Programmazione I

Lezione 9

Recap: controllo del flusso

- Formalismo dei diagrammi a blocchi
- Esempio: esperimento “inverso” con codice per rad.q.
- Costrutto #1: la **sequenza** (esempio radq, esempio in GO)
- Costrutto #2: la **selezione binaria**

if <condizione (vero/falso)>
then <sequenza di istruzioni (se cond. vera) >
else <sequenza di istruzioni (se cond. falsa) >

Esercizi

- *Stabilisci se un numero è negativo*
- *Stabilisci se un numero è pari o dispari*
- *Stampa il minore*
- *Dato il voto di uno studente, stampa l'esito dell'esame (ripeti, promosso, orale facoltativo)*
- Determinare se la prima cifra decimale è pari
- Stampa la minore tra due frazioni
- Stampa il numero di soluzioni (reali) distinte di un'equazione di secondo grado (in forma normale)

Ciclo di vita del software

- Analisi delle esigenze
- Studio di fattibilità

1) Stesura delle specifiche

2) Progettazione

3) Sviluppo

4) Testing e Debugging (goto 3 o anche 2)

5) Rilascio (goto 1)

6) Uso, aggiornamento e manutenzione

Dalla selezione ai cicli

- Idea (sviluppata su esempio di controllo dell'input):
  - Parto da un costrutto di selezione if-then-else ...
  - ... in particolare, la versione con ramo else vuoto

    if <condizione> **{**

    <sequenza>

    }
  - Rendo il costrutto "iterativo": al termine della sequenza lo ripeto, partendo dalla valutazione della condizione

    for <condizione> **{**

    <sequenza>

    }
  - Osservo che ho sempre tre ingredienti:
    (1) inizializzazione (2) condizione (3) aggiornamento

Esercizi

- Stampare la somma delle cifre di un numero di 3 cifre
- Stabilire se la somma delle prime tre cifre di un numero sia < 10
- Leggi numeri (di quantità nota) e stampa la somma
- Leggi numeri e stampa la media
- Leggi numeri e somma fino a trovare uno 0

Il controllo del flusso: costrutto di iterazione

- Costrutto #3: il **ciclo**
- Forma “unaria”

  for <condizione> **{**

  <sequenza>

  }
- Forma “ternaria”

  for <init>; <condizione>; <aggiornamento> **{**

  <sequenza>

  }
- Forma “zeraria”: **for {** <sequenza> **}**