



POLITECNICO
DI MILANO

INFORMATICA

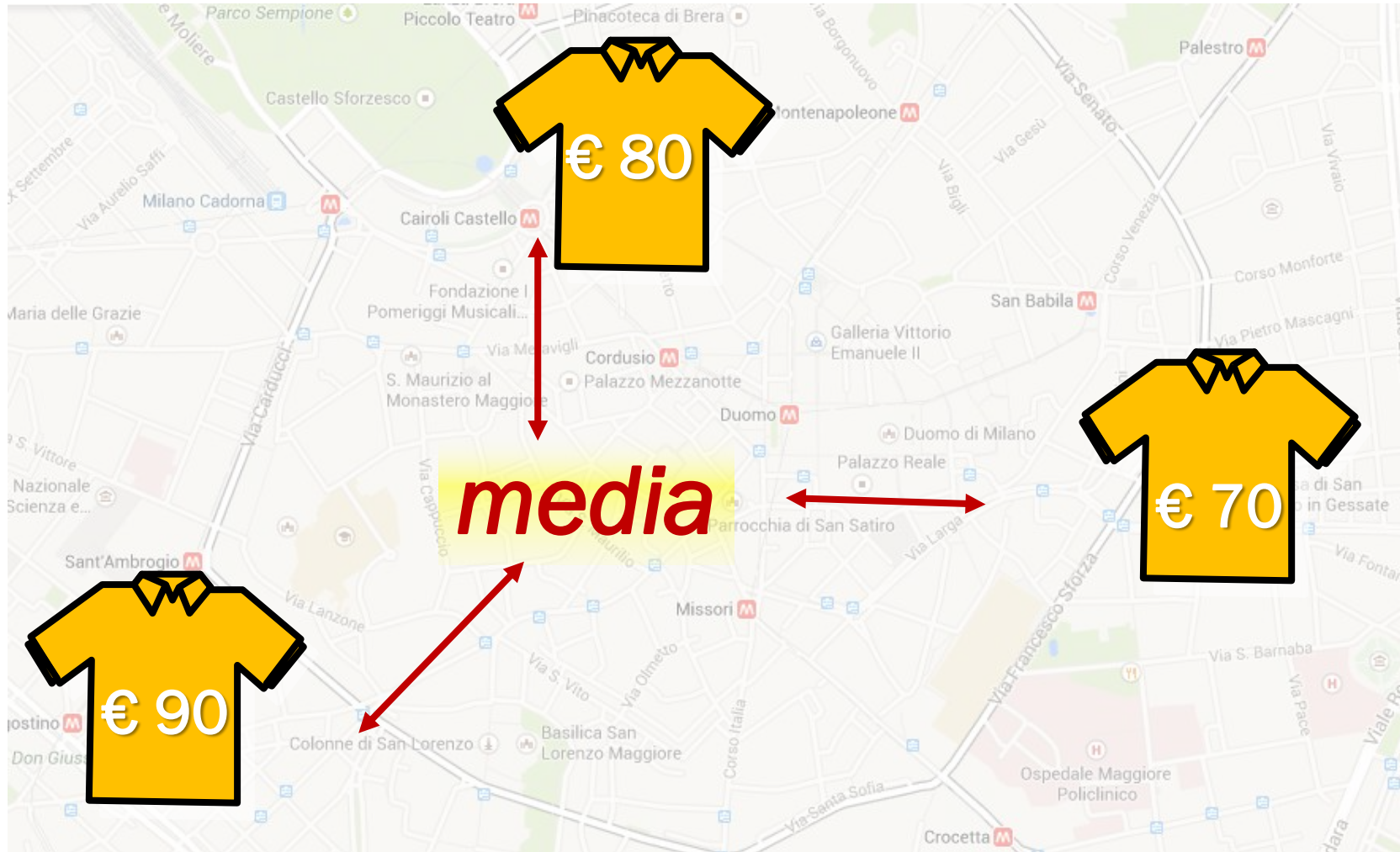
Analisi del prezzo di
un prodotto in
un'area geografica

```
void funzione(int Array[])  
{...}
```

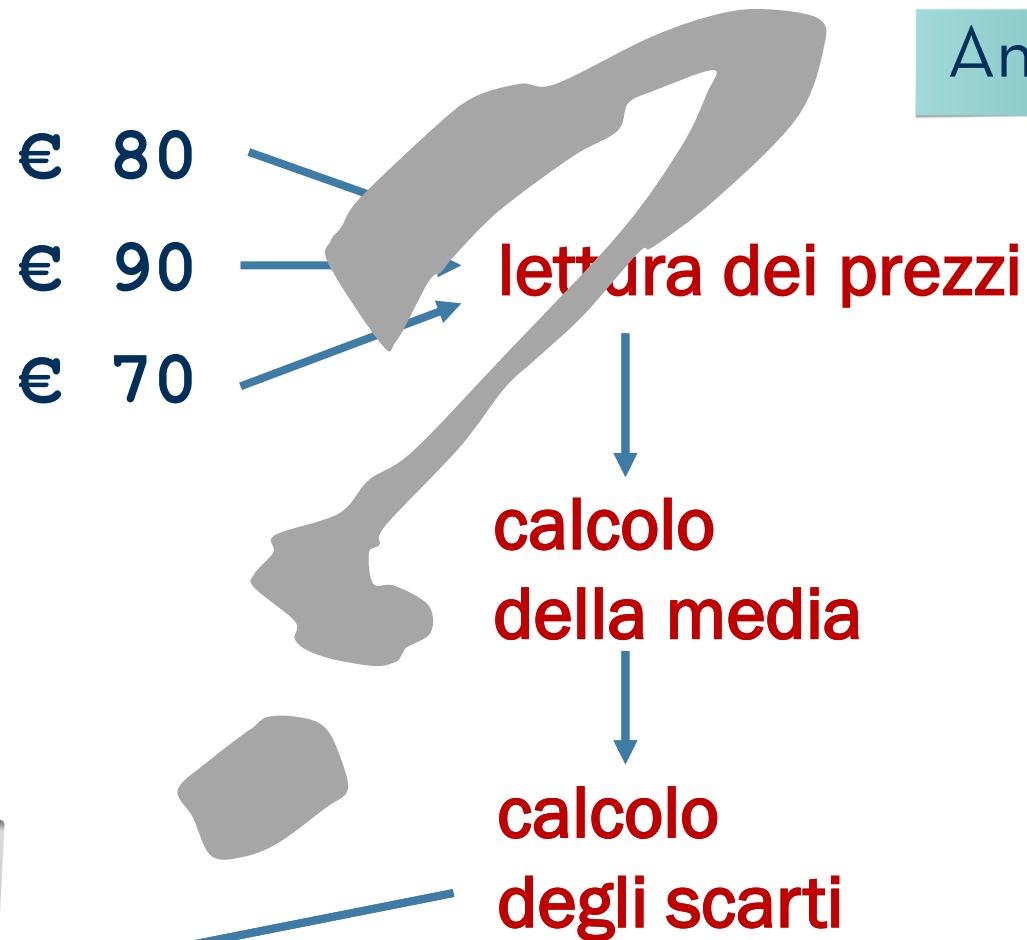
```
funzione( parametro );
```



Scarto



Analisi e specifica



Analisi e specifica



acquisizione rilevazioni

prezzo della nuova rilevazione: 80
prezzo della nuova rilevazione: 90
prezzo della nuova rilevazione: 70

prezzo rilevati	deviazione rispetto al prezzo medio
80	0.00
90	10.00
70	-10.00

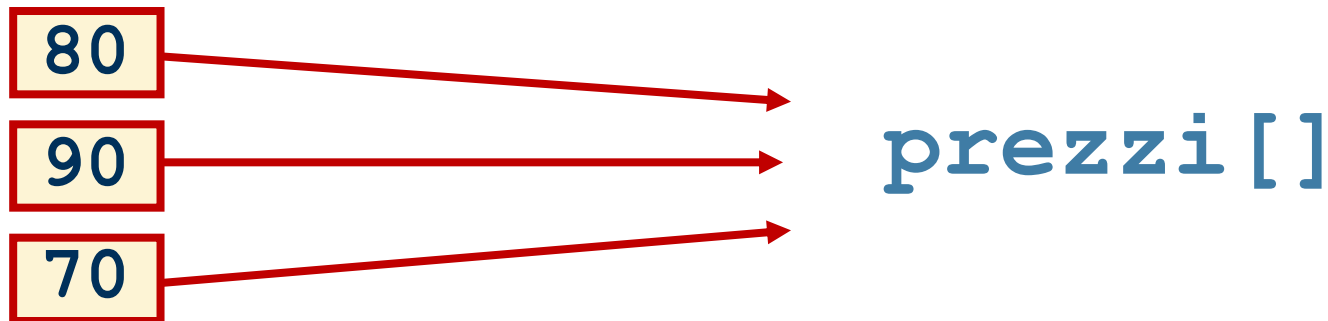
Soluzione

Obiettivo

**Calcolare lo scarto
di ciascuna rilevazione
rispetto alla media**

Soluzione

Calcolare lo scarto
di ciascuna rilevazione
rispetto alla media




```
const int MAXDIM = 100;  
int prezzi[ MAXDIM ];
```

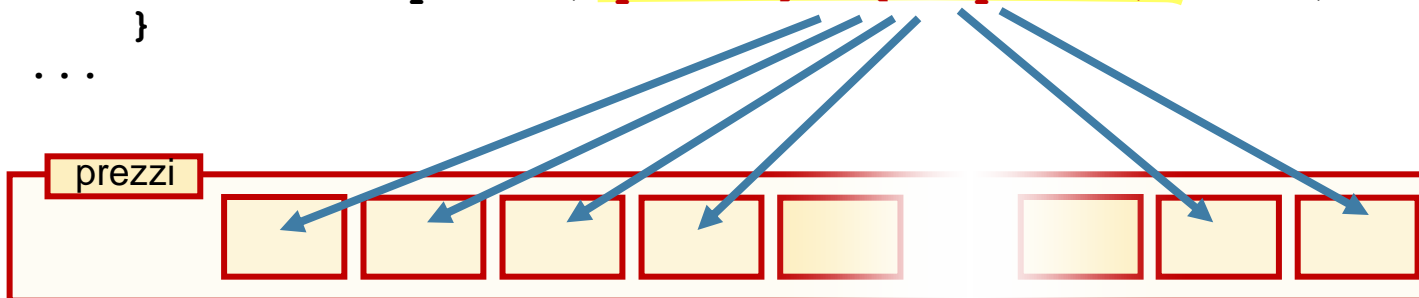


```
// Analisi del prezzo di un prodotto in una città
```

```
...  
int main ()  
{ const int MAXDIM = 100;  
  int prezzo,           // prezzo della rilevazione corrente  
    prezzi[MAXDIM],     // prezzi rilevati  
    num, somma,         // numero e somma dei prezzi  
    i;  
  float media, deviazione // media dei prezzi e scarto  
  cout << setw(53) << "acquisizione rilevazioni" << endl;  
  num = 0; somma = 0;  
  while(1)  
  { cout << "prezzo della nuova rilevazione: "; cin >> prezzo;  
    if (cin.eof() || num >= MAXDIM)  
      break;  
    somma += prezzo; prezzi[num] = prezzo; num++;  
  }  
  ...  
}
```

```
// Analisi del prezzo di un prodotto in una città
```

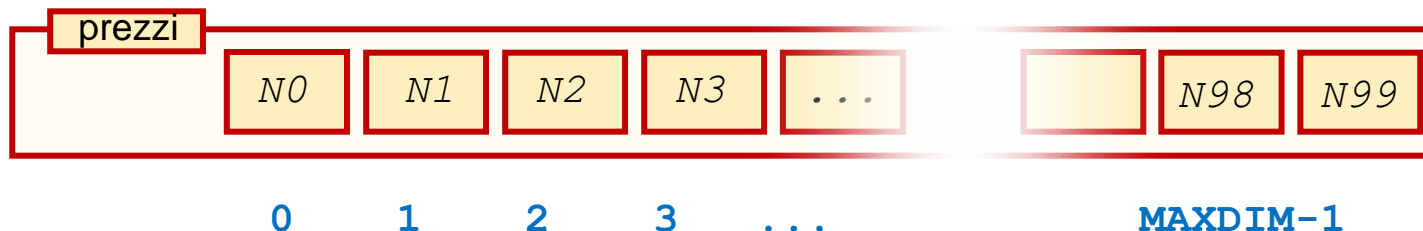
```
...  
int main ()  
{ const int MAXDIM = 100;  
  int prezzo,           // prezzo della rilevazione corrente  
    prezzi[MAXDIM],     // prezzi rilevati  
    num, somma,         // numero e somma dei prezzi  
    i;  
  float media, deviazione // media dei prezzi e scarto  
  cout << setw(53) << "acquisizione rilevazioni" << endl;  
  num = 0; somma = 0;  
  while(1)  
  { cout << "prezzo della nuova rilevazione: "; cin >> prezzo;  
    if (cin.eof() || num >= MAXDIM)  
      break;  
    somma += prezzo; prezzi[num] = prezzo; num++;  
  }  
  ...
```



