezziP);
{
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
    { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
    { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

numC = 0; numP = 0; sommaC = 0; sommaP = 0;

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      1) sommaC = 0
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      2) sommaP = 0
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",     sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}
visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; C1 = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; I1 = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",     sommaC, numC, prezziC);
```



```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona == CENTRO && numC >= MAXDIM)
      || (zona == PERIFERIA && numP >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                     && num[zona] >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                     && num[zona] >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro", sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona == CENTRO || zona == PERIFERIA)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
      { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
    if (zona == PERIFERIA)
      { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
  }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",     sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",     sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { sommaC += prezzo; prezziC[numC] = prezzo; numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { somma[CENTRO] += prezzo;
          prezziC[numC] = prezzo;
          numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { somma[CENTRO] += prezzo;
          prezzi[CENTRO][numC] = prezzo;
          numC++; }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { somma[CENTRO] += prezzo;
          prezzi[CENTRO][num[CENTRO]] = prezzo;
          num[CENTRO]++;
        }
      if (zona == PERIFERIA)
        { sommaP += prezzo; prezziP[numP] = prezzo; numP++; }
    }
  else
    cout << "zona non corretta: inserire nuovo valore";
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { somma[CENTRO] += prezzo;
          prezzi[CENTRO][num[CENTRO]] = prezzo;
          num[CENTRO]++;
        }
      if (zona == PERIFERIA)
        { somma[PERIFERIA] += prezzo;
          prezzi[PERIFERIA][num[PERIFERIA]] = prezzo;
          num[PERIFERIA]++;
        }
    }
  }
else
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
    { somma[CENTRO] += prezzo;
      prezzi[CENTRO][num[CENTRO]] = prezzo;
      num[CENTRO]++;
    }
    if (zona == PERIFERIA)
    { somma[PERIFERIA] += prezzo;
      prezzi[PERIFERIA][num[PERIFERIA]] = prezzo;
      num[PERIFERIA]++;
    }
  }
  else
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    if (zona == CENTRO)
    { somma[CENTRO] += prezzo;
      prezzi[CENTRO][num[CENTRO]] = prezzo;
      num[CENTRO]++;
    }
    if (zona == PERIFERIA)
    { somma[PERIFERIA] += prezzo;
      prezzi[PERIFERIA][num[PERIFERIA]] = prezzo;
      num[PERIFERIA]++;
    }
  }
  else
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      if (zona == CENTRO)
        { somma[CENTRO] += prezzo;
          prezzi[CENTRO][num[CENTRO]] = prezzo;
          num[CENTRO]++;
        }
      if (zona == PERIFERIA)
        { somma[PERIFERIA] += prezzo;
          prezzi[PERIFERIA][num[PERIFERIA]] = prezzo;
          num[PERIFERIA]++;
        }
    }
  else
}
```

```
num[2], somma[2];
const int MAXNOME = 10;
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione    rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < 2)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    somma[zona] += prezzo;
    prezzi[zona][num[zona]] = prezzo;
    num[zona]++;
  }
  else
    cout << "zona non corretta: inserire un nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
visualizzaDeviazioni("periferia",   sommaP, numP, prezziP);
```

```
for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                     && num[zona] >= MAXDIM) )
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
      num[zona]++;
    }
  else
    cout << "zona non corretta: inserire un nuovo valore";
}

visualizzaDeviazioni("centro",      sommaC, numC, prezziC);
visualizzaDeviazioni("periferia",   sommaP, numP, prezziP);
}
```

```
for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                     && num[zona] >= MAXDIM) )
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
      num[zona]++;
    }
  else
    cout << "zona non corretta: inserire un nuovo valore";
}

for (zona = 0; zona < 2; zona++)
visualizzaDeviazioni
  (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);
}
```

```
for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                     && num[zona] >= MAXDIM) )
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
      num[zona]++;
    }
  else
    cout << "zona non corretta: inserire un nuovo valore";
}

for (zona = 0; zona < 2; zona++)
  visualizzaDeviazioni
  (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);
}
```

...

```
void visualizzaDeviazioni  
    (char nomeZona[], int somma, int num, int prezzi[])
```

...

```
    break;  
if (zona >= 0 && zona < 2)  
{ cout << "prezzo della nuova rilevazione: ";  
  cin >> prezzo;  
  somma[zona] += prezzo;  
  prezzi[zona][num[zona]] = prezzo;  
  num[zona]++;  
}  
else  
  cout << "zona non corretta: inserire un nuovo valore";  
}  
  
for (zona = 0; zona < 2; zona++)  
  visualizzaDeviazioni  
    (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);  
}
```

...

