LABORATORIO DI PROGRAMMAZIONE 1

Docenti: Vincenzo Bonnici (A - L)
Maurizio Boscaini (M - Z)

Lezione 6 – a.a. 2016/2017

Costrutti Decisionali

- Il linguaggio C offre diversi costrutti per prendere decisioni:
  - L'istruzione `if`.
  - L'istruzione `switch`.
  - L'operatore condizionale.

Istruzione If (e If-Else)

- L'istruzione `if` (o `if-else`) è utilizzata per determinare l'esecuzione di una o più istruzioni in base a particolari condizioni:

```
if (espressione) {
  istruzione_1;
  …
  istruzione_n;
} else {
  istruzione_1;
  …
  istruzione_m;
}
```

Esempio con gli operatori di Incremento/Decremento

```
int a = 0;
if (a++ == 0) {
  printf("Il valore di a e' zero\n.);
} else {
  printf("Il valore di a e' positivo\n.");
}

if (++a == 0) {
  printf("Il valore di a e' zero\n.");
} else {
  printf("Il valore di a e' positivo\n.");
}
```

Espressioni Logiche

- La condizione specificata nell'istruzione `if` può essere una qualsiasi espressione logica (valori vero o falso).

- L'espressione logica può essere formata anche da più condizioni unite attraverso un *operatore logico binario*.
  - `&&`: and logico.
  - `||` : or logico.

- Il valore di un'espressione logica può essere negato attraverso *operatore logico unario*:
  - `!`: not logico.

Istruzioni If Annidate

Le istruzioni `if` possono essere annidate.

```
if (n % 2 == 0)
  if (n < 0)
    printf("Il numero %i è pari e negativo.\n", n);
  else
    printf("Il numero %i è pari e positivo.\n", n);
else
  if (n < 0)
    printf("Il numero %i è dispari e negativo.\n", n);
  else
    printf("Il numero %i è dispari e positivo.\n", n);
```

Attenzione: il codice sopra riportato è corretto, ma è consigliabile in ogni caso l'uso delle parentesi per i blocchi anche di una sola istruzione.

Costrutto Else-If

- Quando il numero di casi da distinguere è maggiore di 2, è possibile usare una o più istruzioni `else-if`.

```
if (espressione_1) {
  istruzione_1;
} else if (espressione_2) {
  istruzione_2;
}
…
} else {
  istruzione_3;
}
```

Operatore Condizionale

- L'operatore condizionale è un operatore ternario caratterizzato dalla seguente forma:
  - *condizione* ? *istruzione_1* : *istruzione_2*
  - `condizione` è un'espressione booleana
  - Per prima cosa viene valutata `condizione`.
  - Se `condizione` è vera, allora viene eseguita `istruzione_1`, altrimenti viene eseguita `istruzione_2`.

- È una forma compatta solitamente utilizzata per assegnare un valore ad una variabile in base ad una determinata condizione.
  - `d = (y != 0 ) ? x/y : 0;`

Istruzione Switch

- Una serie di istruzioni `if-else` può essere sostituita dall'uso dell'istruzione switch quando i valori di confronto sono costanti semplici o espressioni costanti.

```
switch (espressione) {
  case valore_1:
  istruzione_1;
  …
  break;
  case valore_2:
  istruzione_2;
  …
  break;
  default :
  istruzione_d;
  …
  break;
}
```

Istruzione Switch

- Le istruzioni all'interno di ciascun `case` *non* devono essere racchiuse tra parentesi graffe!
- Ciascun blocco `case` deve essere concluso con l'istruzione *`break`*, che determina la fine dell'istruzione *`switch`*, altrimenti l'esecuzione prosegue con il prossimo blocco `case`.
- Il blocco *`default`* è facoltativo ed è eseguito quando l'espressione non assume nessuno dei valori specificati nei `case`.
- Non possono esistere due blocchi case con valori uguali, però è possibile associare lo stesso blocco di istruzioni a più valori:

```
case value_1:
case value_2:
  istruzioni;
  break;
```

Istruzione Break all'interno di un ciclo

- L'istruzione break può essere usata anche all'interno di un ciclo.
- L’istruzione break permette di uscire immediatamente dal ciclo che si sta eseguendo.
  - Utilizzata per terminare un ciclo perché si è verificata una certa condizione.
- Le successive istruzioni all’interno del ciclo vengono ignorate e l’esecuzione riprende dalla prima istruzione dopo il ciclo.
- Se l’istruzione break viene eseguita all’interno di un gruppo di cicli annidati, termina soltanto il ciclo più interno che la contiene.

