

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main(int argc, char *argv[])
{
    int freq[256]; // vettore di contatori
    della frequenza delle lettere della parola "x"
    char parola[100];
    int i, len, lunghezza;
    int r;
    ...
}

// ...

return 0;
}
```

## Vettori

## I vettori in C

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main(int argc, char *argv[])
{
    int freq[256]; // vettore di contatori
    della frequenza delle lettere della parola "x"
    char parola[100];
    int i, len, lunghezza;
    int r;
    ...
}

// ...

return 0;
}
```

## I vettori in C

## Sintassi della definizione

```
(argc, argv)
{
    ...
}
```

## I vettori in C

- Sintassi della definizione
- Definizione di costanti
- Operazioni di accesso

2

```
(argc, argv)
{
    ...
}
```

## Definizione di vettori in C

```
int dato[10] ;
```

- Tipo di dato base
- Nome del vettore
- Numero di elementi

4

```
(argc, argv)
{
    ...
}
```

## Definizione di vettori in C

```
int dato[10] ;
```

- Tipo di dato base
- Nome del vettore
- Numero di elementi

- Stesse regole che valgono per i nomi delle variabili.
- I nomi dei vettori devono essere diversi dai nomi delle variabili.

5

```
(argc, argv)
{
    ...
}
```

## Definizione di vettori in C

```
int dato[10] ;
```

- Tipo di dato base
- Nome del vettore
- Numero di elementi

- int
- float
- In futuro vedremo: char, struct

6

## Definizione di vettori in C

```
int dato[10] ;
```

Tipo di dato  
base

Nome del  
vettore

Numero di  
elementi

- Intero positivo
- Costante nota a tempo di compilazione
- Impossibile cambiarla in seguito

7

## Esempi

- » `int dato[10] ;`
- » `float lati[8] ;`
- » `int numeriprimi[100] ;`
- » `int is_primo[1000] ;`

8

## Esempi

- » `int dato[10] ;`
- » `float lati[8] ;`
- » `int numeriprimi[100] ;`
- » `int is_primo[1000] ;`

0	1
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19
9	23
10	29
11	31
12	37
...	...

9

## Esempi

- » `int dato[10] ;`
- » `float lati[8] ;`
- » `int numeriprimi[100] ;`
- » `int is_primo[1000] ;`

0	0
1	1
2	1
3	1
4	0
5	0
6	0
7	1
8	0
9	0
10	0
11	1
12	0
...	...

10

## Errore frequente

- » Dichiarare un vettore usando una variabile anziché una costante

```
int N = 10 ;
int dato[N] ;
```

```
int dato[10] ;
```

11

## Errore frequente

- » Dichiarare un vettore usando una variabile non ancora inizializzata

```
int N ;
int dato[N] ;
```

```
...
```

```
scanf("%d",&N) ;
```

```
int dato[10] ;
```

12



## Errore frequente

- Dichiarare un vettore usando il nome dell'indice

```
int i ;
int dato[i] ;

for(i=0; i<10; i++)
    scanf("%d",&dato[i]) ;
```



```
int dato[10] ;
```

13

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

#define MAXPAROLA 30
#define MAXRIGA 80

int main(int argc, char *argv[])
{
    int lunghezza;
    char parola[MAXPAROLA];
    char riga[MAXRIGA];
    int i;

    printf("Inserisci una parola: ");
    fgets(parola, MAXPAROLA, stdin);
    lunghezza = strlen(parola);

    printf("Inserisci una riga: ");
    fgets(riga, MAXRIGA, stdin);

    return 0;
}
```

## I vettori in C

## Definizione di costanti

## Costanti

- La dimensione di un vettore deve essere specificata utilizzando una costante intera positiva
  - **Costante** = valore numerico già noto al momento della compilazione del programma

```
int dato[10] ;
```

15

## Problemi

```
int i ;
int dato[10] ;

. . .

for(i=0; i<10; i++)
    scanf("%d", &dato[i]) ;

for(i=0; i<10; i++)
    printf("%d\n", dato[i]) ;
```

Devono essere  
tutte uguali.  
Chi lo  
garantisce?

16

## Problemi

```
int i ;
int dato[10] ;

. . .

for(i=0; i<10; i++)
    scanf("%d", &dato[i]) ;

for(i=0; i<10; i++)
    printf("%d\n", dato[i]) ;
```

Se volessi  
lavorare con  
20 dati dovrei  
modificare in  
tutti questi  
punti.

17

## Soluzione

- Per risolvere i problemi visti si può ricorrere alle **costanti simboliche**
  - Associamo un nome simbolico ad una costante
  - Nel programma usiamo sempre il nome simbolico
  - Il compilatore si occuperà di sostituire, ad ogni occorrenza del nome, il valore numerico della costante

18