```
void visualizzaDeviazioni  
    (char nomeZona[], int somma, int num, int prezzi[])
```

...

```
    break;  
if (zona >= 0 && zona < 2)  
{ cout << "prezzo della nuova rilevazione: ";  
    cin >> prezzo;  
    somma[zona] += prezzo;  
    prezzi[zona][num[zona]] = prezzo;  
    num[zona]++;  
}  
else  
    cout << "zona non corretta: inserire un nuovo valore";  
}  
  
for (zona = 0; zona < 2; zona++)  
    visualizzaDeviazioni  
        (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);  
}
```

...

```
void visualizzaDeviazioni  
    (char nomeZona[], int somma, int num, int prezzi[])
```

...

```
    break;  
if (zona >= 0 && zona < 2)  
{ cout << "prezzo della nuova rilevazione: ";  
    cin >> prezzo;  
    somma[zona] += prezzo;  
    prezzi[zona][num[zona]] = prezzo;  
    num[zona]++;  
}  
else  
    cout << "zona non corretta: inserire un nuovo valore";  
}  
  
for (zona = 0; zona < 2; zona++)  
    visualizzaDeviazioni  
        (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);  
}
```

```
for(zona = 0; zona < 2; zona++)  
  
}  
  
while(1)  
{ cout << "zona della nuova rilevazione: "; cin >> zona;  
    if (cin.eof() || (zona >= 0 && zona < 2  
                      && num[zona] >= MAXDIM))  
        break;  
    if (zona >= 0 && zona < 2)  
    { cout << "prezzo della nuova rilevazione: ";  
        cin >> prezzo;  
        somma[zona] += prezzo;  
        prezzi[zona][num[zona]] = prezzo;  
        num[zona]++;  
    }  
    else  
        cout << "zona non corretta: inserire un nuovo valore";  
}  
  
for (zona = 0; zona < 2; zona++)  
    visualizzaDeviazioni  
        (nomeZone[zona], somma[zona], num[zona], prezzi[zona]);  
}
```

Più compatto

```
#include <iostream>
using namespace std;
#include <iomanip.h>

void visualizzaDeviazioni
    (char nomeZona[], int somma, int num, int prezzi[])
{ float media, deviazione;
int i;
if (num != 0)
{ cout << endl << setw(20)
    << "rilevazioni zona "
    << setw(10) << nomeZona << endl << endl
    << setw(20) << "prezzi rilevati"
    << setw(40) << "deviazione rispetto al prezzo medio"
    << endl;
media = (float)somma / num;
for (i = 0; i < num; i++)
{ deviazione = prezzi[i] - media;
cout << setw(20) << prezzi[i]
    << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione
    << endl;
}
}
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM],
      num[2], somma[2];
  const int MAXNOME = 100;
  const char nomeZone[2][MAXNOME] = {"centro", "periferia"};

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  for(zona = 0; zona < 2; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < 2
                      && num[zona] >= MAXDIM))
      break;
    if (zona >= 0 && zona < 2)
      { cout << "prezzo della nuova rilevazione: ";
        cin >> prezzo;
        somma[zona] += prezzo;
        prezzi[zona][num[zona]] = prezzo;
        num[zona]++;
      }
    else
      cout << "zona non corretta: inserire nuovo valore";
  }
}
```

```
for (zona = 0; zona < 2; zona++)
    visualizzaDeviazioni(nomeZone[zona], somma[zona],
                          num[zona], prezzi[zona]);
}
```

Ampliamo il programma

```
FERIA = 1;  
const int MAXDIM = 100;  
MAXNOME = 10;  
int zona;  
int prezzo,  
    prezzi[2][MAXDIM],  
    num[2], somma[2];  
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};  
  
cout << setw(53) << "acquisizione    rilevazioni" << endl;  
  
for(zona = 0; zona < 2; zona++)  
{ num[zona] = 0; somma[zona] = 0;  
}  
  
while(1)  
{ cout << "zona della nuova rilevazione: "; cin >> zona;  
if (cin.eof() || (zona >= 0 && zona < 2  
                  && num[zona] >= MAXDIM))  
    break;  
if (zona >= 0 && zona < 2)  
{ cout << "prezzo della nuova rilevazione: ";  
    cin >> prezzo;  
    somma[zona] += prezzo;  
    prezzi[zona][num[zona]] = prezzo;  
    num[zona]++;  
}
```

Ampliamo il dominio dei dati

Aumentiamo il
numero delle zone