Specifica

Prezzo di un prodotto in una città

```
...  
int main ()  
{ const int MAXDIM = 100;  
  int prezzo,           // prezzo della rilevazione corrente  
    prezzi[MAXDIM],     // prezzi rilevati  
    num, somma,         // numero e somma dei prezzi  
    i;  
  float media, deviazione // media dei prezzi e scarto  
  cout << setw(53) << "acquisizione rilevazioni" << endl;  
  num = 0; somma = 0;  
  while(1)  
  { cout << "prezzo della nuova rilevazione: "; cin >> prezzo;  
    if (cin.eof() || num >= MAXDIM)  
      break;  
    somma += prezzo; prezzi[num] = prezzo; num++;  
  }  
  ...
```



```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
  }
}
```

prezzi rilevati	deviazione rispetto al prezzo medio
80	0.00
90	10.00
70	-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
  }
}
```

prezzi rilevati

deviazione rispetto al prezzo medio

80

0.00

90

10.00

70

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
}
```

prezzi rilevati

80

90

70

deviazione rispetto al prezzo medio

0.00

10.00

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
  }
}
```

setw(n)
setw

prezzi rilevati

80

deviazione rispetto al prezzo medio

0.00

Giustificata a destra

zione zona e prezzo per ogni rilevazione
nuova rilevazione: "; cin >> prezzo;

```
if (cin.eof() || num >= MAXDIM)
    break;
somma += prezzo; prezzi[num] = prezzo; num++;
}
```

```
if (num != 0)
{
    cout << 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;
    }
}
```

prezzi rilevati

deviazione rispetto al prezzo medio

80

20

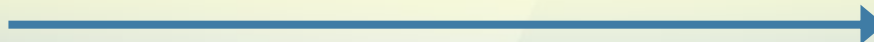
```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
}
```

prezzi rilevati



deviazione rispetto al prezzo medio



```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
  }
}
```

prezzi rilevati	deviazione rispetto al prezzo medio
80	0.00
90	10.00
70	-10.00

```
// Analisi del prezzo di un prodotto in una città
```

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

```
#include <iomanip.h>
```

← **setw(n)**

```
int main ()
{ const int MAXDIM = 100;
  int prezzo,                // prezzo della rilevazione corrente
    prezzi[MAXDIM],          // prezzi rilevati
    num, somma,              // numero e somma dei prezzi
    i;
  float media, deviazione    // media dei prezzi e scarto
  cout << setw(53) << "acquisizione rilevazioni" << endl;
  num = 0; somma = 0;
  while(1)
  { cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
    if (cin.eof() || num >= MAXDIM)
      break;
    somma += prezzo; prezzi[num] = prezzo; num++;
  }
}
```

prezzi rilevati

deviazione rispetto al prezzo medio

80

0.00

90

10.00

70

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
}
```

prezzi rilevati

80

90

70

deviazione rispetto al prezzo medio

0.00

10.00

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}
```

+0.1+10⁺²

```
if (num != 0)
{
  cout << 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;
  }
}
```

10.00

fixed point

prezzi rilevati

deviazione rispetto al prezzo medio

80

0.00

90

10.00

70

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
}
```

setiosflags

prezzi rilevati

deviazione rispetto al prezzo medio

80

0.00

90

10.00

70

-10.00


```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << 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;
    }
}
```

setiosflags

prezzi rilevati

deviazione rispetto al prezzo medio

80

0.00

90

10.00

70

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num == MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}
```

```
if (num != 0)
{
  cout << endl << setw(20) << "prezzi rilevati" << endl;
  << 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;
  }
}
```

+1.00*10⁺¹

10.00

setiosflags

prezzi rilevati

80

90

70

deviazione rispetto al prezzo medio

0.00

10.00

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num == MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}
```

```
if (num != 0)
{
  cout << endl << setw(20) << "prezzi rilevati" << endl;
  cout << 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] << endl;
    cout << setw(40) << setprecision(2)
    << setiosflags (ios::fixed | ios::showpoint)
    << deviazione << endl;
  }
}
```

+1.00*10⁺¹

10.00

prezzi rilevati

80

90

70

deviazione rispetto al prezzo medio

0.00

10.00

-10.00

```
while(1) // ciclo di acquisizione zona e prezzo per ogni rilevazione
{ cout << "prezzo della nuova rilevazione: "; cin >> prezzo;
  if (cin.eof() || num >= MAXDIM)
    break;
  somma += prezzo; prezzi[num] = prezzo; num++;
}

if (num != 0)
{
  cout << endl << setw(20) << "prezzi rilevati" << endl;
  cout << 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] << endl;
    cout << setw(40) << setprecision(2) << deviazione << endl;
    cout << setw(40) << setiosflags (ios::fixed | ios::showpoint) << deviazione << endl;
  }
}
```

10.00



prezzi rilevati

80

90

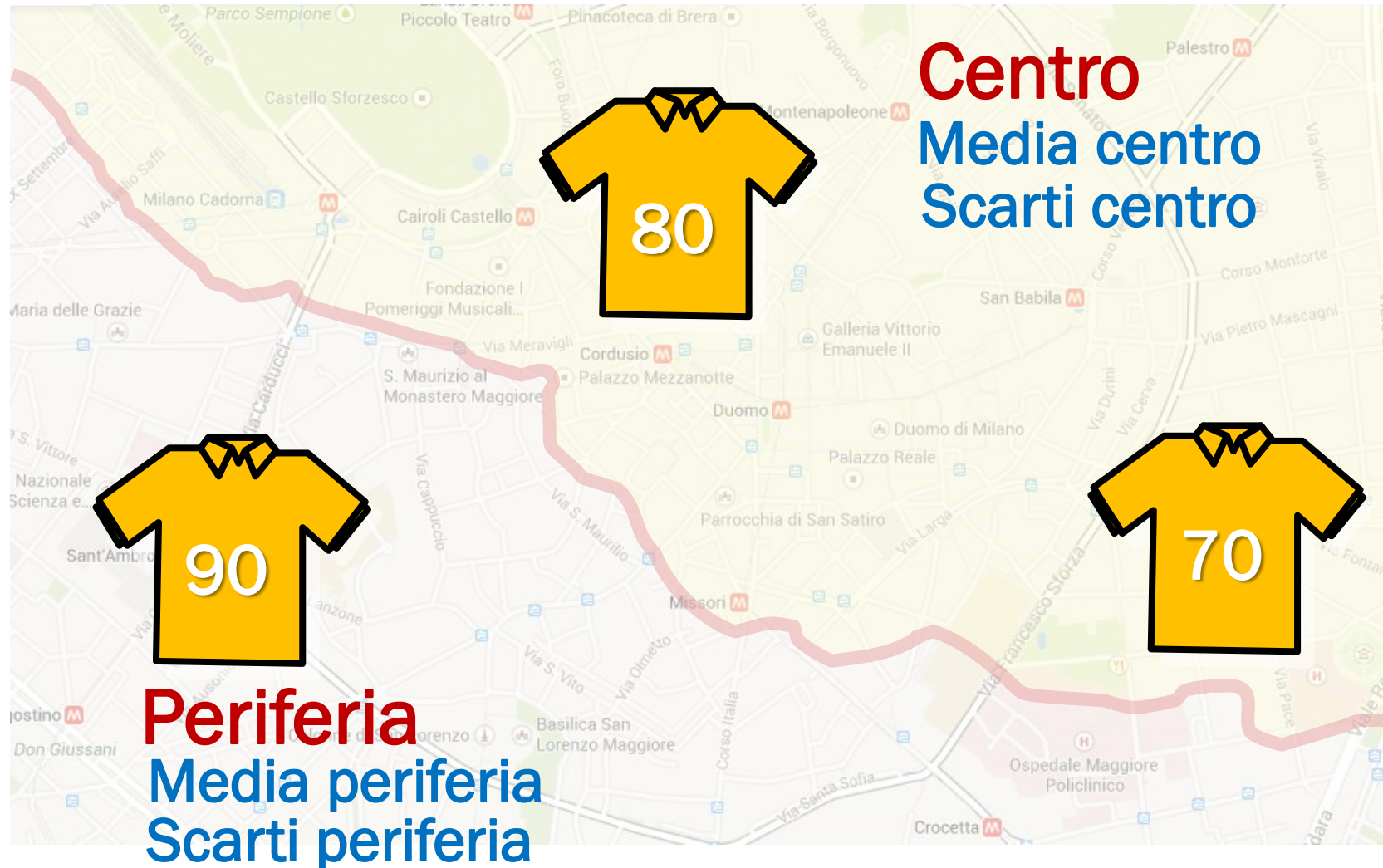
70

deviazione rispetto al prezzo medio

0

10

-10





acquisizione rilevazioni

zona della nuova rilevazione: c

prezzo della nuova rilevazione: 80

Presentazione dei risultati

rilevazioni zona centro

prezzi rilevati	deviazione rispetto al prezzo medio
80	5.00
70	-5.00

rilevazioni zona periferia

prezzi rilevati	deviazione rispetto al prezzo medio
90	0.00

1

prezzi rilevati	deviazione rispetto al prezzo medio	zona
80	0.00	c
90	10.00	p
70	-10.00	c



