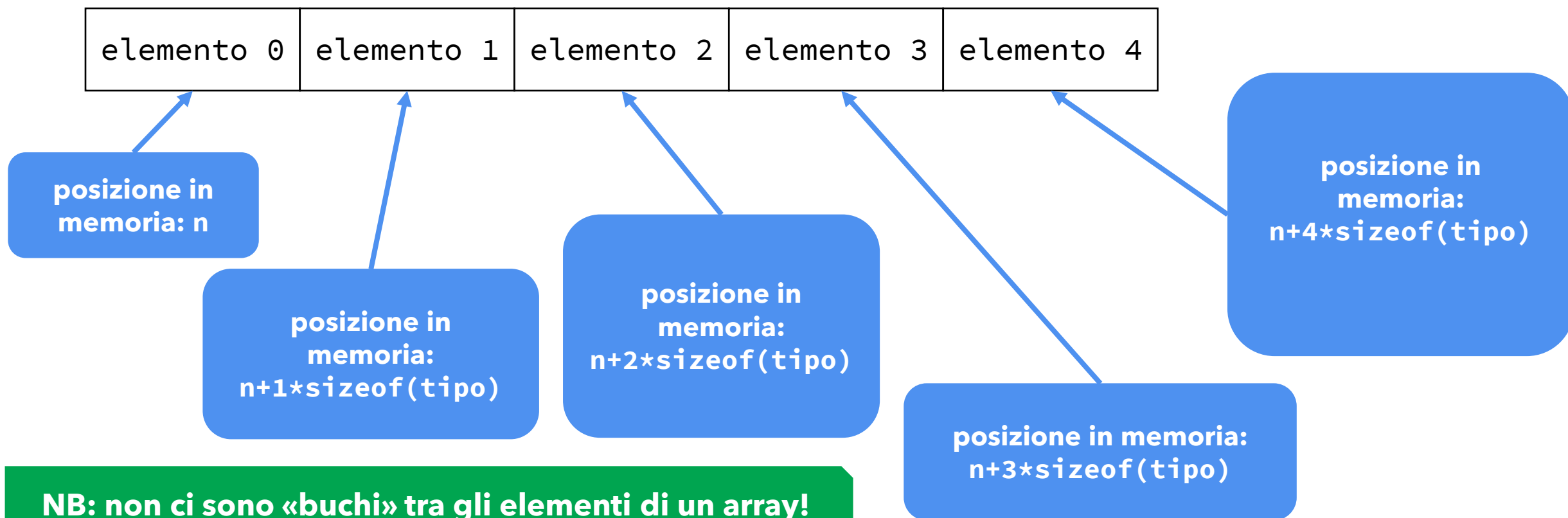


Gli array (vettori) in C/C++

Liceo G.B. Brocchi - Bassano del Grappa (VI)
Liceo Scientifico - opzione scienze applicate
Giovanni Mazzocchin

Gli array in C/C++

- Un array è una sequenza di oggetti omogenei (dello **stesso tipo**, e quindi della stessa dimensione in byte) allocati in posizioni di memoria contigue



Gli array in C/C++

```
int vect_in[20];           //array of 20 integers  
vect_in[0];                //array's first element (at index 0)
```

008FFBA4: indirizzo di memoria di vect_in[0]

008FFBA8: indirizzo di memoria di vect_in[1]

008FFBAC: indirizzo di memoria di vect_in[2]

008FFBB0: indirizzo di memoria di vect_in[3]

008FFBB4: indirizzo di memoria di vect_in[4]

008FFBB8: indirizzo di memoria di vect_in[5]

008FFBBC: indirizzo di memoria di vect_in[6]

. . .
. . .

la differenza tra l'indirizzo dell'elemento i-esimo e l'indirizzo dell'elemento (i-1)-esimo è 4 perché un int su questa macchina occupa 4 byte

Gli array in C/C++

```
char vect_ch[10];           //array of 10 characters
```

010FFE94: indirizzo di memoria di vect_ch[0]

010FFE95: indirizzo di memoria di vect_ch[1]

010FFE96: indirizzo di memoria di vect_ch[2]

010FFE97: indirizzo di memoria di vect_ch[3]

• • •

• • •

**analizzare gli indirizzi e dire quanta memoria
occupa un char**

Gli array in C/C++

```
int size = 5;  
int v[size];
```

- non compila: la dimensione di un array non può essere una variabile, perché deve essere nota a *compile time* (al momento della compilazione del programma)
-

```
const int size = 5;  
int v[size];
```

- compila: la dimensione dell'array in questo caso è una **costante** nota a *compile time*

