


Fondamenti di Informatica - Ingegneria Informatica – Prof. Maristella Matera

Prima prova in itinere – 24 novembre 2016

Il tempo massimo a disposizione per svolgere la prova è di **2h e 15 minuti**. È vietato utilizzare calcolatrici, telefoni, PC. **Il voto minimo per superare la prova è 18.**

Esercizio 1. Codifica Binaria (4 punti)

1. Si dica qual è l'intervallo di valori interi rappresentabile con la codifica in complemento a due a 8 bit.
2. Con riferimento a tale codifica indicare, giustificando brevemente le risposte, quali delle seguenti operazioni possono essere eseguite correttamente:
 - (a) $-126 - 127$
 - (b) $+126 - 125$
 - (c) $-18 + 115$
 - (d) $+104 + 88$
3. Mostrare in dettaglio come avviene il calcolo delle operazioni (i) e (ii), illustrando la codifica dei numeri in complemento a due ed evidenziando nelle operazioni eventuali bit di riporto e di overflow ottenuti.

Esercizio 2 (8 punti). Il Politecnico vuole calcolare delle statistiche sulle modalità con cui gli studenti sostengono gli esami. In particolare vuole calcolare quanti esami sono ripetuti dagli studenti sia nella sessione invernale, sia in quella estiva.

Si supponga che, dato uno studente, le sue iscrizioni agli appelli invernali e a quelli estivi siano memorizzate in due array distinti, `appelliInvernali` e `appelliEstivi`. Tali array sono entrambi del tipo `sessione` di seguito definito:

```
#define N ...
typedef struct {
    char codiceEsame [5];
    char nomeEsame [51];
    char dataIscrizione[11];
}iscrizione;
typedef iscrizione sessione[N];
sessione appelliInvernali, appelliEstivi;
```

Si scriva una funzione in C che, dati i due array `appelliInvernali` e `appelliEstivi` relativi a un certo studente, **restituisca il numero degli esami che sono contenuti in entrambi gli array, facendo attenzione a non contare le ripetizioni**. Per esempio, se:

```
appelliInvernali =
{{ "085A", "Analisi Matematica", "31/01/2016"},
{"084B", "Fondamenti di Informatica", "04/02/2016"},
{"085A", "Analisi Matematica", "15/02/2016"},
{"072N", "Geometria e Algebra Lineare", "16/02/2016"},
{"072N", "Geometria e Algebra Lineare", "24/02/2016"}}
```

```

appelliEstivi =
{{"084B", "Fondamenti di Informatica", 30/06/2016},
{"012B", "Fisica", "04/07/2016"},
{"065Z", "Elettrotecnica", "05/07/2016" },
{"065Z", "Elettrotecnica", "15/07/2016"},
{"085A", "Analisi Matematica", "18/07/2016"}}

```

allora la funzione restituisce il valore 2, in quanto gli esami comuni ai due array, a meno di ripetizioni in ogni singolo array, sono "Fondamenti di Informatica" e "Analisi Matematica".

N.B.: Gli esami in un array possono essere ripetuti nell'eventualità in cui lo studente si sia iscritto per uno stesso esame a più appelli nella stessa sessione.

Esercizio 3 (10 punti). Sia data una matrice quadrata A di ordine N contenente valori interi 0 e 1. Sia data una seconda matrice quadrata, B, di ordine 3.

Si scriva un programma in C che, letti in input gli elementi delle matrici A e B, verifichi se in A esiste una sottomatrice identica a B (è sufficiente individuarne una – la prima). In caso affermativo, il programma deve stampare le coordinate di riga e colonna in A nelle quali si trova l'elemento B[0][0].

Esempio: Consideriamo le due seguenti matrici:

A =	1 1 0 1 1 1 0 . . . 1 0 0 0 0 0 1 . . . 1 1 0 0 1 0 1 . . . 1 1 0 1 1 1 0 . . .	B =	1 0 1 0 0 0 1 0 0
-----	--	-----	-------------------------

In questo caso il programma stamperà un messaggio per indicare che la matrice B può essere collocata nella matrice A a partire dalla **riga di indice 0** e la **colonna di indice 1**.

Esercizio 4 (8 punti). Sia data una stringa memorizzata in un array. Si supponga che la stringa sia inizializzata in modo esplicito e non debba essere letta in input (per l'inizializzazione si può usare l'esempio sotto riportato). Si scriva un programma in C che, operando solo sull'array a disposizione (senza utilizzare strutture dati aggiuntive), sostituisca nella stringa le eventuali sotto-sequenze di elementi delimitati da parentesi tonde (parentesi aperta all'inizio della sequenza, parentesi chiusa alla fine della sequenza) con un solo asterisco, cioè "*". Il programma quindi stampa la nuova sequenza così ottenuta.

Per esempio, data la sequenza iniziale {a, b, (, a, c, g), b, e, a, (, x, x), f, \0}, alla fine l'array conterrà la seguente sequenza {a, b, (, *,), b, e, a, (, *,), f, \0}.

NB.:

1. La sequenza di caratteri inizialmente memorizzata nell'array non potrà contenere coppie di parentesi che si "intersecano"; per ogni parentesi aperta, esiste una parentesi chiusa; tra due parentesi tonde è sempre incluso almeno un carattere.
2. Non è necessario leggere in input la sequenza; l'array può essere inizializzato in modo esplicito, usando per esempio la sequenza sopra riportata.