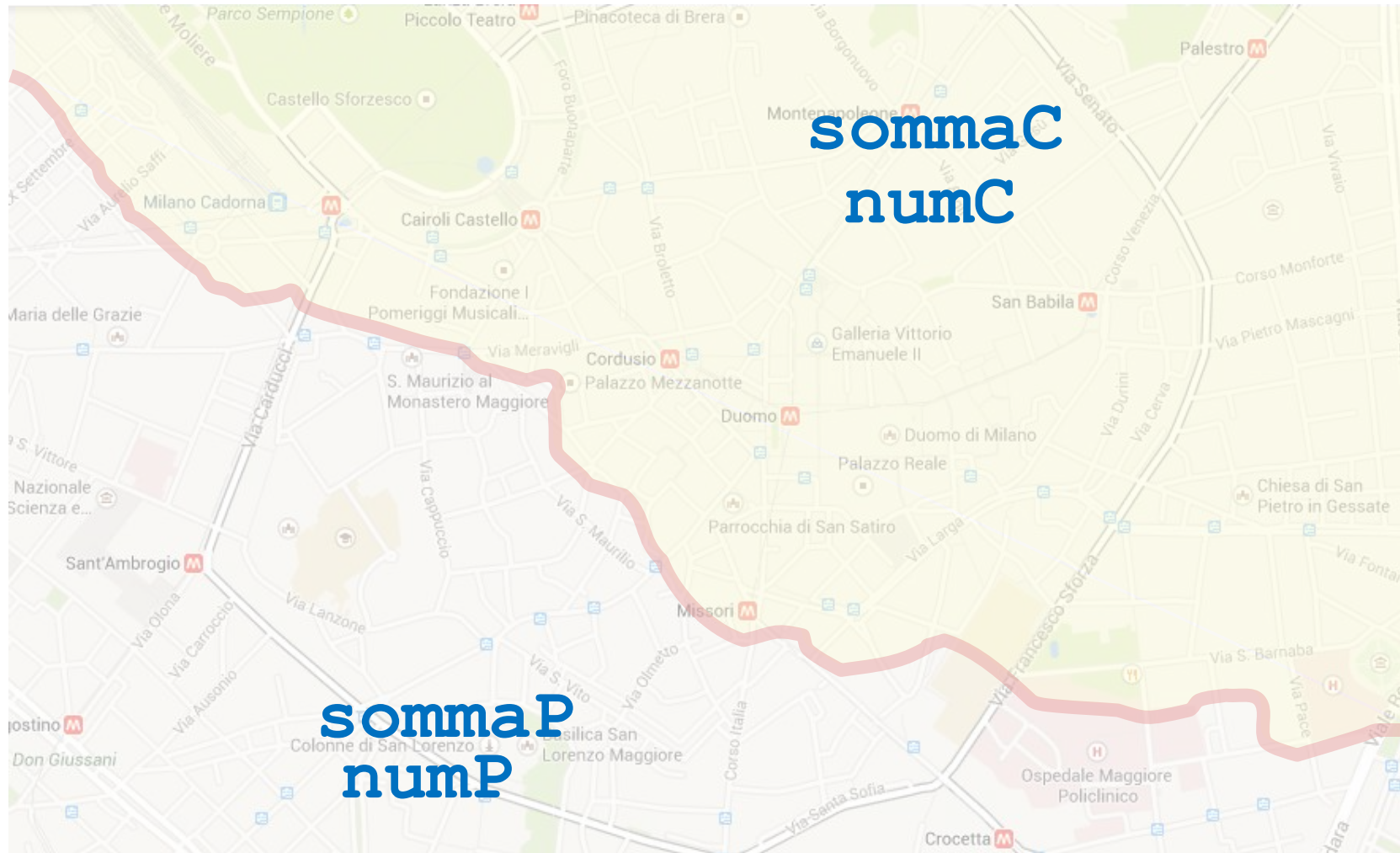
2

Nessuna ipotesi
sull'ordine dei
dati in ingresso

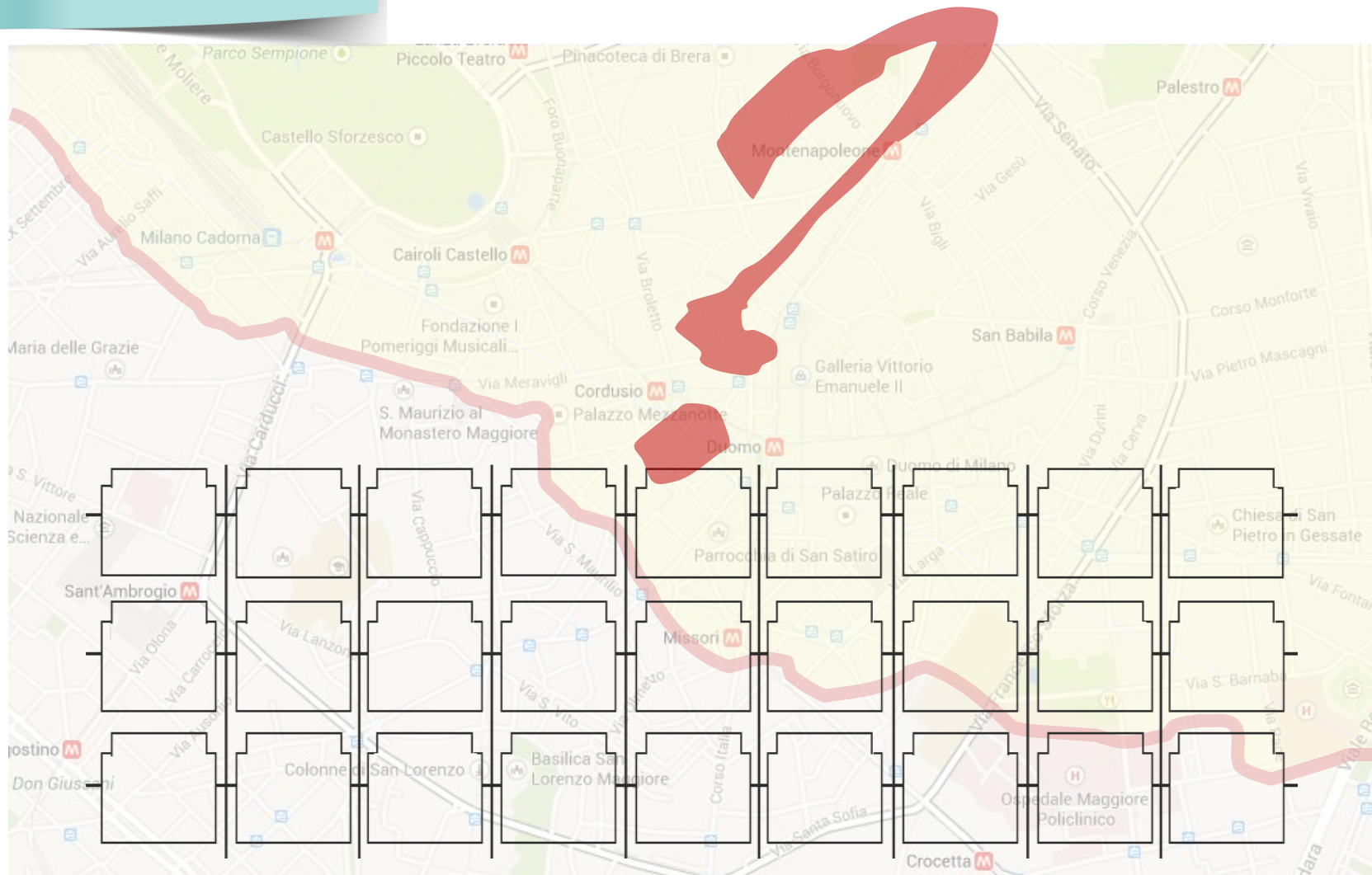
prezzi rilevati	deviazione rispetto al prezzo medio	zona
80	0.00	c
90	10.00	p
70	-10.00	c



