

Fondamenti di Informatica - Esercitazione n. 3

Esercizio 1 Scrivere una funzione C che riceve come parametro un array di **int** e la sua lunghezza e restituisce 1 se la sequenza di interi memorizzata nell'array è *palindroma*, 0 se non lo è. Una sequenza è palindroma se è uguale a se stessa letta al contrario. Ad esempio se l'array contenesse i valori 5 23 76 2 76 23 5, la funzione dovrebbe restituire 1, se invece l'array contenesse i valori 4 6 2 87 9 4, la funzione dovrebbe restituire 0. Scrivere poi un programma che legge una sequenza di interi dall'utente e dice all'utente se la sequenza è palindroma o meno.

Esercizio 2 Scrivere una funzione C che dati due interi n e b (con $n > 0$ e $2 \leq b \leq 9$) calcola la rappresentazione in base b di n . La funzione deve ricevere come parametri i due interi n e b , un array di interi e la dimensione di tale array. L'array dovrà essere modificato dalla funzione per contenere la rappresentazione di n in base b . In particolare l'elemento i -esimo dell'array dovrà memorizzare la cifra i -esima della rappresentazione in base b di n (cioè quella moltiplicata per la potenza b^i). Scrivere poi un programma che fa inserire due valori n e b e, utilizzando la funzione scritta, stampa la rappresentazione in base b di n .

Nota: la lunghezza dell'array passato alla funzione deve essere esattamente pari al numero di cifre necessarie a rappresentare n in base b . Tale valore può essere calcolato mediante l'espressione $\text{floor}(\log(n)/\log(b))+1$

Esercizio 3 Scrivere un programma C che svolga i seguenti punti:

- sceglie un numero casuale intero compreso tra 1 e 1000 e chiede all'utente di indovinare il numero scelto;
- dopo ciascun tentativo dell'utente il programma deve segnalare se la risposta data dall'utente è giusta, è maggiore oppure è minore rispetto al numero scelto;
- il programma deve continuare a chiedere numeri all'utente fino a che questi non abbia dato la risposta corretta, oppure abbia scelto di rinunciare ad indovinare digitando 0;
- se l'utente indovina il numero scelto, il programma deve congratularsi con l'utente, visualizzando anche il numero di tentativi fatti, se l'utente rinuncia, il programma deve indicare qual era il numero da indovinare.

Se, ad esempio, il numero scelto dal calcolatore fosse 31, l'interazione tra il programma e l'utente potrebbe essere la seguente:

```
Ho pensato un numero tra 1 e 1000. Indovina che numero ho pensato:
40
40 è troppo alto, prova ancora.
30
30 è troppo basso, prova ancora.
32
32 è troppo alto, prova ancora.
```

Bravo! Hai indovinato in 4 tentativi.

Nota: per generare il numero da indovinare si usino le funzione `rand()` e `srand()`.

Esercizio 4 *Strike & ball* è un gioco in cui un giocatore deve indovinare un numero di quattro cifre scelto dall'avversario. Le cifre devono essere tutte diverse. Ad ogni mano il giocatore fa un tentativo per indovinare le quattro cifre e riceve come risposta il numero di cifre sono state indovinate ma nel posto sbagliato (*ball*) e il numero di quelle che sono state indovinate nel posto giusto (*strike*). Ad esempio supponiamo che le cifre da indovinare siano 6532. Se il giocatore dice 6321 avrà realizzato uno *strike* e due *ball*. Il numero 6 infatti è stato indicato ed è al posto giusto, i numeri 3 e 2 invece sono stati indicati ma al posto sbagliato. Se il giocatore dicesse 6235 avrebbe realizzato due *strike* (6 e 3) e due *ball* (5 e 2).

Scrivere un programma che:

- sceglie casualmente le quattro cifre da indovinare e chiede all'utente di indovinare il numero scelto;
- dopo ciascun tentativo indica il numero di *strike* e il numero di *ball* realizzati;
- continua a chiedere tentativi all'utente fino a che questi non abbia dato la risposta corretta.
- se l'utente indovina il numero scelto, il programma deve congratularsi con l'utente, visualizzando anche il numero di tentativi fatti.

Per generare le quattro cifre da indovinare scrivere una funzione che riceve come parametro un array di interi **a** di dimensione 4 e lo modifica in maniera che **a[i]** sia la cifra *i*-esima (con $i=0,1,2,3$).

Nota: per garantire che le quattro cifre generate siano distinte, generare ripetutamente ciascuna cifra finché essa non è diversa da tutte quelle generate prima.

Esercizio 5 Modificare il programma dell'esercizio precedente per permettere all'utente di abbandonare la partita. Per farlo è sufficiente che inserisca il valore -1 al posto della prima cifra. Se l'utente rinuncia ad indovinare il programma deve mostrargli quali erano le cifre da indovinare.