

Esercitazione 3

Array, Stringhe

Informatica A - Ingegneria Matematica (sez. M-Z)

09 ottobre 2023



switch-case, la sintassi

switch, case, default

keywords

int_expr espressione a valori
integral (char o int)

constant-expr1 numero o
carattere

default opzionale

```
switch (int_expr)
{
  case constant-expr1:
    statement1
  case constant-expr2:
    statement2
    ...
  case constant-exprN:
    statementN
  [default : statement]
}
```



switch-case, la sintassi

NB: `constant-expr1` non può contenere una variabile,

NB: `int_expr` può contenere variabili

NB: a differenza di `if`, `while` e `for`,

- `int_expr` non è un'espressione booleana
- Non occorre delimitare gli `statement` tra `{ }`, anche nel caso contengano più istruzioni. Questi sono delimitati dal case seguente

```
switch (int_expr)
{
  case constant-expr1:
    statement1
  case constant-expr2:
    statement2
  ...
  case constant-exprN:
    statementN
  [default : statement]
}
```



switch-case, l'esecuzione

1. Viene valutata **expression** (eventualmente convertita)
2. Si controlla se **expression** è uguale a **constant-expr1**
3. Se sono uguali eseguo **statement1**, ed in cascata, tutti gli **statement** dei **case** seguenti (senza verifiche, incluso lo **statement** di **default**)
4. Altrimenti controllo se **expression** è uguale a **constant-expr2** ...
5. Eseguo lo **statement** di **default** [se presente]

```
switch (int_expr)
{
  case constant-expr1:
    statement1
  case constant-expr2:
    statement2
    ...
  case constant-exprN:
    statementN
  [default : statement]
}
```



Note switch-case

Esempio di utilizzo di **switch**

```
scanf ("%c" , &a) ;  
switch (a)  
    {case 'A' : nA++;  
    case 'E' : nE++;  
    case 'O' : nO++;  
    default : nCons++;}
```

Se **a== 'A'** , verranno incrementate **nA , nE , nO , nCons** ;

Se **a== 'E'** , verranno incrementate **nE , nO , nCons** ;

Se **a== 'O'** , verranno incrementate **nO , nCons** ;

Se **a== 'K'** , verranno incrementa **nCons** ;



Note switch-case

Per evitare l'**esecuzione in cascata** alla prima corrispondenza trovata, occorre inserire negli statements opportuni la keyword **break**

```
scanf ("%c" , &a) ;  
switch (a)  
    { case 'A' : nA++; break ;  
      case 'E' : nE++; break ;  
      case 'O' : nO++; break ;  
      default : nCons++; }
```

Se **a == 'A'** , verrà incrementata **nA** ;

Se **a == 'E'** , verrà incrementata **nE** ;

Se **a == 'O'** , verrà incrementata **nO** ;

Se **a == 'K'** , verrà incrementa **nCons** ;

Esercizio 1

Scrivere un programma in C:

- Acquisire una sequenza di numeri fino a quando non viene inserito uno zero
- Il programma conta quanti sono i:
 - Multipli di 2
 - Multipli di 4
 - Multipli di 6
 - Multipli di 8
- Tra i numeri inseriti e stampa a schermo un istogramma

Esercizio 2

Scrivere un programma in C:

- Acquisire un numero
- Controllare che il numero sia primo
- Stampare l'esito del controllo

Esercizio 3

Scrivere un programma in C:

- Acquisire un array di numeri dall'utente
- Invertire la sequenza del primo array in un secondo array
- Stampare la sequenza invertita

Esercizio 4

Scrivere un programma in C:

- Acquisire una sequenza di numeri dall'utente
- Cercare le coppie di numeri per cui il primo è il doppio del secondo
- Stampare tali coppie

Esercizio 5

Scrivere un programma in C:

- Acquisire una sequenza di numeri dall'utente
- Costruire un insieme (no elementi ripetuti)
- Stampare l'insieme

Esercizio 6

Scrivere un programma in C:

- Acquisire due stringhe
- Concatenarle
- Stampare la stringa concatenata

Esercizio 7

Scrivere un programma in C:

- Acquisire una stringa
- Verificare se la stringa è palindroma

Esercizio 8

Scrivere un programma in C:

- Acquisire due stringhe
- Verificare se le stringhe sono anagrammi