

Laboratorio di Fondamenti di Informatica

LAB10_2015.11.17

Anno accademico 2015/2016

Sorpresa: torniamo sui nostri passi

La verifica si avvicina, è il momento di consolidare le fondamenta, non di alzare il palazzo di un piano.

Sono stati assegnati tanti esercizi, quasi sicuramente a ognuno ne resta qualcuno non affrontato, da finire, da perfezionare.

Magari potete confrontare le vostre soluzioni con quelle dei compagni, criticarvele a vicenda, domandarvi non solo in cosa possono essere incomplete o scorrette, ma anche quali sono «migliori», e perché.

O, se proprio non riuscite a stare senza un esercizio «nuovo», potete costruire un programma per giocare a MasterMind (magari facendo generare le combinazioni a caso al calcolatore per provare a indovinarle).

Ne riparleremo a fine corso (discutendo di come si fa a farle *indovinare* al calcolatore). Intanto si parte dall'inizio... costruendo la funzione di verifica delle combinazioni, come spiegata nella slide successiva.

MasterMind (semplificato)

Il *MasterMind Semplificato* è un gioco di abilità e fortuna in cui bisogna indovinare una combinazione segreta (detta il **codice**) formata da N colori (scelti tra C diversi colori disponibili, codificati come C caratteri da 'A' a 'A' + C-1) disposti in un ordine specifico e senza ripetizioni.

Il giocatore effettua dei **tentativi** (cioè combinazioni proposte) per indovinare il codice, e ottiene in **risposta** ad ogni tentativo il numero di colori da lui collocati in posizione corretta e il numero di colori presenti nel codice ma da lui collocati in posizione errata.

Si considerino le definizioni seguenti (N e C sono costanti simboliche, $N \leq C$) e gli esempi che mostrano le risposte ai tentativi (con $N=4$ per brevità):

```
typedef char Combinazione[N];  
typedef struct { int esatti; int spostati; } Risposta;  
char colori[C] = { 'A', 'B', 'C', 'D', 'E', ... }
```

Esempi: ipotizzando che il codice sia: {A,B,C,D}

Tentativo → risposta:

{G,L,M,V} → {0,0} {F,C,I,B} → {0,2} {A,R,B,C} → {1,2} {A,C,B,D} → {2,2} {A,B,C,D} → {4,0}

Si codifichi la funzione

Risposta verifica(Combinazione codice, Combinazione tentativo)

che riceve come parametri il codice e il tentativo del giocatore e restituisce la risposta opportuna. Eventuali funzioni ausiliarie (non standard) devono essere chiaramente definite e codificate.

ESTENSIONE (probabilmente non banale)

Si consideri il caso (che è quello del gioco più comunemente giocato) in cui sia nel codice sia nel tentativo **si possono usare numeri ripetuti**.