

14BHD INFORMATICA, A.A. 2021/2022

Esercitazione di Laboratorio 4

Obiettivi dell'esercitazione

- Esecuzione ripetuta di istruzioni fino al raggiungimento di un obiettivo
- Elaborazione di dati in ingresso costituiti da molti valori

Contenuti tecnici

- Realizzare cicli con gli enunciati `while` e `for`
 - Comprendere il funzionamento dei cicli annidati
 - Elaborazione di stringhe
-

Da risolvere in laboratorio

- Esercizio 1. Scrivete programmi che leggano una sequenza di numeri interi (la sequenza termina quando viene inserita la *stringa vuota*) che visualizzino quanto segue:
- a. il valore minimo e il valore massimo tra quelli acquisiti
 - b. il numero di valori pari e il numero di valori dispari tra quelli acquisiti;
 - c. le somme parziali di tutti i numeri acquisiti, calcolate e visualizzate dopo ogni nuova acquisizione (se, ad esempio, i valori in ingresso sono 1 7 2 9, il programma visualizzerà 1 8 10 19);
 - d. i valori adiacenti duplicati (se, ad esempio, i valori acquisiti sono 1 3 3 4 5 5 6 6 6 3, il programma visualizzerà 3 5 6). [P4.2]
- Esercizio 2. Scrivete programmi che leggano una riga di dati in ingresso sotto forma di stringa e visualizzino quanto segue:
- a. le sole lettere maiuscole della stringa;
 - b. a partire dalla seconda lettera della stringa, una lettera viene visualizzata e l'altra no, alternativamente;
 - c. la stringa con tutte le vocali sostituita da un carattere di sottolineatura (*underscore*);
 - d. il numero di cifre presenti nella stringa;
 - e. le posizioni di tutte le vocali presenti nella stringa. [P4.3]
- Esercizio 3. Scrivete un programma che legga un numero intero, n , e visualizzi usando asterischi, un quadrato e un rombo pieni il cui lato abbia lunghezza n . Se, ad esempio, l'utente fornisce il numero 4, il programma deve visualizzare:

```

****
****
****
****

    *
  ***
*****
*****
*****
***
*
[P4.22]

```

- Esercizio 4. Scrivete un programma che legga una parola e visualizzi:
- la parola al contrario. Se, ad esempio, l'utente fornisce la stringa "Ciao", il programma deve visualizzare oaiC [P4.9]
 - le lettere maiuscole partendo dal fondo. Se, ad esempio, l'utente fornisce la stringa "CiAo", il programma deve visualizzare AC.

Da risolvere a casa

- Esercizio 5. *Numeri primi*. Scrivete un programma che chieda all'utente un numero intero e, poi, visualizzi tutti i numeri primi minori o uguali a tale numero. Se, ad esempio, l'utente fornisce il numero 20, il programma deve visualizzare:

```

2
3
5
7
11
13
17
19
[P4.17]

```

- Esercizio 6. Scrivete un programma che legga una parola e visualizzi tutte le sue sottostringhe, ordinate per lunghezza crescente. Se, ad esempio, l'utente fornisce la stringa "rum", il programma deve visualizzare

```

r
u
m
ru
um

```

rum

[P4.12]

Esercizio 7. *Il gioco di Nim.* Si tratta di un gioco molto noto, con un certo numero di varianti: quella qui descritta ha una strategia vincente davvero interessante. Due giocatori prelevano alternativamente biglie da un mucchietto. Ad ogni mossa, un giocatore sceglie quante biglie prendere: almeno una e al massimo metà delle biglie disponibili. Poi è il turno dell'altro giocatore. Il giocatore che prende l'ultima biglia perde la partita.

Scrivete un programma che consenta all'utente di giocare contro il computer.

Generate un numero intero compreso tra 10 e 100 e usatelo come dimensione iniziale del mucchietto di biglie. Generate un numero intero, 0 o 1, e utilizzatelo per decidere se sarà l'utente o il computer a giocare per primo. Generate un numero intero, 0 o 1, e usatelo per decidere se il computer giocherà in modo intelligente o stupido:

giocando in modo stupido, ad ogni sua mossa il computer semplicemente preleva dal mucchietto un numero di biglie casuale (ma valido, cioè compreso tra 1 e $n/2$, se nel mucchietto sono rimaste n biglie); in modalità intelligente, invece, preleva un numero di biglie tale che il numero di quelle che rimangono nel mucchio sia una potenza di due diminuita di un'unità, cioè 3, 7, 15, 31 o 63. Quest'ultima è sempre una mossa valida, tranne quando la dimensione del mucchio è proprio uguale a una potenza di due diminuita di un'unità: in tal caso il computer fa una mossa scelta a caso (ovviamente tra quelle valide). Come potrete verificare sperimentalmente, il computer non può essere battuto quando gioca in modalità intelligente e fa la prima mossa, a meno che la dimensione iniziale del mucchio non sia 15, 31 o 63.

Analogamente, un giocatore umano che faccia la prima mossa e conosca la strategia qui descritta è in grado di battere il calcolatore.

[P4.23]