```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  int zona;
  int prezzo,
    prezzi[2][MAXDIM],
    num[2], somma[2];
  const char nomeZone[2][MAXNOME] = { "centro", "periferia" };

  cout << setw(53) << "acquisizioni rilevazioni" << endl;

  for(zona = 0; zona < 2; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 2 & zona < 2
                      && num[zona] >= MAXDIM) )
      break;
    if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
      num[zona]++;
    }
  }
}
```

**Numero massimo
di zone**

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;
  int zona;
  int prezzo,
      prezzi[2][MAXDIM],
      num[2], somma[2];
  const char nomeZone[2][MAXNOME] = {"centro", "periferia"};
  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  for(zona = 0; zona < 2; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < 2
                      && num[zona] >= MAXDIM) )
      break;
    if (zona >= 0 && zona < 2)
      { cout << "prezzo della nuova rilevazione: ";
        cin >> prezzo;
        somma[zona] += prezzo;
        prezzi[zona][num[zona]] = prezzo;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
int zona;
int prezzo,
  prezzi[2][MAXDIM],
  num[2], somma[2];
const char nomeZone[2][MAXNOME] = {"centro", "periferia"};
cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < 2; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < 2
                    && num[zona] >= MAXDIM) )
    break;
  if (zona >= 0 && zona < 2)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
int zona;
int prezzo,
  prezzi[MAXZONE][MAXDIM],
  num[MAXZONE], somma[MAXZONE];
const char nomeZone[MAXZONE][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < MAXZONE; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < MAXZONE
                    && num[zona] >= MAXDIM) )
    break;
  if (zona >= 0 && zona < MAXZONE)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;

  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << setiosflags(ios::fixed) << "accu. rilevazioni" << endl;

  for(zona = 0; zona < MAXZONE; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < MAXZONE
                      && num[zona] >= MAXDIM) )
      break;
    if (zona >= 0 && zona < MAXZONE)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
    }
  }
}
```

Effettivo numero di zone



```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;
int numZone;
int zona;
int prezzo,
    prezzi[MAXZONE][MAXDIM],
    num[MAXZONE], somma[MAXZONE];
const char nomeZone[MAXZONE][MAXNOME] = {"centro", "periferia"};

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < MAXZONE; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
  if (cin.eof() || (zona >= 0 && zona < MAXZONE
                    && num[zona] >= MAXDIM))
    break;
  if (zona >= 0 && zona < MAXZONE)
  { cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    somma[zona] += prezzo;
    prezzi[zona][num[zona]] = prezzo;
}
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  const char* zone[MAXZONE][MAXNOME] = { "centro", "periferia" };
  cout << setw(53) << "acquisizione" << rilevazioni" << endl;
  for(zona = 0; zona < MAXZONE; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }
  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < MAXZONE
                      && num[zona] >= MAXDIM))
      break;
    if (zona >= 0 && zona < MAXZONE)
      { cout << "prezzo della nuova rilevazione: ";
        cin >> prezzo;
        somma[zona] += prezzo;
        prezzi[zona][num[zona]] = prezzo;
      }
  }
```

Dare un nome
alle zone



```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  const char nomeZone[MAXZONE][MAXNOME] = {"centro", "periferia"};
cout << setw(53) << "acquisizione rilevazioni" << endl;
  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < MAXZONE; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < MAXZONE
                      && num[zona] >= MAXDIM))
      break;
    if (zona >= 0 && zona < MAXZONE)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  MAXNOME = 10;
  MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << setw(53) << "acquisizione    rilevazioni" << endl;

  for(zona = 0; zona < MAXZONE; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < MAXZONE
                      && num[zona] >= MAXDIM))
      break;
    if (zona >= 0 && zona < MAXZONE)
    { cout << "prezzo della nuova rilevazione: ";
      cin >> prezzo;
      somma[zona] += prezzo;
      prezzi[zona][num[zona]] = prezzo;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << ") : ";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 20;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 99;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];
  {
    cout << "numero di zone presenti (massimo " << MAXZONE << "):";
    cin >> numZone;
    for (zona = 0; zona < numZone; zona++)
    { cout << "nome della zona denotata nel seguito"
        << " con il numero progressivo " << zona << ": ";
      cin >> nomeZone[zona];
    }
    cout << setw(53) << "acquisizione rilevazioni" << endl;
    for(zona = 0; zona < numZone; zona++)
    { num[zona] = 0; somma[zona] = 0;
    }
    while(1)
    { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << ") : ";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
  const int MAXNOME = 10;
  const int MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << ") : ";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):"; cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << "):";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

Limite di leggibilità

