

Sapienza Università di Roma – Consorzio NETTUNO
Ingegneria Informatica, Elettronica, Telecomunicazioni
Fondamenti di Informatica 1 – 25-7-09

Parte pratica

Svolgete **SOLO DUE** esercizi a scelta tra questi 3

Nota: se lo ritenete utile potete suddividere le funzioni richieste in più sottofunzioni

Esercizio 1 (stringhe – 8 punti)

Si deve cercare una sottostringa in un testo. Si supponga però che la stringa da cercare nel testo possa contenere dei caratteri errati e che i caratteri possono essere errati SOLO sul loro bit meno significativo. Si implementi la funzione **trova** che riceve come argomenti il **testo** e la **stringa** da cercare e tutti gli altri argomenti che ritenete necessari, e che cerca la stringa nel testo ignorando il valore del bit meno significativo di ciascun carattere (quindi due caratteri sono uguali se al più differiscono per il loro bit meno significativo)

La funzione deve tornare, tramite due variabili passate per riferimento:

- la posizione di inizio della prima corrispondenza trovata della sottostringa nel testo (oppure -1 se non viene trovata)
- il numero di caratteri “sbagliati” (quelli che corrispondono ma hanno il bit meno significativo diverso)

Suggerimento: implementate in una funzione separata il confronto tra due caratteri che ignora il loro bit meno significativo.

Esempio: il carattere 'C' ed il carattere 'B', che hanno codici ASCII **67** e **66** in decimale, ovvero 01000011 e 01000010 in binario differiscono solo per il bit meno significativo (sottolineato), quindi vanno considerati uguali durante la ricerca ed inoltre debbono essere contati come caratteri “sbagliati”.

Esercizio 2 (vettori e matrici – 10 punti)

Siano dati 2 vettori di interi di 100 elementi, che si chiamano **MosseUno** e **MosseDue**. Si supponga di voler giocare a battaglia navale e che ciascuno dei due vettori contenga le coppie di coordinate per le mosse del giocatore corrispondente. **Esempio:** gli elementi **MosseUno[0]** e **MosseUno[1]** indicano la colonna e la riga della prima mossa del giocatore Uno. La seconda mossa sta alle posizioni 2 e 3 e così via.

Scrivere una funzione **battaglia_navale** che riceve come argomenti due matrici di interi di dimensioni 10 per 10, contenenti valori 0 o 1, che rappresentano la posizione delle navi (0 acqua, 1 nave) e i due vettori delle mosse che indicano i punti in cui si vuole sparare. La somma degli 1 in ogni matrice è pari a 25 (1 nave da 5, 1 da 4, 2 da 3, 3 da 2 e 4 da 1).

La funzione esamina nell'ordine le mosse (in maniera alternata tra i due giocatori) e restituisce **1** se è il **primo** giocatore quello che distrugge per primo tutte le navi dell'avversario, **-1** se è il **secondo** giocatore che distrugge per primo tutte le navi dell'avversario, altrimenti restituisce **0** se arriva al termine di dei due vettori senza che alcun giocatore abbia battuto l'avversario.

(segue)

Esercizio 3 (matrici – 6 punti)

Scrivete una funzione che riceve un argomento intero $N < 100$ ed una matrice M di dimensioni adeguate che riempie con il triangolo di Tartaglia di ordine N , una riga per volta, la matrice M .

Esempio: il triangolo di Tartaglia di ordine 5 è:

1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

La prima riga contiene solo due 1

Ciascuna riga successiva si ottiene dalla precedente sommando le coppie di elementi consecutivi e aggiungendo un 1 all'inizio e alla fine; quindi la sesta riga è

1 6 15 20 15 6 1

(in cui sottolineo i valori ottenuti sommando le coppie di elementi della riga 5).

Parte teorica

Svolgete **SOLO DUE** esercizi a scelta tra questi 3

Esercizio A (rappresentazione dei numeri – 5 punti)

Siano dati i numeri decimali $N = 57$, $M = 125$.

- 1) Stabilire il numero minimo B di cifre binarie occorrenti a rappresentare N ed M come numeri binari puri;
- 2) scrivere la rappresentazione di N ed M come numeri binari puri con B cifre ed eseguire la somma $N+M$;
- 3) Stabilire il numero minimo di cifre necessarie per rappresentare N e $-M$ nella codifica in complemento a due e mostrare tali rappresentazioni, dettagliando il procedimento usato per ottenerle;
- 4) infine eseguire l'operazione $N-M$, e scrivere il risultato usando 16 bit (sempre nella rappresentazione in complemento a 2).

Esercizio B (operatori logici in C – 5 punti)

Cosa sono gli operatori booleani? Dove si usano?

Descrivere quelli conosciuti, specificandone la definizione mediante congrui esempi.

Discutere poi l'implementazione di alcuni di tali operatori in C.

Usare gli esempi mostrati precedentemente per dimostrare il funzionamento degli operatori logici in C.

In particolare discutere le importanti modalità con le quali le espressioni logiche C vengono valutate.

Esercizio C (ricorsione – 8 punti)

Si implementi ricorsivamente la funzione **palindromaP** che riceve come argomenti una stringa contenente una frase, un intero contenente il suo numero di caratteri e tutti gli altri argomenti che ritenete necessari, e che verifica se la frase è palindroma (ovvero se è la stessa letta dall'inizio o dalla fine) **ignorando eventuali spazi disposti tra le parole.**

Esempio: la frase “**amo Roma**” è palindroma (infatti, tranne per lo spazio è la stessa letta da sinistra o da destra).

Nota: ad ogni chiamata ricorsiva potete sicuramente confrontare due caratteri oppure ignorare uno spazio.