

Laboratorio di Programmazione: Linguaggio C

Lezione 9 del 27 novembre 2013

Damiano Macedonio

Giochi d'Azzardo

Note Iniziali

Per generare un *numero casuale* basta includere la libreria `stdlib.h` e utilizzare la funzione `rand()` che genera una sequenza *pseudo-casuale* di interi positivi. Per produrre tale sequenza si deve settare un *seme* iniziale. Se si mantiene il seme di default, ogni volta verrà generata la stessa sequenza. Per modificare il seme iniziale si utilizza la funzione `srand()`. È sufficiente chiamare la `srand()` una sola volta all'interno del programma, prima della prima invocazione di `rand()`, magari subito dopo la dichiarazione iniziale delle variabili. *Non serve invocare `srand()` ogni volta che si invoca `rand()`.* Per avere un seme diverso ad ogni esecuzione del programma includiamo la libreria `time.h` e sfruttiamo la funzione `time()` che restituisce il numero di secondi trascorsi dalle ore 00 : 00 : 00 del 01/01/1970 (il timestamp unix corrente). Vediamo un esempio.

```
#include <stdio.h>      /* printf, NULL */
#include <stdlib.h>     /* srand, rand */
#include <time.h>       /* time */

int main() {
    int n;

    srand (time(NULL));

    n = rand() % 100;
    printf("%i e' un valore casuale tra 0 e 99\n", n);
    n = rand() % 100 + 1;
    printf("%i e' un valore casuale tra 1 e 100\n", n);
    n = rand() % 100 + 23;
    printf("%i e' un valore casuale tra 23 e 123\n", n);

    return 0;
}
```

Esercizio 1

Lancio di una moneta. Scrivere un programma che simula il lancio di una moneta: chiede di scegliere tra testa o croce e dice se si indovina o no.

Esempi

Testa (t) o croce (c)? c
Testa. Hai perso.

Testa (t) o croce (c)? c
Croce. Hai vinto!

Esercizio 2

Morra Cinese. Scrivere un programma che fa giocare a morra cinese contro il computer. L'utente sceglie quante partite fare. Per ogni partita il programma chiede all'utente una scelta tra *sasso*, *carta* o *forbice*. Anche il computer fa la sua scelta (casuale!). Il programma dice se c'è un pareggio o se l'utente ha vinto o perso. Ricordo che

- *sasso* vince contro *forbice*
- *carta* vince contro *sasso*
- *forbice* vince contro *carta*.

Attenzione: per leggere un carattere da tastiera con `scanf()` conviene usare la formattazione `scanf(" %c")` così si saltano gli eventuali spazi bianchi e caratteri di escape come `\n` o `\t`.

Esempio

```
Giochiamo a morra cinese. Quante partite vuoi fare? 5
Partita 1:
sasso (s), carta (c) o forbice (f)? f
Io gioco Forbice! Pareggio.
Partita 2:
sasso (s), carta (c) o forbice (f)? s
Io gioco Carta! Hai perso la partita.
Partita 3:
sasso (s), carta (c) o forbice (f)? c
Io gioco Sasso! Hai vinto la partita!
Partita 4:
sasso (s), carta (c) o forbice (f)? f
Io gioco Forbice! Pareggio.
Partita 5:
sasso (s), carta (c) o forbice (f)? c
Io gioco Forbice! Hai perso la partita.
Risultato finale:
vittorie 1
sconfitte 2
pareggi 2
```

Esercizio 3

Indovina il numero. Scrivere un programma che genera un numero casuale in un intervallo fissato. L'utente deve indovinare il numero. A ogni tentativo, il programma stampa un messaggio che aiuta il giocatore nella ricerca. In pratica il programma deve compiere le seguenti operazioni:

1. Viene chiesto all'utente di inserire un numero $max > 0$. Se l'utente inserisce un numero minore o uguale a 0, l'operazione di lettura va ripetuta.
2. Viene generato casualmente un numero da indovinare, compreso tra 1 e max .
3. Il programma chiede all'utente di inserire un numero. A seconda del numero inserito, viene stampato uno dei messaggi:
 - il numero da indovinare è più piccolo;
 - Il numero da indovinare è più grande;
 - Hai indovinato.

Il programma termina quando l'utente indovina il numero oppure sono stati effettuati un numero prefissato tentativi. In quest'ultimo caso, viene stampato il messaggio del tipo "Non hai indovinato. Il valore era ...".