```
PERIFERIA = 1;  
const int MAXDIM = 100;  
MAXNOME = 10;  
MAXZONE = 15;  
  
int numZone;  
int zona;  
int prezzo,  
    prezzi[MAXZONE][MAXDIM];  
int num[MAXZONE], somma[MAXZONE];  
char nomeZone[MAXZONE][MAXNOME];  
  
cout << "numero di zone presenti (massimo " << MAXZONE << ") :";  
cin >> numZone;  
for (zona = 0; zona < numZone; zona++)  
{ cout << "nome della zona denotata nel seguito"  
     << endl;  
  cin >> nomeZone[zona];  
}  
  
cout << setw(53) << "acquisizione rilevazioni" << endl;  
  
for(zona = 0; zona < numZone; zona++)  
{ num[zona] = 0; somma[zona] = 0;  
}  
  
while(1)  
{ cout << "zona della nuova rilevazione: "; cin >> zona;
```



```
int main()
{ const int CENTRO = 0, PERIFERIA = 1;
  const int MAXDIM = 100;
    MAXNOME = 10;
    MAXZONE = 15;
  int numZone;
  int zona;
  int prezzo,
      prezzi[MAXZONE][MAXDIM],
      num[MAXZONE], somma[MAXZONE];
  char nomeZone[MAXZONE][MAXNOME];

  cout << "numero di zone presenti (massimo " << MAXZONE << ") : ";
  cin >> numZone;
  for (zona = 0; zona < numZone; zona++)
  { cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
  }

  cout << setw(53) << "acquisizione rilevazioni" << endl;

  for(zona = 0; zona < numZone; zona++)
  { num[zona] = 0; somma[zona] = 0;
  }

  while(1)
  { cout << "zona della nuova rilevazione: "; cin >> zona;
```

```
for (zona = 0; zona < numZone; zona++)
{ cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
}

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < numZone; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
if (cin.eof() || (zona >= 0 && zona < numZone
                    && num[zona] >= MAXDIM))
    break;
if (zona >= 0 && zona < numZone)
{ cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    somma[zona] += prezzo;
    prezzi[zona][num[zona]] = prezzo;
    num[zona]++;
}
else
    cout << "zona non corretta: inserire nuovo valore";
}
```

```
for (zona = 0; zona < numZone; zona++)
{ cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
}

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < numZone; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
    if (cin.eof() || (zona >= 0 && zona < numZone
                      && num[zona] >= MAXDIM))
        break;
    if (zona >= 0 && zona < numZone)
        { cout << "prezzo della nuova rilevazione: ";
          cin >> prezzo;
          somma[zona] += prezzo;
          prezzi[zona][num[zona]] = prezzo;
          num[zona]++;
        }
    else
        cout << "zona non corretta: inserire nuovo valore";
}
```

```
for (zona = 0; zona < numZone; zona++)
{ cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
}

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < numZone; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
if (cin.eof())
    break;
if (zona >= 0 && zona < numZone)
{ if (num[zona] >= MAXDIM)
    break;
    cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    somma[zona] += prezzo;
    prezzi[zona][num[zona]] = prezzo;
    num[zona]++;
}
else
    cout << "zona non corretta: inserire nuovo valore";
}
```

```
for (zona = 0; zona < numZone; zona++)
{ cout << "nome della zona denotata nel seguito"
    << " con il numero progressivo " << zona << ": ";
    cin >> nomeZone[zona];
}

cout << setw(53) << "acquisizione rilevazioni" << endl;

for(zona = 0; zona < numZone; zona++)
{ num[zona] = 0; somma[zona] = 0;
}

while(1)
{ cout << "zona della nuova rilevazione: "; cin >> zona;
if (cin.eof())
    break;
if (zona >= 0 && zona < numZone)
{ if (num[zona] >= MAXDIM)
    break;
    cout << "prezzo della nuova rilevazione: ";
    cin >> prezzo;
    somma[zona] += prezzo;
    prezzi[zona][num[zona]] = prezzo;
    num[zona]++;
}
else
    cout << "zona non corretta: inserire nuovo valore";
}
```

Programma

- Rileva il costo di un prodotto

- in una zona geografica che l'utente suddividere in diverse zone

- di cui l'utente può scegliere numero e nome

Programma

- Rileva il costo di un prodotto
 - in una zona geografica che l'utente può suddividere in diverse zone
 - di cui l'utente può scegliere numero e nome