



POLITECNICO
MILANO 1863

Esercitazione 3

Loop: For – While – Do While

SWITCH

Statement Switch per l'esecuzione di blocchi di codice

Switch Statement 1/3

- In C il costrutto **switch** è utilizzato per eseguire alcuni blocchi di codice in base al valore di un'espressione
- Utile quando ci sono molte condizioni basate sul valore di una singola variabile
- Sostituibile con una serie di **if-else**

Switch Statement 2/3

Keywords:

- **switch**
- **case**
- **default**

Espressioni:

- **int_expr**: espressione a valori int oppure char (non booleani)
- **constant_expr**: numero oppure carattere (non variabili)

Il caso **default** è opzionale

```
switch(int_expr){  
    case constant-expr1:  
        statement-1;  
    case constant-expr2:  
        statement-2;  
    ...  
    case constant-exprN:  
        statement-N;  
    [default:  
        statement;]  
}
```

Switch Statement 3/3

Gli **statement** di ogni caso non hanno bisogno di essere delimitati da **{}** anche nel caso in cui contengano più istruzioni

Gli **statement** di ogni **case** sono limitati dal **case** successivo (o dal **default**, se presente)

```
switch(int_expr){  
    case constant-expr1:  
        statement-1;  
    case constant-expr2:  
        statement-2;  
    ...  
    case constant-exprN:  
        statement-N;  
    [default:  
        statement;]  
}
```

Switch Statement: Esecuzione

1. Viene valutata **int_expr**
2. Si controlla se **int_expr** è uguale a **constant-expr-1**
 1. Se sono uguali viene eseguito **statement-1**, poi in cascata vengono eseguiti tutti gli altri statement relativi agli altri casi (senza verificare l'uguaglianza delle espressioni)
 2. Se NON sono uguali, si controlla se **int_expr** è uguale a **constant-expr-2** [...]
3. Si esegue lo **statement** nel caso **default**

Per evitare l'esecuzione in cascata, si può usare la keyword **break**!

```
switch(int_expr){  
    case constant-expr1:  
        statement-1;  
    case constant-expr2:  
        statement-2;  
    ...  
    case constant-exprN:  
        statement-N;  
    [default:  
        statement;]  
}
```

Switch: to break or not to break?

Che risultato ci aspettiamo?

```
int main(int argc, char* argv[]){
    char c;
    int nA = 0, nE = 0, nCons = 0;

    scanf("%c", &c);
    switch(c){
        case 'a':
            nA++;
            break;
        case 'e':
            nE++;
            break;
        default:
            nCons++;
            break;
    }

    printf("%d %d %d\n", nA, nE, nCons);
    return 0;
}
```

```
int main(int argc, char* argv[]){
    char c;
    int nA = 0, nE = 0, nCons = 0;

    scanf("%c", &c);
    switch(c){
        case 'a':
            nA++;
            // break;
        case 'e':
            nE++;
            // break;
        default:
            nCons++;
            // break;
    }

    printf("%d %d %d\n", nA, nE, nCons);
    return 0;
}
```

ESERCIZI

For – While – Do While

Esercizio Switch 1

Calcolatrice

Testo

Si scriva un programma che replichi il funzionamento di una calcolatrice. Implementare solamente le 4 operazioni: '+', '-', '*', '/'. Usare il costrutto switch.

Esercizio 1a

Inserimento Sequenziale di Caratteri con Criterio di Stop

Testo

Si scriva un programma che permetta di inserire all'utente un carattere alla volta. Ogni volta che l'utente inserisce un carattere il programma deve stampare:

1. il numero di caratteri inseriti finora;
2. l'ultimo carattere inserito.

Il programma termina quando l'utente inserisce il carattere '*'.

Esercizio 1b

Inserimento Sequenziale di Numeri in Ordine Ascendente

Testo

Scrivere un programma che consenta all'utente di inserire una sequenza (potenzialmente infinita) di numeri da tastiera. I numeri inseriti dall'utente devono rispettare un ordinamento strettamente crescente. Il programma termina non appena l'utente inserisce un numero che non rispetta la condizione di crescita.

Bonus

Stampare a schermo la sequenza inserita solamente dopo che si è usciti dal ciclo.

Esercizio 2

Somma dei Primi N Numeri Naturali

Testo

Si scriva un programma che permetta all'utente di inserire un numero intero positivo N. Il programma deve calcolare e stampare a schermo la somma dei primi N numeri naturali (senza usare la formula di Gauss).

Esempio

5 → 15

Esercizio 3

Calcolo delle Cifre di un Numero

Testo

Si scriva un programma che permetta all'utente di inserire un numero intero N in input (positivo o negativo). Il programma calcola il numero di cifre di N e lo stampa a schermo.

Esempio

1234 → 4

Esercizio 4

Inversione delle Cifre di un Numero

Testo

Si scriva un programma che permetta all'utente di inserire un numero intero positivo N in input. Il programma deve stampare a schermo il numero inserito dall'utente con tutte le cifre invertite.

Esempio

1234 → 4321

Esercizio 5

Tabella ASCII

Testo

Scrivere un programma in grado di stampare tutta la tabella ASCII (id e carattere corrispondente).

Esercizio 6

Numeri Triangolari

Testo

Si scriva un programma che verifichi se un numero intero positivo inserito dall'utente è triangolare. Il programma deve stampare a schermo se il numero è effettivamente triangolare o no.

Nota: un numero N si dice triangolare quando esiste un numero naturale T tale per cui $N = \sum_{i=1}^T i$.

Esempi

$10 \rightarrow OK$ $6 \rightarrow OK$ $5 \rightarrow NO$

Esercizio 7

Stampa delle Tabelline

Testo

Si scriva un programma in grado di stampare le tabelline (da 1 a 10) usando solo cicli while.

Bonus

Si scriva lo stesso programma usando solo cicli for.

Esercizio 8

Caratteri Compresi tra Due Estremi

Testo

Si scriva un programma in grado di ricevere in input due caratteri da parte dell'utente. Il programma, una volta stabilito quale carattere precede l'altro, stampa tutti i caratteri compresi tra i due inseriti dall'utente.

Esempi

$$a - e \rightarrow abcde \quad e - a \rightarrow abcde$$



POLITECNICO
MILANO 1863

Contatti

Alessandro Montenegro

Email: alessandro.montenegro@polimi.it

Sito: <https://montenegroalessandro.github.io/InfoA2425/index.html>