Istruzione Continue all'interno di un ciclo

- L’istruzione continue può essere usata solo all'interno di un ciclo.
- L'istruzione `continue` determina l’uscita dall’iterazione corrente e si prosegue con l’esecuzione di una nuova iterazione.
  - Le istruzioni che restano dal punto di `continue` alla fine del corpo del ciclo sono saltate.
  - Utilizzata per saltare un blocco di istruzione del corpo del ciclo quando si verifica una certa condizione.

Esercizio 1

- Scrivere un programma C che chiede all'utente di inserire un numero intero, calcola se il valore inserito è positivo o negativo, e stampa a video il risultato.

- Il confronto tra *numeri* si esegue con gli operatori:
  - `a < b`: `a` **minore di** `b`,
  - `a <= b`: `a` **minore o uguale a** `b`,
  - `a > b`: `a` **maggiore di** `b`,
  - `a >= b`: `a` **maggiore o uguale a** `b`,
  - `a == b`: `a` **uguale a** `b`.
  - `a != b`: `a` **diverso da** `b`.

Esercizio 2

- Scrivere un programma C che richiede all'utente di inserire un numero intero, calcola se il numero è positivo, negativo oppure nullo, e stampa il risultato ottenuto.

Esercizio 3

- Scrivere un programma C che richiede all'utente di inserire un numero intero e restituisce il valore 0 se il numero è pari oppure 1 se il numero è dispari.

- Usare una variabile ausiliaria per memorizzare il valore da ritornare e l'operatore condizionale per assegnarle il valore.

Esercizio 4

- Scrivere un programma C che richiede all'utente tre numeri interi che rappresentano una data:
  - Giorno, mese, anno
- e verifica se la data inserita è valida.
  - Anno: 4 cifre.
  - Mese: valore compreso tra 1 e 12
  - Giorno: compreso tra 1 e …
    - 29 o 28 se il mese è Febbraio e l'anno è/non è bisestile
      - Un anno è bisestile se:
        - È divisibile per 4 e non è divisibile per 100
        - oppure è divisibile per 400.
    - 31 se il mese è Gennaio, Marzo, Maggio, Luglio, Agosto, Ottobre, Dicembre
    - 30 se il mese è Aprile, Giugno, Settembre, Novembre.

Esercizio 5

- Scrivere un programma C che richiede all'utente due date (spezzare la richiesta di giorno/mese/anno come nell'esempio precedente) e stampa a video la più recente.
  - Per questo esempio assumere che le date inserite siano valide.

Esercizio 6

- Scrivere un programma C che calcola la media dei voti inseriti dall'utente:
  - Ciclo `while` che itera la richiesta di un nuovo voto finché l'utente sceglie di continuare.
    - Lettura dei caratteri s/n.
  - 2 variabili intere aggiornate ad ogni ciclo per tener traccia del numero di voti inseriti e della somma totale dei voti.
  - Il risultato deve essere stampato con 2 cifre decimali.

Esercizio 7

- Scrivere un programma C che richiede all'utente un numero intero n e stampa le sue cifre in lettere.

Suggerimento: usate due cicli.

- Primo ciclo: determina il numero di cifre che compongono n.
  - Se 10^x è la potenza di 10 immediatamente superiore a n, allora $x - 1$ corrisponde al numero di cifre di n.

- Secondo ciclo: divide ripetutamente n per una potenza di 10, partendo da quella immediatamente più piccola di n fino a 1, per recuperare le cifre da sinistra a destra.