Gli array in C/C++

```
int ai[30];
```

- cosa conterrà la memoria allocata per l'array ai?



```
0 5242392 13 5242600 14598928 -334170950 -2 5242476 14631564 5242464 5242456 5242468
5242484 5242488 2 2 -323339802 5242492 14631425 14768304 14700998 5242504 14587344
14700998 5242516 14765836 5242524 14553135 0 5242548
```

**memoria non inizializzata. Ci sono i valori che
c'erano prima della dichiarazione nelle stesse
locazioni di memoria**

Gli array in C/C++

```
char ac[] = {'h', 'e', 'l', 'l', 'o'};
```

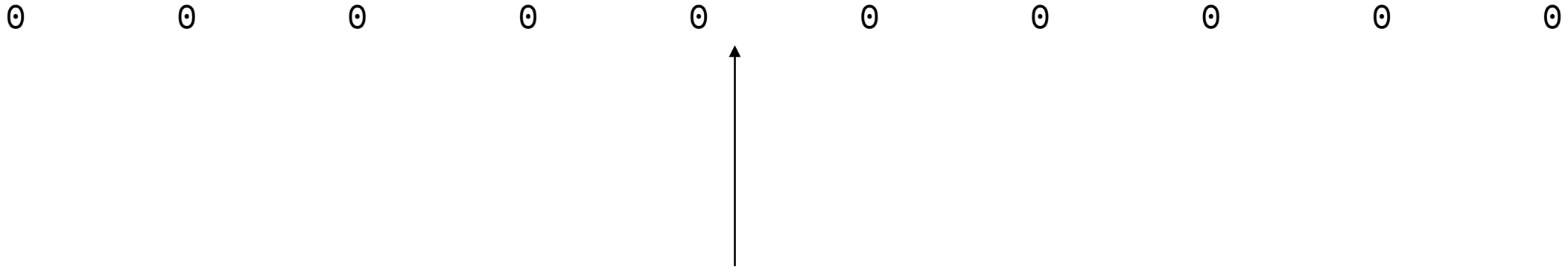
'h'	'e'	'l'	'l'	'o'
-----	-----	-----	-----	-----

il compilatore calcola la dimensione dell'array sulla base del numero di elementi della lista di inizializzazione {...}

Gli array in C/C++

```
char ac[5] = {'h', 'e', 'l', 'l', 'o'};
```

```
int vi[10] = {};
```



contenuto dell'array. I 10 elementi vengono inizializzati a 0 grazie all'inizializzatore {}

Gli array in C/C++

```
double vi[5] = {4.5, 6.0, 3.2, 30.2, 3.43, 3.14, 6.28};
```



**perché non
compila?**

lecture2.cpp(134): error C2078: troppi inizializzatori

Gli array in C/C++

```
int vi[8] = {1, 3, 2, 4};
```

1

3

2

4

0

0

0

0

Gli array in C/C++

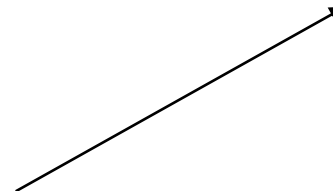
```
int vi[] = { 21, 12, 6, 8 };
```

21	12	6	8
vi[0]	vi[1]	vi[2]	vi[3]

```
cout << vi;
```



0093FD04



perché questo output?

Gli array in C/C++

```
int vi[] = {21, 12, 6, 8};  
  
for (int i = 0; i < 4; i++) {  
    cout << vi[i] << '\t';  
}
```



ecco come si stampa il contenuto di un array

Gli array in C/C++

```
int vi[] = {21, 12, 6, 8};
```

```
for (int i = 0; i < 30; i++) {  
    cout << vi[i] << '\t';  
}
```



compila! In C/C++ non c'è alcun controllo sull'indice utilizzato per accedere ad un elemento di un array. Ecco l'output:

21	12	6	8	-1461440336	11532736	2921131	1	
11893664		11954752		-1461440504	2921267	2921267	9383936	0
0	11532684		0	11532824	2933408	-1469777448	0	
11532752		1983932217		9383936	1983932192	11532840		2005176274
9383936								

Gli array in C/C++

- Scrivere un programma che somma e memorizza in una variabile tutti gli elementi maggiori di 7 di un array di interi senza segno (utilizzare un ciclo for)
- Scrivere un programma che memorizza in una variabile l'elemento massimo di un array di interi con segno
- Scrivere un programma che memorizza in una variabile l'elemento minimo di un array di interi con segno
- Scrivere un programma che calcola e memorizza i numeri di Fibonacci (fino al 9-esimo)