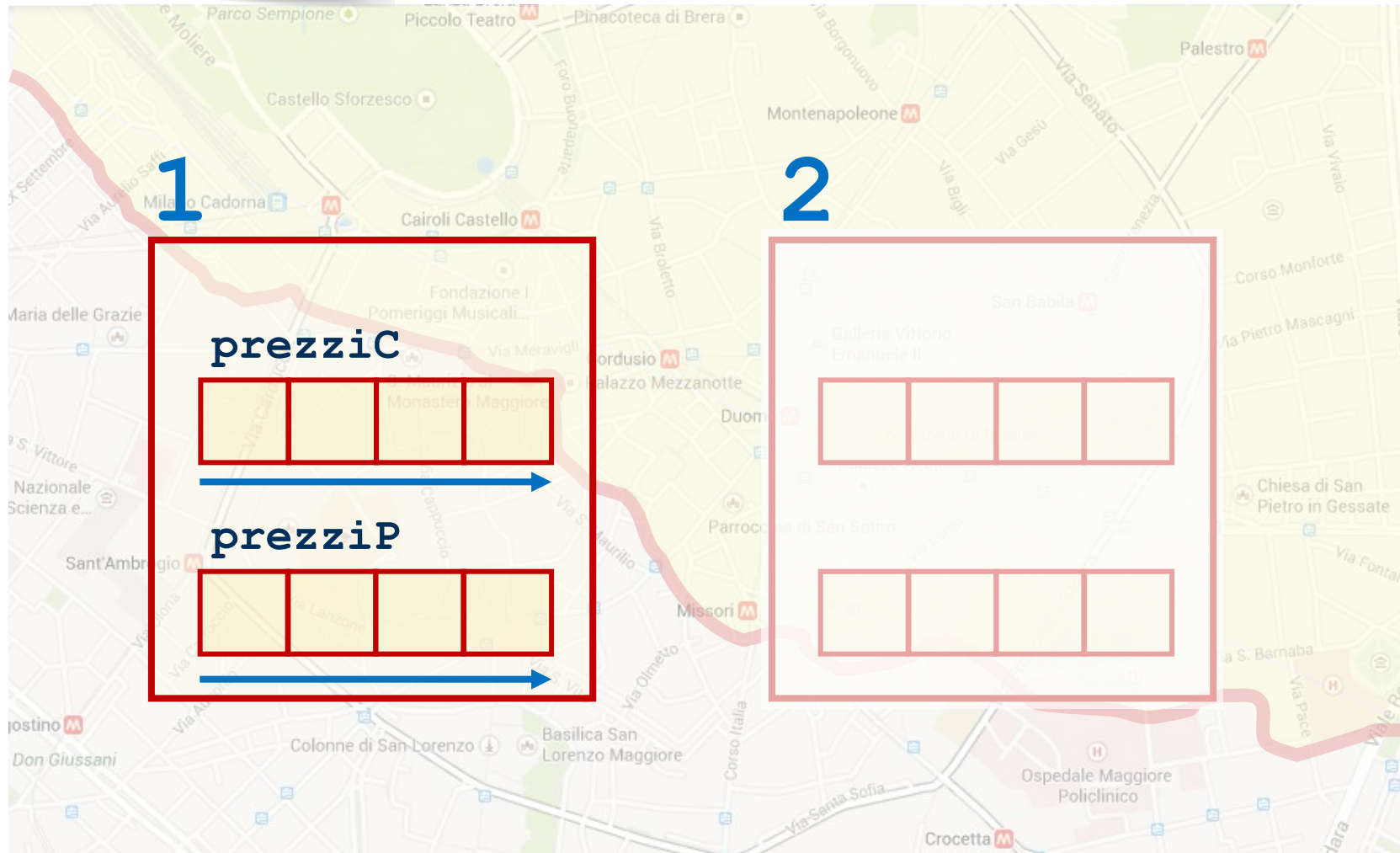




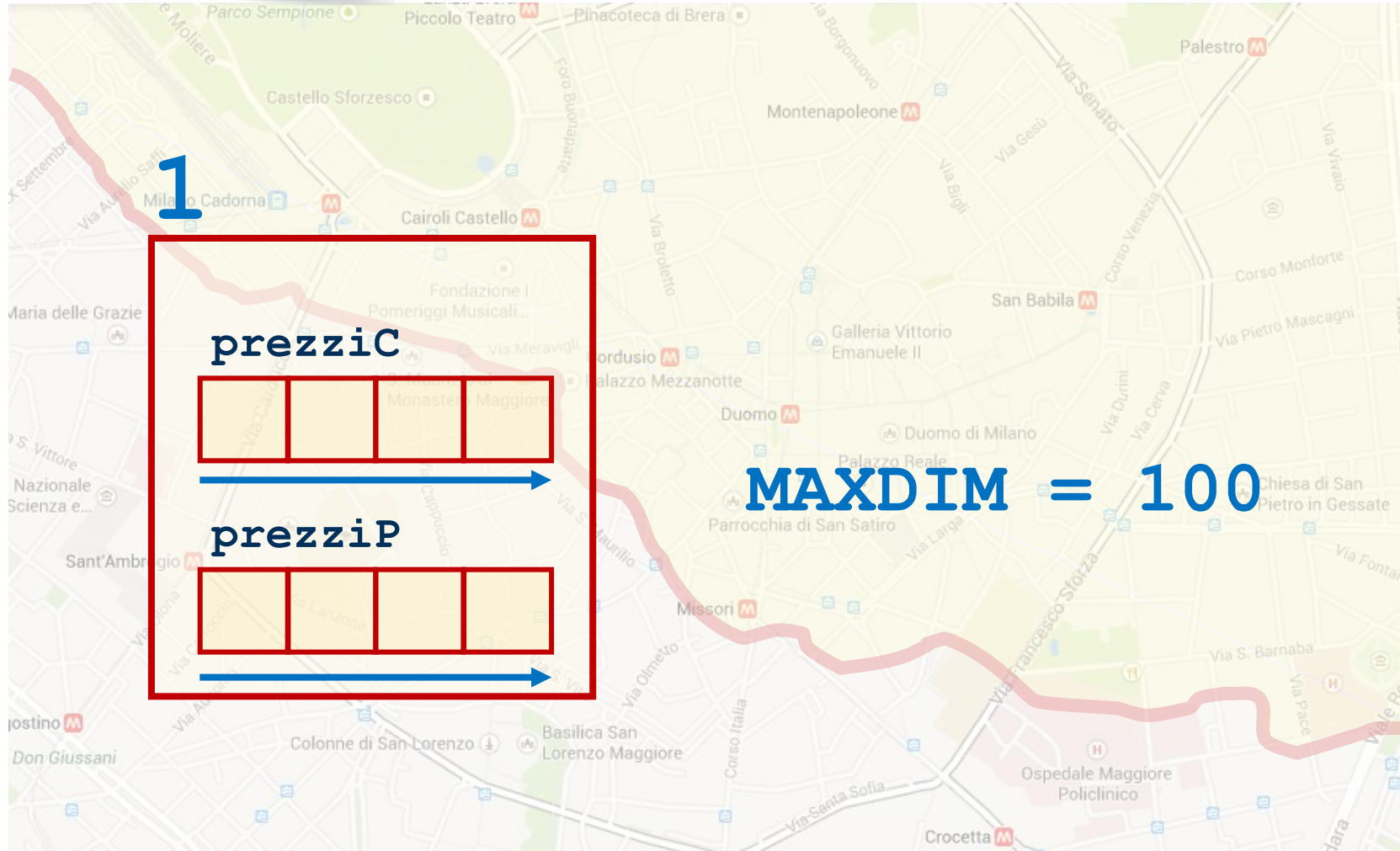
Memorizzazione



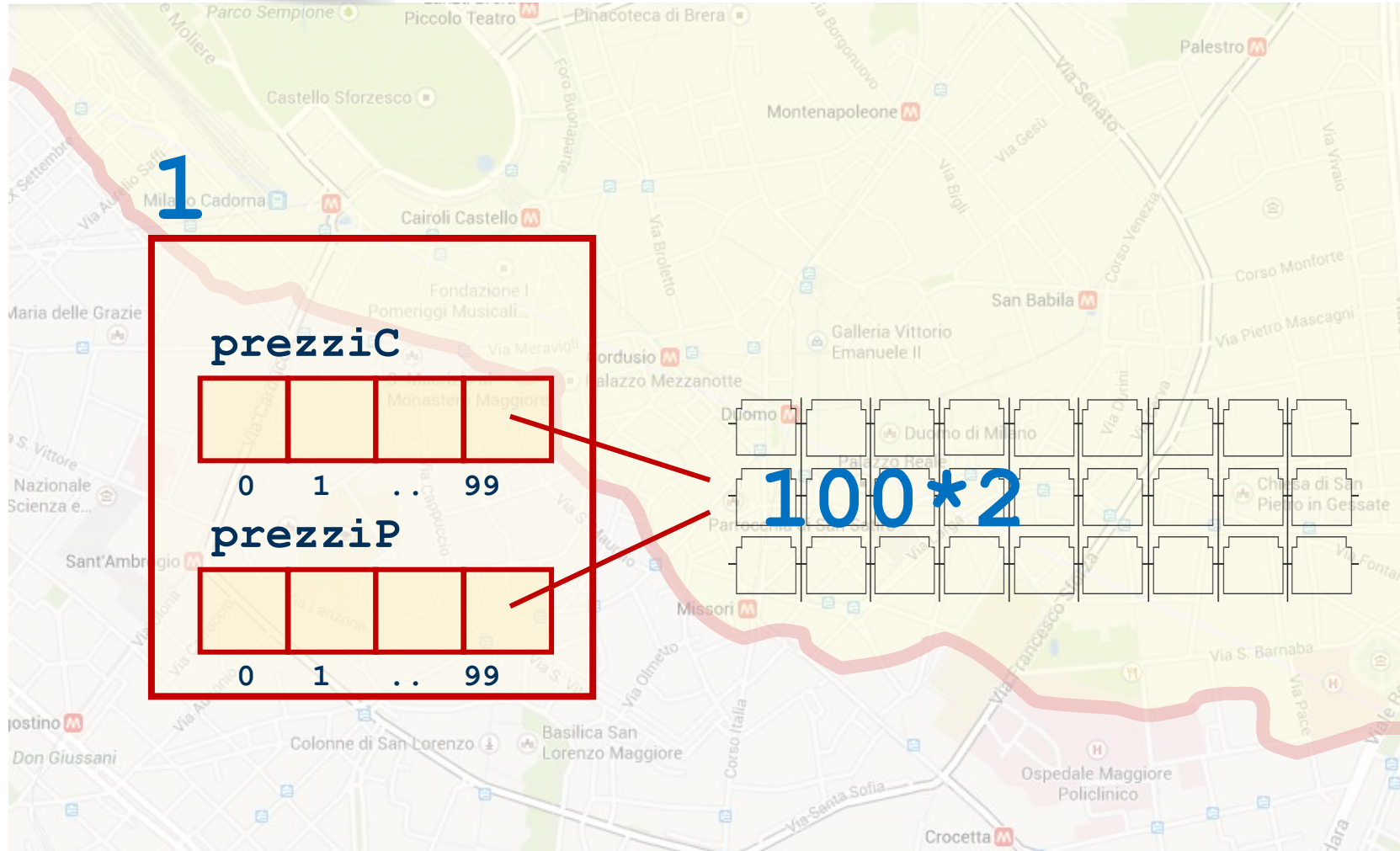
Memorizzazione



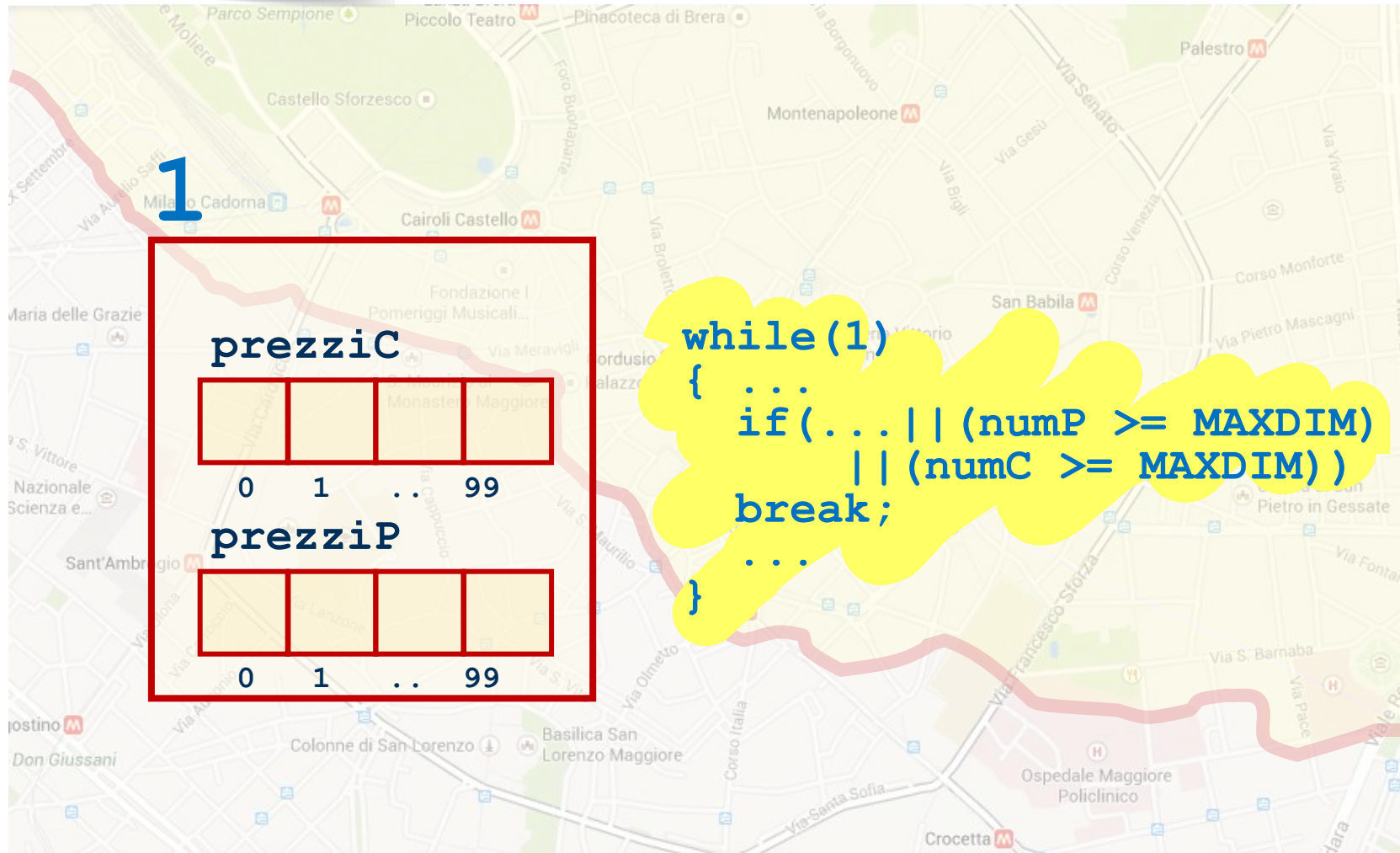
Memorizzazione



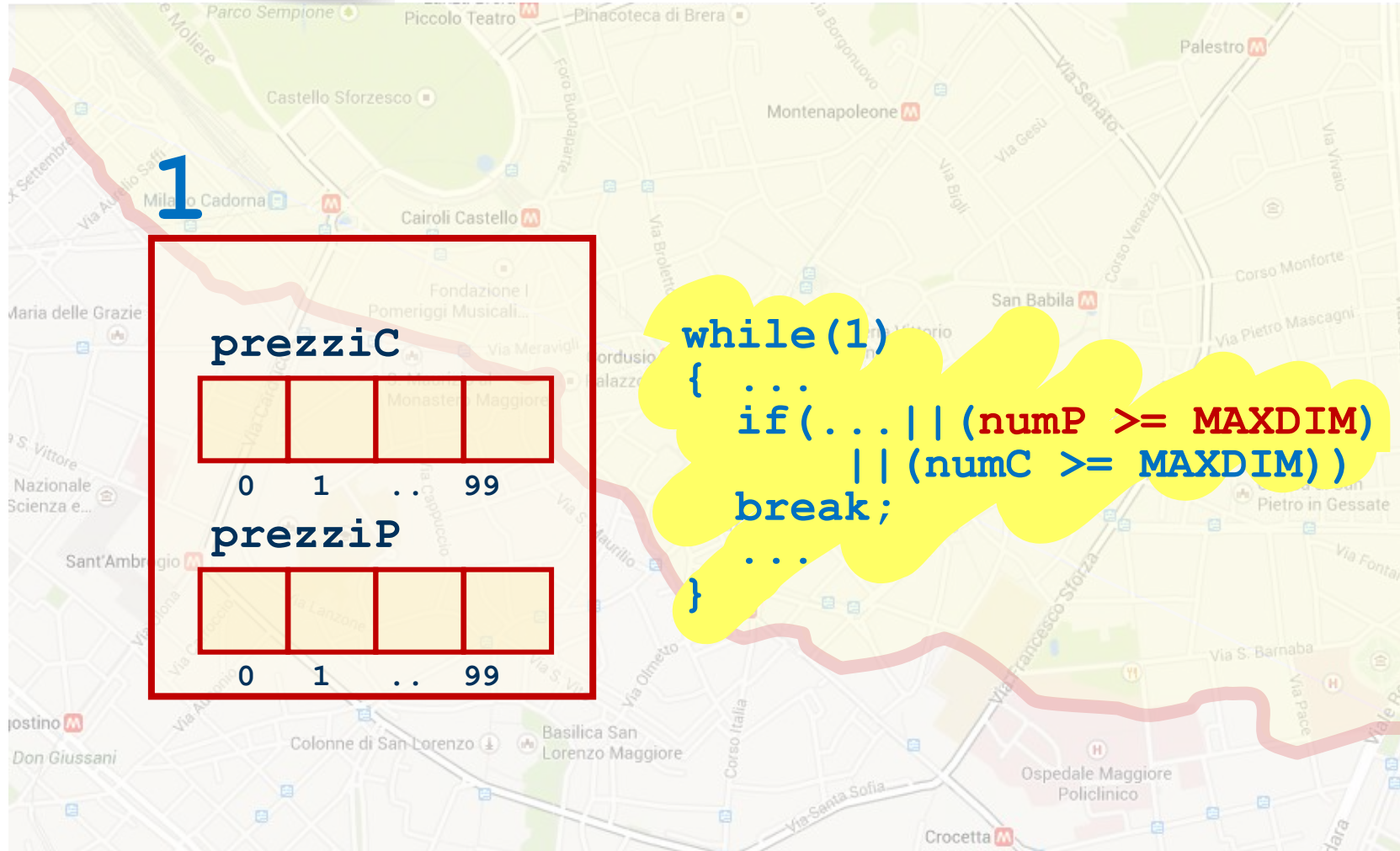
Memorizzazione



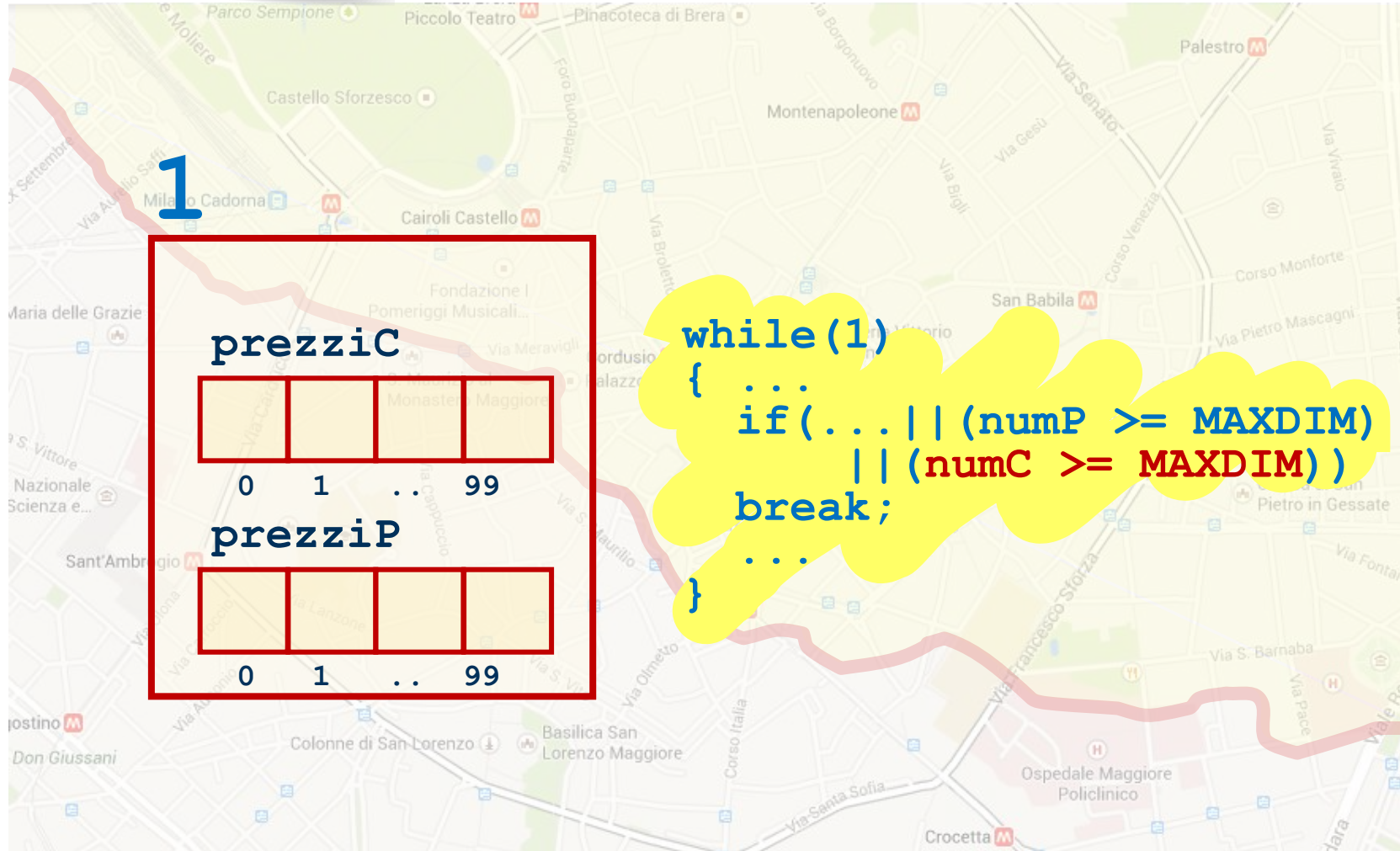
Memorizzazione



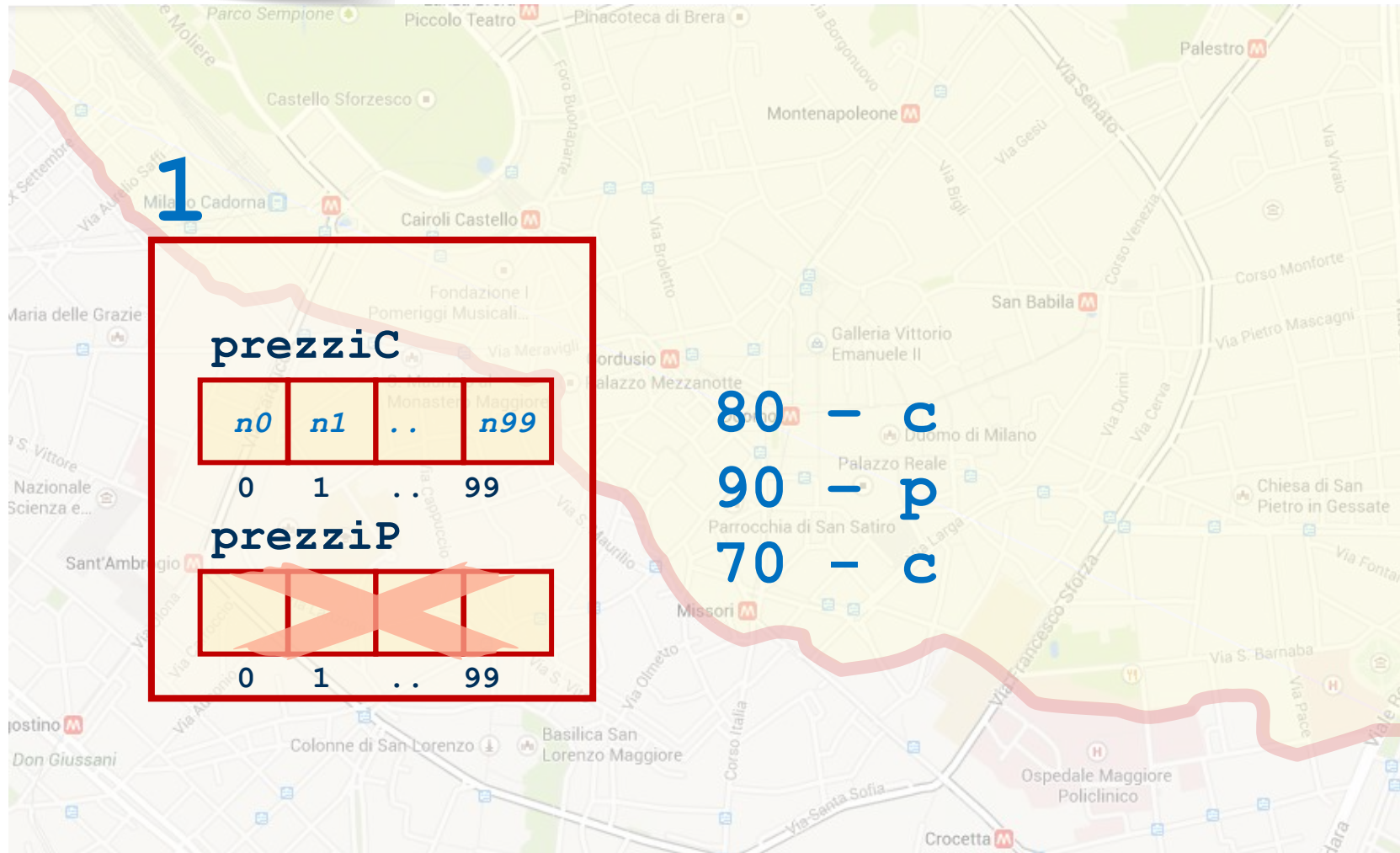
Memorizzazione



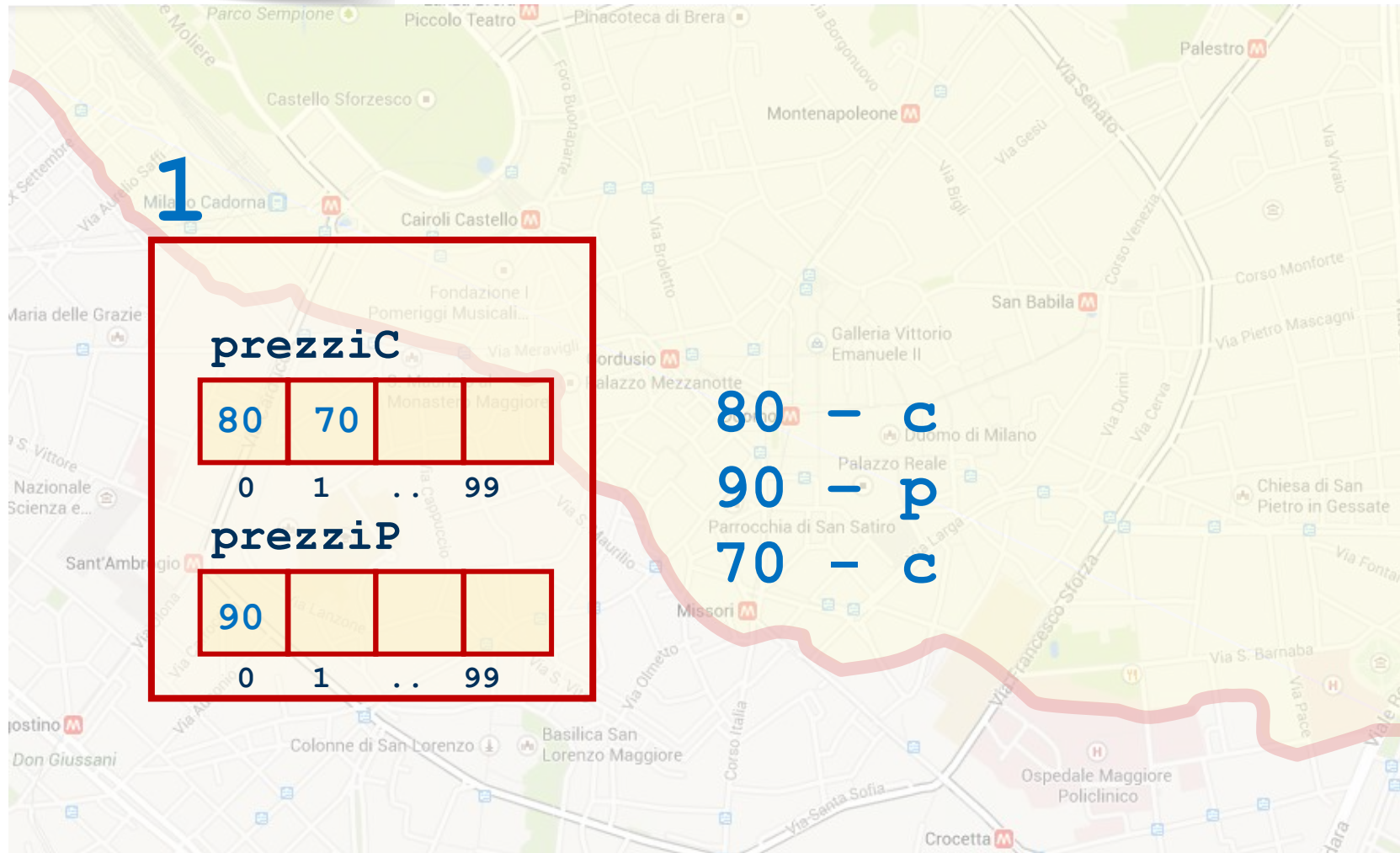
Memorizzazione



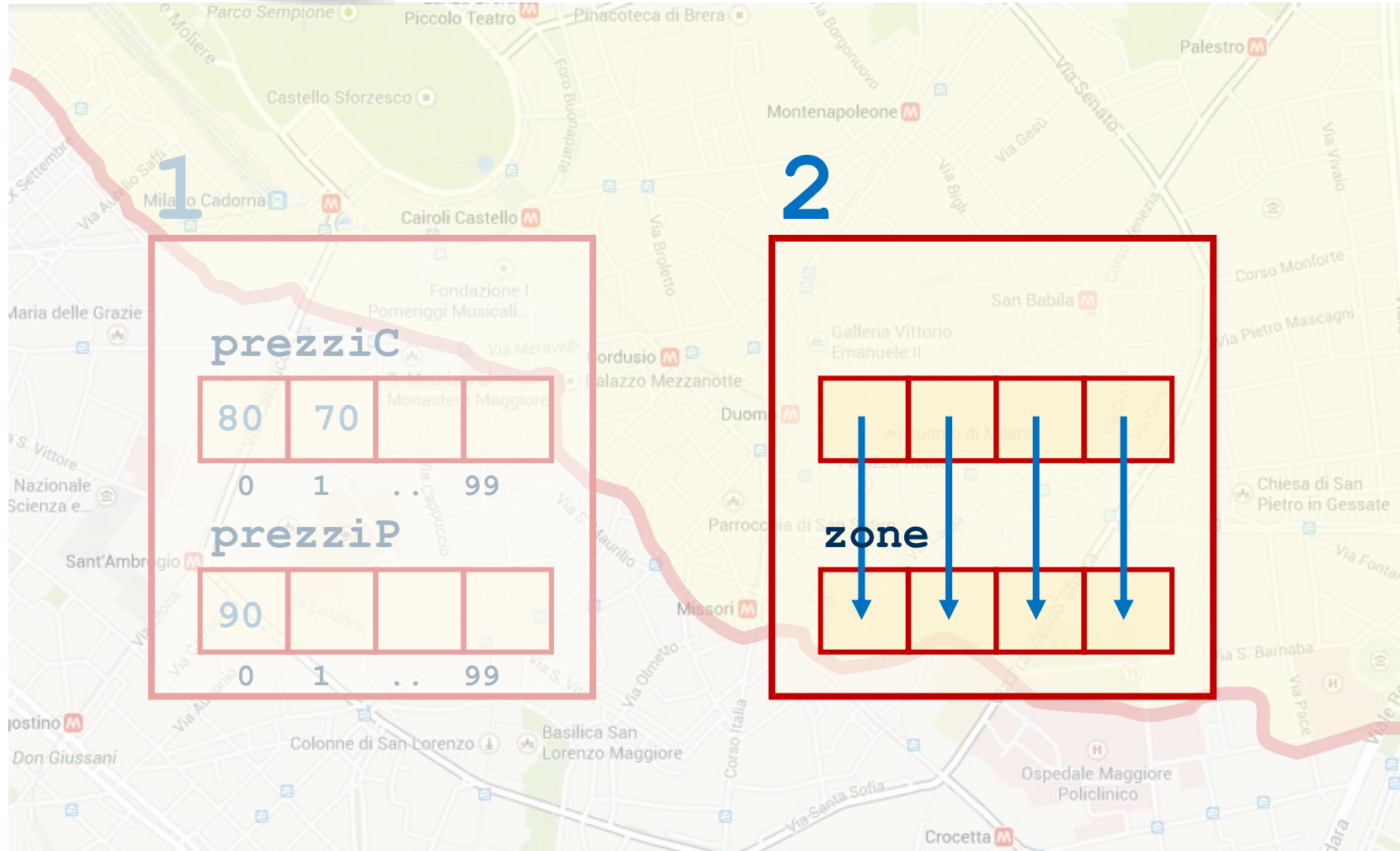
Memorizzazione