Esempi (con 5 tentativi massimi)

```
Inserisci il valore massimo: -100
```

```
Inserisci il valore massimo: 100
```

```
Sto pensando ad un numero tra 1 e 100, prova ad indovinarlo: 20
```

```
Hai sbagliato, numero minore di 20. Riprova: 10
```

```
Hai sbagliato, numero maggiore di 10. Riprova: 15
```

```
Hai indovinato! Il valore era 15.
```

```
Sto pensando ad un numero tra 1 e 100, prova ad indovinarlo: 69
```

```
Hai sbagliato, numero maggiore di 69. Riprova: 89
```

```
Hai sbagliato, numero minore di 89. Riprova: 70
```

```
Hai sbagliato, numero maggiore di 70. Riprova: 75
```

```
Hai sbagliato, numero maggiore di 75. Riprova: 77
```

```
Non hai indovinato. Il valore era 85.
```

Esercizio 4

Indovina il numero, restringendo l'intervallo. Migliorare il programma dell'esercizio precedente con le seguenti operazioni

- Viene chiesto all'utente un intervallo entro cui generare il numero da indovinare (min , max). Se i valori inseriti non rispettano il vincolo $0 < min \leq max$, l'operazione di lettura va ripetuta.
- Viene generato casualmente un numero compreso tra min e max .
- Ad ogni tentativo viene stampato un messaggio che indica all'utente l'intervallo entro cui cercare il numero da indovinare in base ai tentativi fatti.

Esempi (con 5 tentativi massimi)

```
Inserisci l'intervallo di ricerca (min max): 87 45
Inserisci l'intervallo di ricerca (min max): -2 87
Inserisci l'intervallo di ricerca (min max): 34 87
Sto pensando a un numero tra 34 e 87, prova ad indovinarlo: 50
Hai sbagliato, numero compreso tra 51 e 87. Riprova: 74
Hai sbagliato, numero compreso tra 51 e 73. Riprova: 60
Hai sbagliato, numero compreso tra 51 e 59. Riprova: 55
Hai sbagliato, numero compreso tra 51 e 54. Riprova: 53
Non hai indovinato. Il valore era 51.
```

```
Inserisci l'intervallo di ricerca (min max): 12 69
Sto pensando a un numero tra 12 e 69, prova ad indovinarlo: 26
Hai sbagliato, numero compreso tra 12 e 25. Riprova: 54
Hai sbagliato, numero compreso tra 12 e 25. Riprova: 3
Hai sbagliato, numero compreso tra 12 e 25. Riprova: 18
Hai sbagliato, numero compreso tra 19 e 25. Riprova: 23
Hai indovinato! Il valore era 23.
```

Nota. Gli estremi dell'intervallo vanno aggiornati solamente se l'intervallo di ricerca si restringe. Ad esempio, se il programma comunica che il numero è compreso tra 12 e 25 e l'utente inserisce 54, il messaggio da stampare al passo successivo è ancora

Hai sbagliato, numero compreso tra 12 e 25. Riprova:

(e non Hai sbagliato, numero compreso tra 12 e 54. Riprova: che ha meno informazioni rispetto al messaggio precedente).

Esercizio 5

Pensa a un numero. Invertiamo le parti rispetto ai due esercizi precedenti: l'utente pensa ad un numero e il computer lo indovina. In pratica il programma compie le seguenti operazioni.

1. L'utente fissa un intervallo entro cui generare il valore da indovinare (min , max). Se i valori inseriti non rispettano il vincolo $0 < min \leq max$, l'operazione di lettura va ripetuta.
2. L'utente pensa ad un numero compreso tra min e max .
3. Il programma cerca di indovinare la scelta dell'utente e propone un numero. A seconda della situazione l'utente deve dire al programma se
 - il numero da indovinare è più piccolo;
 - Il numero da indovinare è più grande;
 - Hai indovinato.
4. Non poniamo limite al numero di tentativi (il programma deve essere furbo!)

Esempi (due alternative di soluzione)

```
Inserisci l'intervallo di ricerca (min max): 23 89
Pensa ad un numero compreso tra 23 e 89
Hai pensato a 56? (s/n) n
il numero viene prima (<) o dopo (>) di 56? >
Hai pensato a 73? (s/n) n
il numero viene prima (<) o dopo (>) di 73? >
Hai pensato a 81? (s/n) n
il numero viene prima (<) o dopo (>) di 81? <
Hai pensato a 77? (s/n) n
il numero viene prima (<) o dopo (>) di 77? <
Hai pensato a 75? (s/n) s
Ho indovinato!!
```

```
Inserisci l'intervallo di ricerca (min max): 45 124
Pensa ad un numero compreso tra 45 e 124
Il numero e' minore (<) uguale (=) o maggiore (>) di 84? >
Il numero e' minore (<) uguale (=) o maggiore (>) di 104? <
Il numero e' minore (<) uguale (=) o maggiore (>) di 94? >
Il numero e' minore (<) uguale (=) o maggiore (>) di 99? <
Il numero e' minore (<) uguale (=) o maggiore (>) di 96? >
Il numero e' minore (<) uguale (=) o maggiore (>) di 97? =
Hai pensato a 97!
```