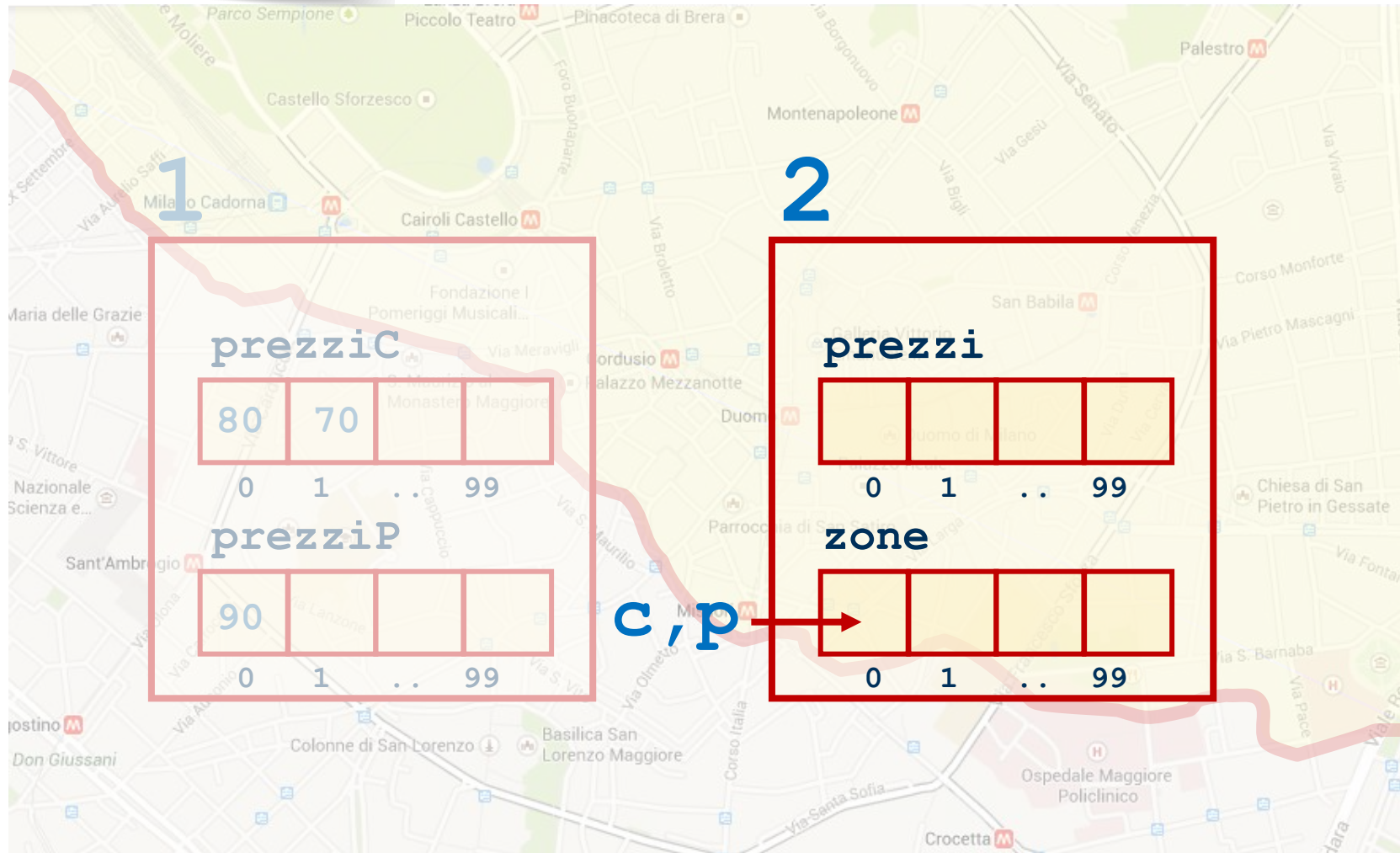
Memorizzazione



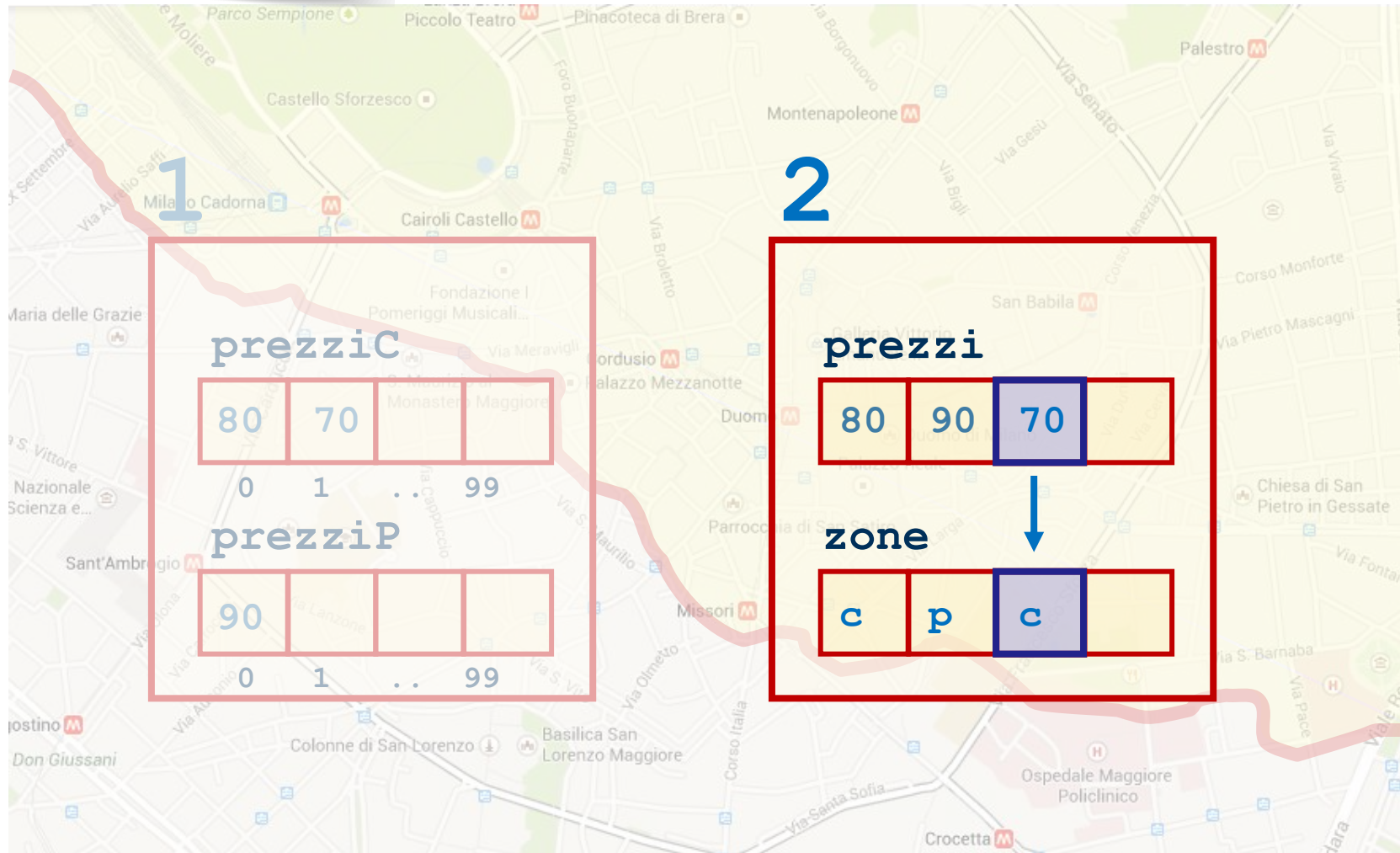
Memorizzazione



Memorizzazione



Memorizzazione



rilevazioni zona centro

prezzi rilevati	deviazione rispetto al prezzo medio
80	5.00
70	-5.00

rilevazioni zona periferia

prezzi rilevati	deviazione rispetto al prezzo medio
90	0.00

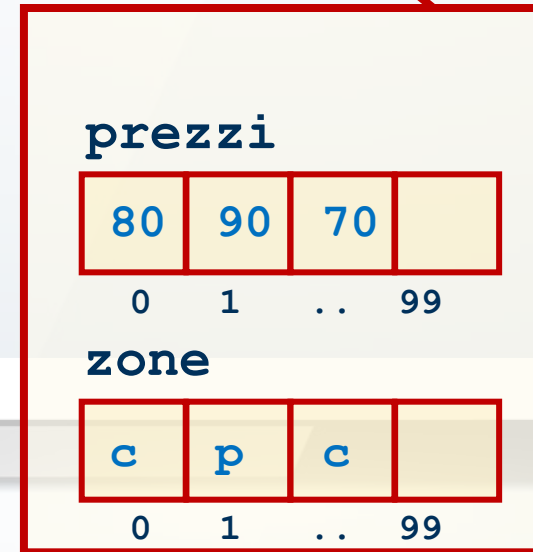
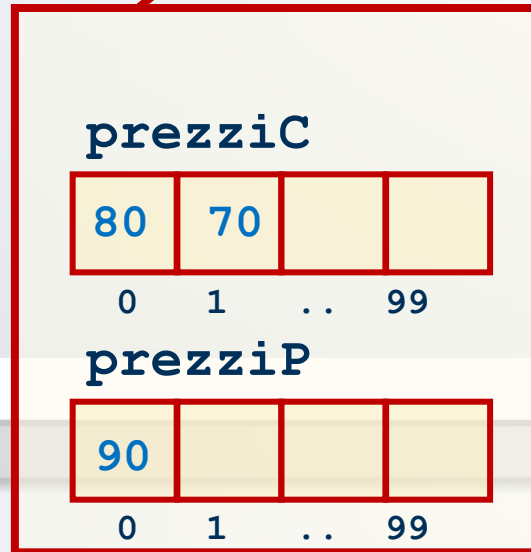
1

prezzi rilevati	deviazione rispetto al prezzo medio	zona
80	5.00	c
90	0.00	p
70	-5.00	c

2

Tutti i prezzi del centro
POI

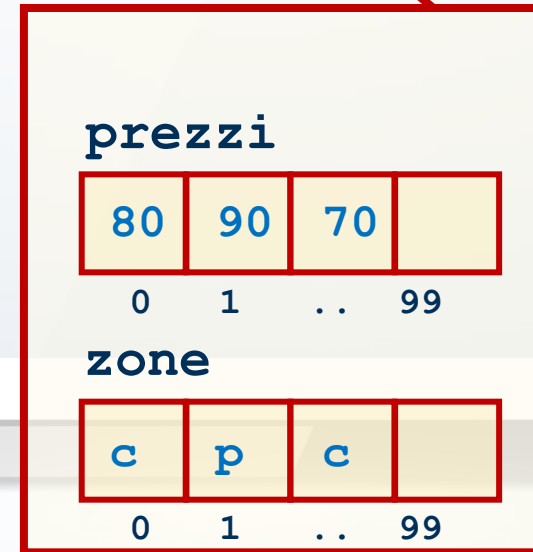
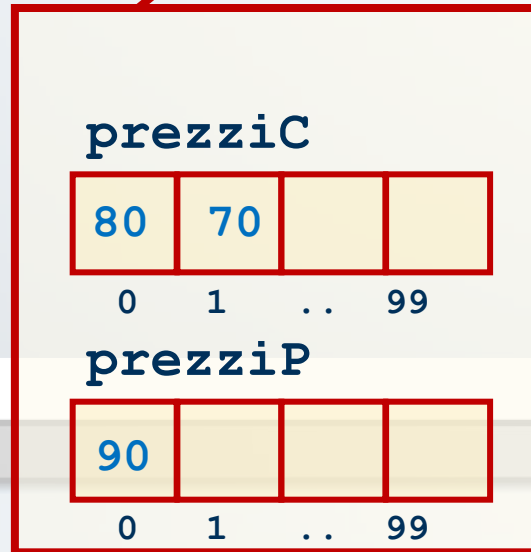
Tutti i prezzi della periferia



Tutti i prezzi
nello stesso ordine
dei dati in ingresso

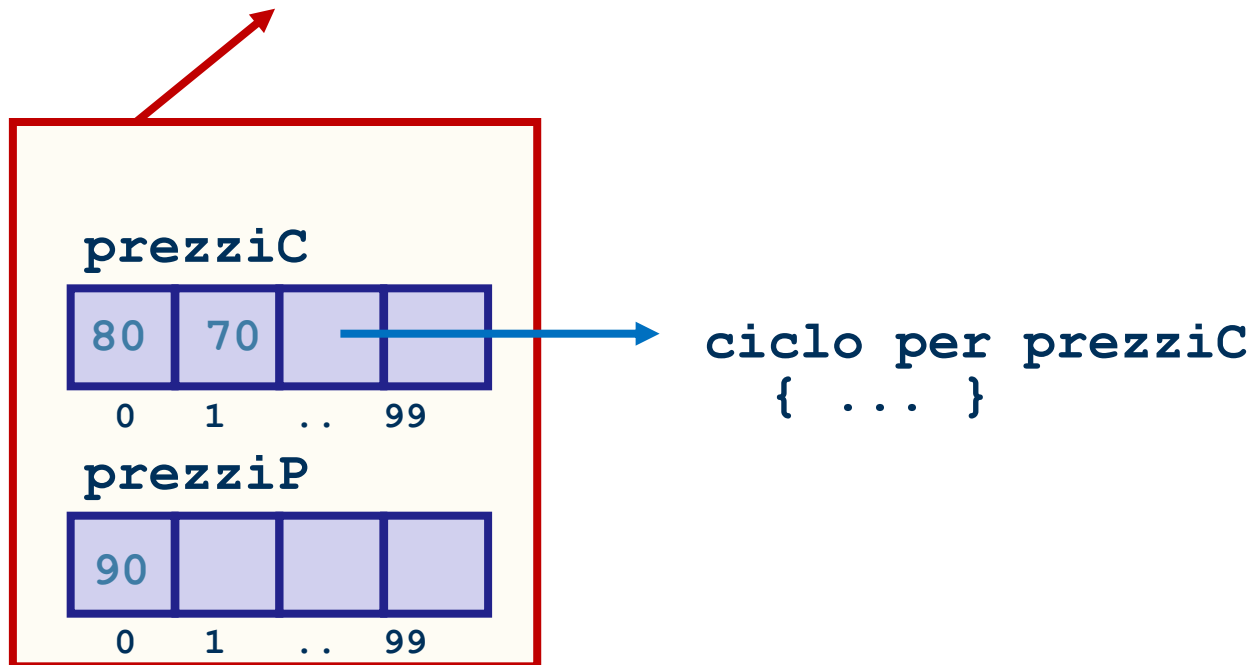
Tutti i prezzi del centro
POI

Tutti i prezzi della periferia

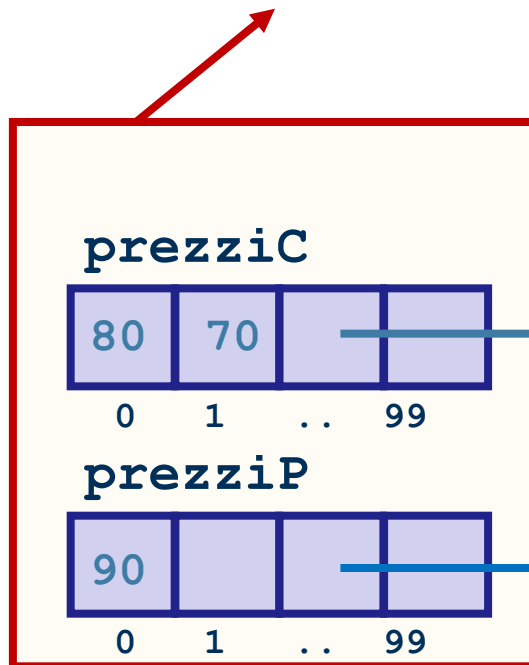


Tutti i prezzi
nello stesso ordine
dei dati in ingresso

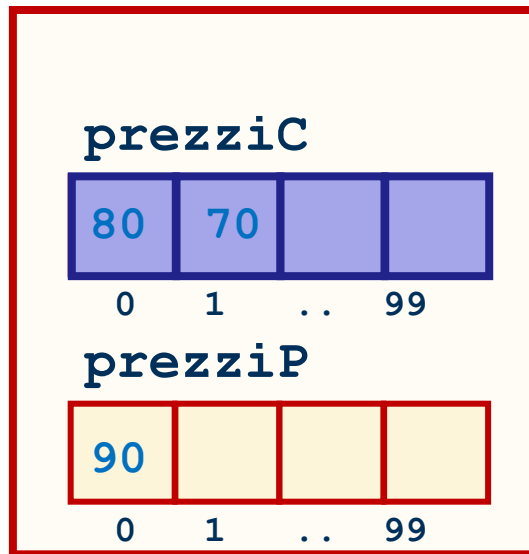
Tutti i prezzi del centro
POI
Tutti i prezzi della periferia



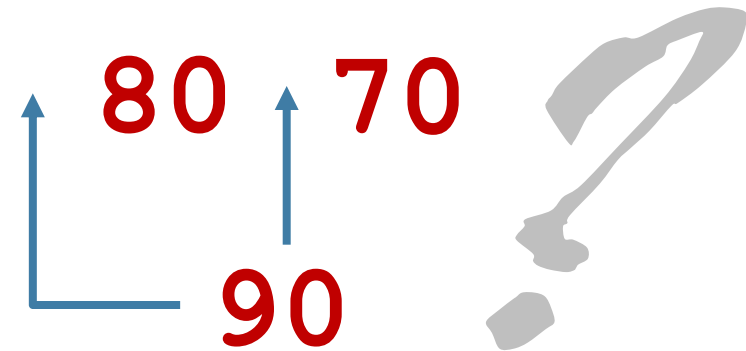
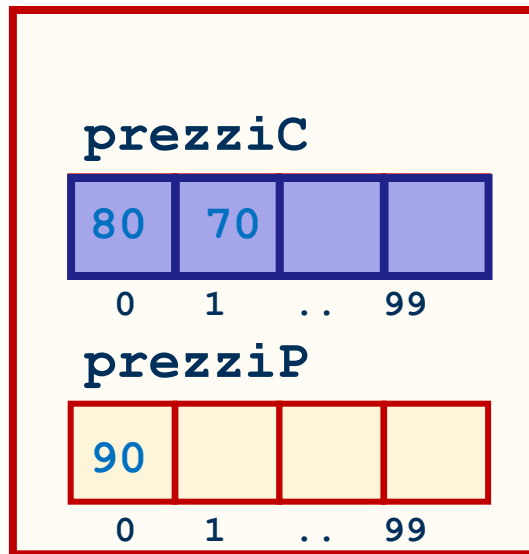
Tutti i prezzi del centro POI Tutti i prezzi della periferia



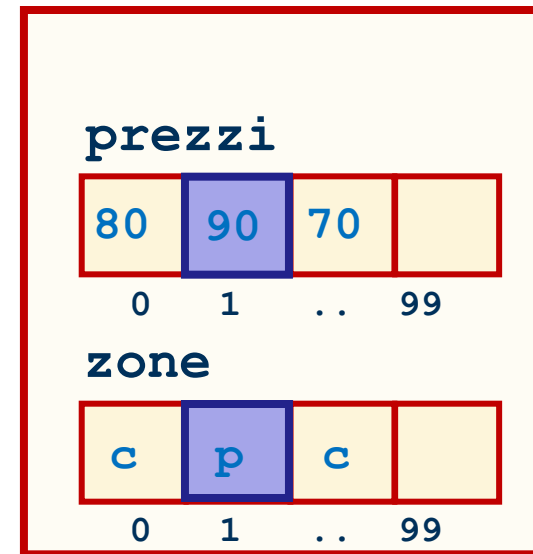
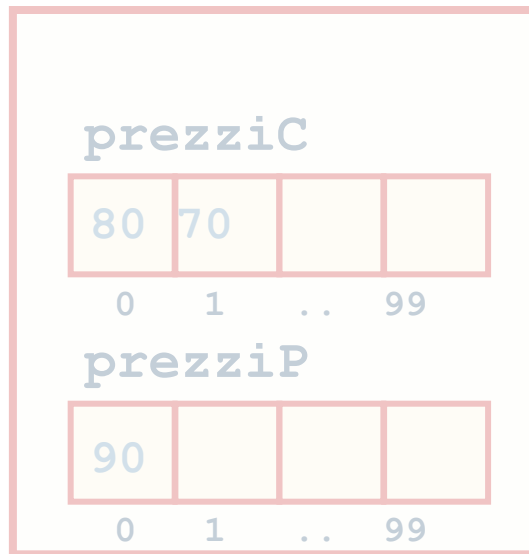
```
ciclo per prezziC  
{ per ogni elemento di prezziC  
  calcola scarto  
}  
ciclo per prezziP  
{ per ogni elemento di prezziP  
  calcola scarto  
}
```



Tutti i prezzi
nello stesso ordine
dei dati in ingresso

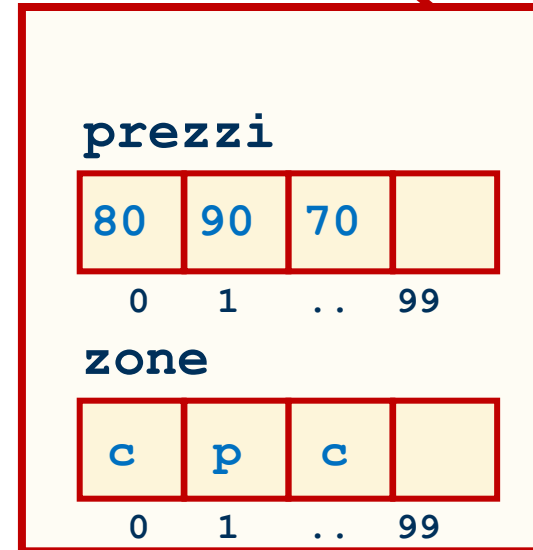


Tutti i prezzi
nello stesso ordine
dei dati in ingresso

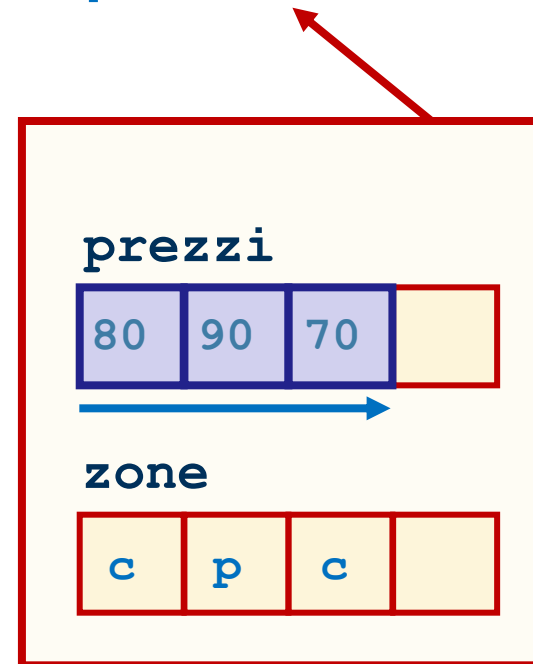


**Tutti i prezzi
nello stesso ordine
dei dati in ingresso**

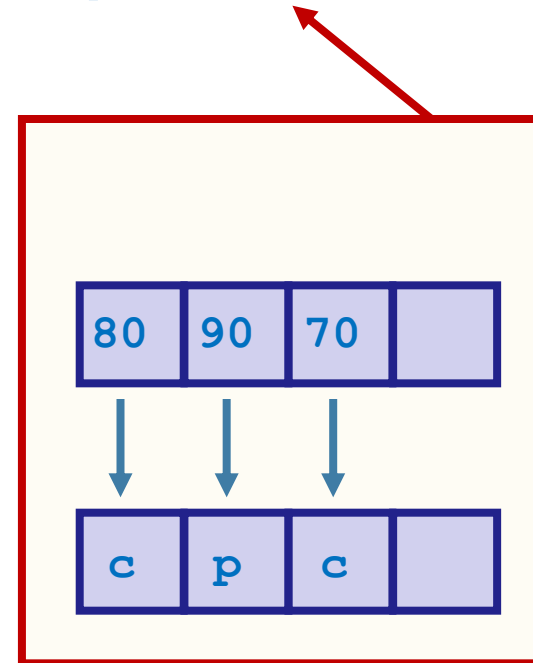
Tutti i prezzi del centro
POI
Tutti i prezzi della periferia



Tutti i prezzi del centro
POI
Tutti i prezzi della periferia

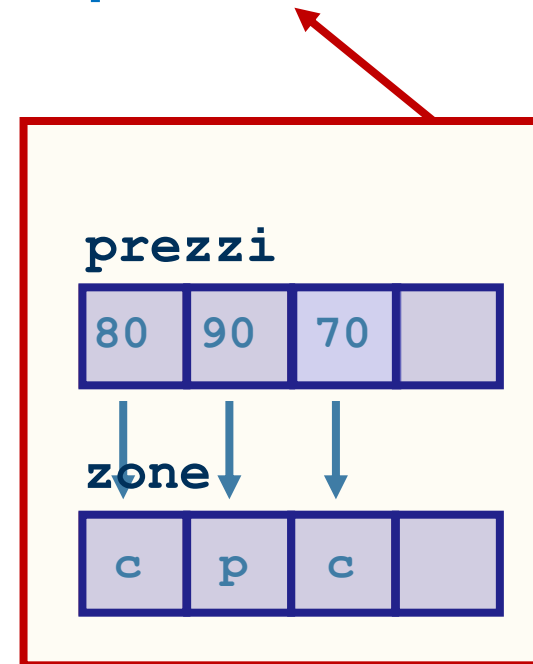


Tutti i prezzi del centro
POI
Tutti i prezzi della periferia



Tutti i prezzi del centro
POI
Tutti i prezzi della periferia

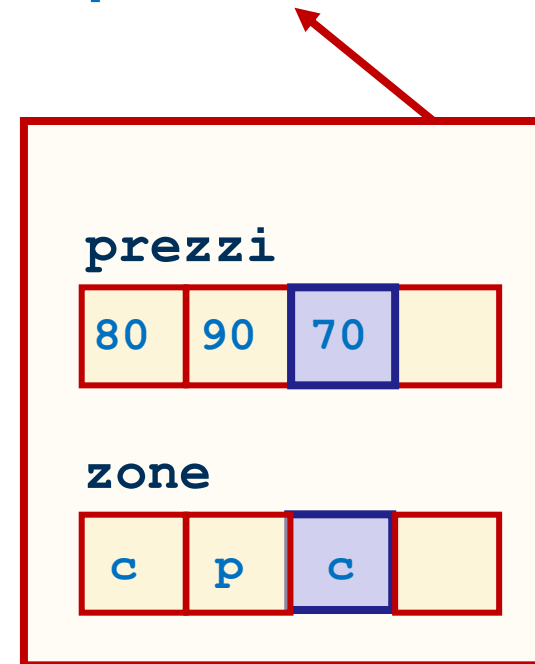
`prezzi[i]`



Tutti i prezzi del centro
POI
Tutti i prezzi della periferia

`prezzi[i]`

`zona[i]`

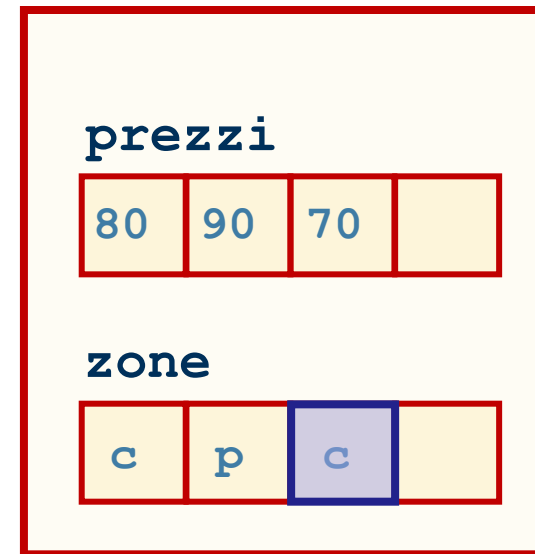


*se zona[i] == CENTRO
calcola lo scarto di prezzi[i]
e stampalo*

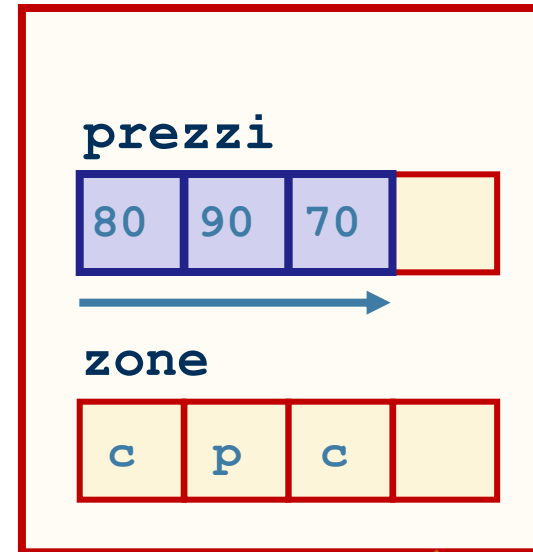
Tutti i prezzi del centro
POI
Tutti i prezzi della periferia

`prezzi[i]`

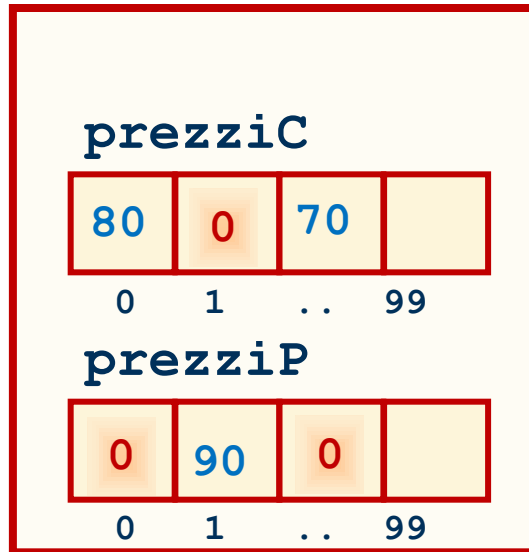
`zona[i]`



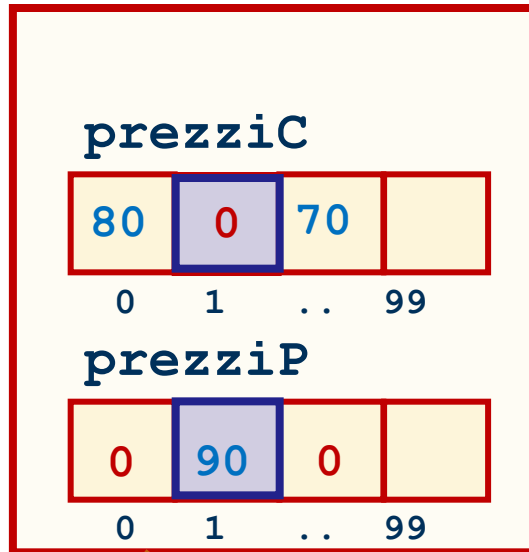
Tutti i prezzi
nello stesso ordine
dei dati in ingresso



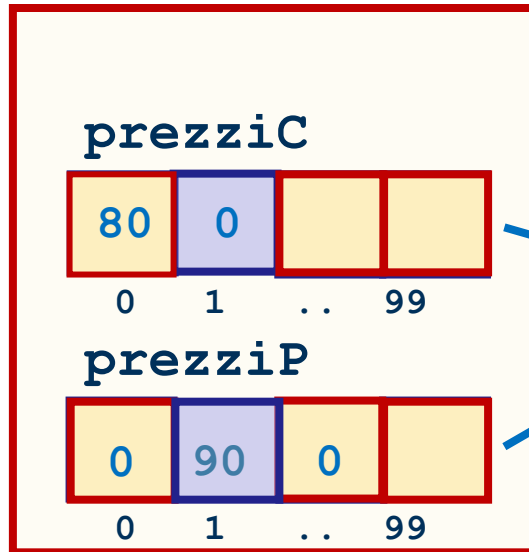
Tutti i prezzi
nello stesso ordine
dei dati in ingresso



Tutti i prezzi
nello stesso ordine
dei dati in ingresso



Tutti i prezzi
nello stesso ordine
dei dati in ingresso



*Scandisci `prezziC` e `prezziP`
e calcola lo scarto per
ogni valore $\neq 0$*

**Tutti i prezzi
nello stesso ordine
dei dati in ingresso**

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
int main ()
```

```
{ const char CENTRO
```

```
const int MAXDIM
```

```
char zona;
```

```
int prezzo;
```

```
prezziC[MAXDIM], prezziP[MAXDIM], // prezzi rilevati nelle due zone
```

```
numC, numP, sommaC, sommaP; // numero e somma dei prezzi delle due zone
```

```
i;
```

```
float media;
```

```
cout << setw(10) << "Iniziamo le rilevazioni" << endl;
```

```
numC = 0; numP = 0;
```

```
while(1) //
```

```
{ cout << "Inserisci la zona e il prezzo per ogni rilevazione: "; cin >> zona;
```

```
if (cin.get() != '\n')
```

```
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++; }
```

Tutti i prezzi del centro

POI

Tutti i prezzi della periferia

prezziC

80

70

0

1

..

99

prezziP

90

0

1

..

99

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone
```

```
i;
```

```
float media, deviazione;
```

```
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++; }
```

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone
```

```
i;
```

```
float media, deviazione;
```

```
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++; }
```

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone
```

```
i;
```

```
float media, deviazione; const char CENTRO = 'c';
```

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

```
cout << "prezzi rilevati";
```

```
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++; }
```



```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone
```

```
  i;
```

```
  float media, deviazione;
```

```
  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++; }
```

```
}  
  
if (numC != 0)          // se c'è almeno una rilevazione della zona centro  
{cout << endl << setw(20)  
  << "rilevazione zona "          // crea intestazione  
  << setw(10) << "centro" << endl << endl          // tabella  
  << setw(20) << "prezzi rilevati"          // per zona centro  
  << setw(40) << "deviazione rispetto al prezzo medio" << endl;  
  media = (float)sommaC / numC; // ne calcola il prezzo medio  
  for (i = 0; i < numC; i++)      // per ogni rilevazione, stampa una  
  { deviazione = prezzi[i] - media; // riga contenente il  
    cout << setw(20) << prezziC[i] // prezzo rilevato e la  
      << setw(40) << setprecision(2) // deviazione rispetto alla  
      << setiosflags (ios::fixed | ios::showpoint) // media, in  
      << deviazione // formato fisso, sempre con il punto  
      << endl; // decimale e due cifre decimali  
  }  
}  
  
if (numP != 0)          // idem per la periferia  
{cout << endl << setw(20) << "rilevazione 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 < numC; i++)  
  { deviazione = prezzi[i] - media;  
    cout << setw(20) << prezziP[i]  
      << setw(40) << setprecision(2)
```

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone
  i;
  float media, deviazione;

  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++; }
    }
  }
}
```

```
// Analisi del prezzo di un prodotto in una città divisa in due zone.
```

```
#include <iomanip.h>
```

```
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 e somma dei prezzi delle due zone  
  i;  
  float media, deviazione;
```

```
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))
```

acquisizione rilevazioni

```
numC,numP,sommaC,sommaP, // numero e somma dei prezzi delle due zone
i;
float media, deviazione;

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";
}

if (numC != 0) // se c'è almeno una rilevazione della zona centro
{ cout << endl << setw(20)
    << "rilevazione    zona " // crea intestazione
    << setw(10) << "centro" << endl << endl // tabella
    << setw(20) << "prezzi rilevati" // per zona centro
```

```
numC,numP,sommaC,sommaP, // numero e somma dei prezzi delle due zone
i;
float media, deviazione;

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";
}

if (numC != 0) // se c'è almeno una rilevazione della zona centro
{ cout << endl << setw(20)
  << "rilevazione    zona " // crea intestazione
  << setw(10) << "centro" << endl << endl // tabella
  << setw(20) << "prezzi rilevati" // per zona centro
```

```
numC,numP,sommaC,sommaP, // numero e somma dei prezzi delle due zone
i;
float media, deviazione;

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";
}

if (numC != 0) // se c'è almeno una rilevazione della zona centro
{ cout << endl << setw(20)
    << "rilevazione    zona " // crea intestazione
    << setw(10) << "centro" << endl << endl // tabella
    << setw(20) << "prezzi rilevati" // per zona centro
```



```
if (zona == CENIRO || 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++; }
}
```

Diagram illustrating the execution of a C++ program. A variable `prezziC` is shown above a horizontal array of 10 yellow cells. The first cell contains the value `80`. Below the array, a code snippet is shown with line numbers 0 to 99. The code includes a loop that prints "prezzi rilevati" and then iterates over the array, printing each value. The variable `numC` is shown below the code, indicating the current index.

```

0   << "rilevazione zona " // crea stazione
1   << setw(10) << "centro" << endl << endl // tabella
2   << setw(20) << "prezzi rilevati" // per zona centro
...
98
99

```

`numC numC`

```
numC,numP,sommaC,sommaP, // numero e somma dei prezzi delle due zone
i;
float media, deviazione;

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";
}

if (numC != 0) // se c'è almeno una rilevazione della zona centro
{ cout << endl << setw(20)
    << "rilevazione    zona " // crea intestazione
    << setw(10) << "centro" << endl << endl // tabella
    << setw(20) << "prezzi rilevati" // per zona centro
```

```
else
    cout << "zona non corretta: inserire nuovo valore";
}

if (numC != 0)          // se c'è almeno una rilevazione della zona centro
{cout << endl << setw(20)
    << "rilevazione zona "          // crea intestazione
    << setw(10) << "centro" << endl << endl          // tabella
    << setw(20) << "prezzi rilevati"          // per zona centro
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float)sommaC / numC;          // ne calcola il prezzo medio
    for (i = 0; i < numC; i++)          // per ogni rilevazione, stampa una
    { deviazione = prezziC[i] - media;    // riga contenente il
      cout << setw(20) << prezziC[i]      // prezzo rilevato e la
        << setw(40) << setprecision(2) // deviazione rispetto alla
        << setiosflags (ios::fixed | ios::showpoint) // media, in
        << deviazione // formato fisso, sempre con il punto
        << endl;          // decimale e due cifre decimali
    }
}

if (numP != 0)          // idem per la periferia
{cout << endl << setw(20) << "rilevazione 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)
```

```

else
    cout << "zona non corretta: inserire nuovo valore";
}
    prezziC
if { 80 70 [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] zona centro
{
    << "rilevazione zona " // crea intestazione
    << setw(10) << "centro" << endl << endl // tabella
    << setw(20) << "prezzi rilevati" // per zona centro
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float) 90
    for (i = 0; i < 10; i++)
    { deviazione = prezzi[i] - media; // riga contenente il
      cout << setw(20) << prezziC[i] // prezzo rilevato e la
        << setw(40) << setprecision(2) // deviazione rispetto alla

```

rilevazioni zona	centro	
prezzi rilevati	deviazione rispetto al prezzo medio	
80		5.00
70		-5.00
rilevazioni zona	periferia	
prezzi rilevati	deviazione rispetto al prezzo medio	
90		0.00

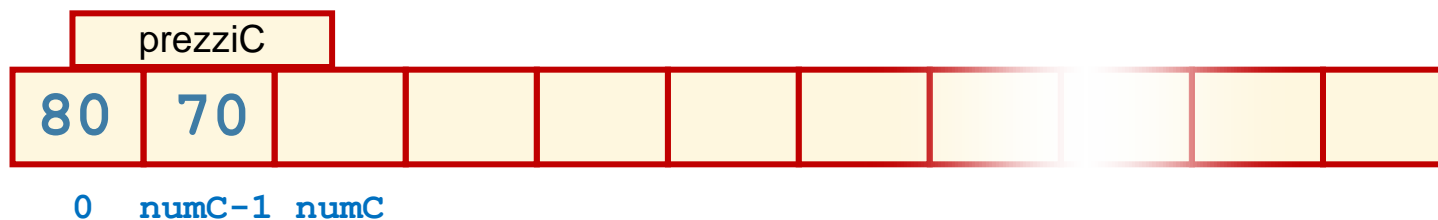
```
else
    cout << "zona non corretta: inserire nuovo valore";
}

if (numC != 0)          // se c'è almeno una rilevazione della zona centro
{
    cout << endl << setw(20)
        << "rilevazioni      zona "      // crea intestazione
        << setw(10) << "centro" << endl << endl      // tabella
        << setw(20) << "prezzi rilevati"      // per zona centro
        << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float)sommaC / numC;      // ne calcola il prezzo medio
    for (i = 0; i < numC; i++)      // per ogni rilevazione, stampa una
    {
        deviazione = prezziC[i] - media;      // riga contenente il
        cout << setw(20) << prezziC[i]      // prezzo rilevato e la
            << setw(40) << setprecision(2) // deviazione rispetto alla
            << setiosflags (ios::fixed | ios::showpoint) // media, in
            << deviazione      // formato fisso, sempre con il punto
            << endl;      // decimale e due cifre decimali
    }
}

if (numP != 0)          // idem per la periferia
{
    cout << endl << setw(20) << "rilevazione 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)
```

```
else
    cout << "zona non corretta: inserire nuovo valore";
}

if (numC != 0)          // se c'è almeno una rilevazione della zona centro
{cout << endl << setw(20)
    << "rilevazioni      zona "      // crea intestazione
    << setw(10) << "centro" << endl << endl      // tabella
    << setw(20) << "prezzi rilevati"      // per zona centro
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float)sommaC / numC;      // ne calcola il prezzo medio
    for (i = 0; i < numC; i++)      // per ogni rilevazione, stampa una
    { deviazione = prezziC[i] - media; // riga contenente il
      cout << setw(20) << prezziC[i]      // prezzo rilevato e la
        << setw(40) << setprecision(2) // deviazione rispetto alla
        << setiosflags (ios::fixed | ios::showpoint) // media, in
        << deviazione // formato fisso, sempre con il punto
        << endl;      // decimale e due cifre decimali
    }
}
```



```
else
    cout << "zona non corretta: inserire nuovo valore";
}

if (numC != 0)          // se c'è almeno una rilevazione della zona centro
{cout << endl << setw(20)
    << "rilevazioni      zona "      // crea intestazione
    << setw(10) << "centro" << endl << endl      // tabella
    << setw(20) << "prezzi rilevati"      // per zona centro
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float)sommaC / numC;      // ne calcola il prezzo medio
    for (i = 0; i < numC; i++)      // per ogni rilevazione, stampa una
    { deviazione = prezzi[i] - media; // riga contenente il
        cout << setw(20) << prezziC[i]      // prezzo rilevato e la
            << setw(40) << setprecision(2) // deviazione rispetto alla
            << setiosflags (ios::fixed | ios::showpoint) // media, in
            << deviazione      // formato fisso, sempre con il punto
            << endl;          // decimale e due cifre decimali
    }
}

if (numP != 0)          // idem per la periferia
{cout << endl << setw(20) << "rilevazione 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)
```




```
else
    cout << "zona non corretta: inserire nuovo valore";
}

if (numC != 0)          // se c'è almeno una rilevazione della zona centro
{cout << endl << setw(20)
    << "rilevazioni      zona "      // crea intestazione
    << setw(10) << "centro" << endl << endl      // tabella
    << setw(20) << "prezzi rilevati"      // per zona centro
    << setw(40) << "deviazione rispetto al prezzo medio" << endl;
    media = (float)sommaC / numC;      // ne calcola il prezzo medio
    for (i = 0; i < numC; i++)      // per ogni rilevazione, stampa una
    { deviazione = prezziC[i] - media; // riga contenente il
      cout << setw(20) << prezziC[i]      // prezzo rilevato e la
        << setw(40) << setprecision(2) // deviazione rispetto alla
        << setiosflags (ios::fixed | ios::showpoint) // media, in
        << deviazione      // formato fisso, sempre con il punto
        << endl;          // decimale e due cifre decimali
    }
}

if (numP != 0)          // idem per la periferia
{cout << endl << setw(20) << "rilevazione 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)
```

```
if (numP != 0) // idem per la periferia
{
    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 (numP != 0) // idem per la periferia
{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;
  }
}
```



rilevazioni zona centro

prezzi rilevati deviazione rispetto al prezzo medio

80	5.00
----	------

70	-5.00
----	-------

rilevazioni zona periferia

prezzi rilevati deviazione rispetto al prezzo medio

```
if (numP != 0) // idem per la periferia
{
    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 (numP != 0) // idem per la periferia
{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;
  }
}
```

mediaC**mediaP**

```
if (numP != 0) // idem per la periferia
{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;
  }
}
```