



Esame del 04/09/2013 - Seconda Prova

Esercizio 1. Dato il numero intero -35 in base 10, calcolare le rappresentazioni in i) modulo e segno, ii) complemento a 1, e iii) complemento a 2 avendo 8 bit a disposizione.

Esercizio 2. Si consideri il seguente rompicapo enigmistico: all'interno di una frase F potrebbe essere celata una parola segreta; per individuarla è necessario applicare le seguenti regole: 1) cancellare da F tutte e sole le occorrenze delle lettere che compaiono più di una volta in F, 2) se nessuna lettera è rimasta dopo la cancellazione, allora nella frase non è nascosta alcuna parola segreta; altrimenti 3) la parola nascosta si ottiene considerando tutte le lettere rimaste (tralasciando ovviamente gli spazi), prese nell'ordine.

Si scriva un programma C++ **OPPORTUNAMENTE MODULARIZZATO IN FUNZIONI** che letta da input un frase F verifichi se in F è nascosta o meno una parola, secondo il meccanismo descritto sopra; in caso affermativo la stampi su standard output, altrimenti stampi il messaggio "Nessuna Parola Nascosta". **NOTA:** per semplicità, si può supporre che la frase contenga solo lettere maiuscole e spazi. Inoltre, non è necessario verificare che la parola segreta sia di senso compiuto. **ESEMPIO:** Se la frase introdotta da input fosse IO NON STIMO ME STESSO, il programma dovrebbe stampare su standard output "Nessuna Parola Nascosta"; se invece la frase in input fosse UNGERE CHIARE CERE, il programma dovrebbe stampare "La parola segreta e' UNGHIA".

Esercizio 3. Si scriva in C++ una funzione RICORSIVA che, ricevuti come parametri (almeno) un array di numeri interi e la sua dimensione, verifichi se sono presenti almeno tre elementi in posizioni distinte di cui uno è pari al prodotto degli altri due, ed in caso positivo, restituisca le loro posizioni. Se sono presenti più triple che soddisfano la proprietà, è necessario restituire le posizioni della prima. **ESEMPIO:** dato l'array qui di fianco, di dimensione 10, la funzione dovrà restituire le posizioni 1, 2 e 4, infatti gli elementi ad esse corrispondenti (8, 4 e 2 rispettivamente), sono tali per cui $4 \cdot 2 = 8$.

1	8	4	3	2	6	5	9	10	11
---	---	---	---	---	---	---	---	----	----

Esercizio 4. La seguente funzione dovrebbe calcolare la somma degli elementi di un array, partendo da quello in ultima posizione e procedendo a ritroso; tuttavia sono presenti alcuni errori sintattici o logici. Individuarli, motivando la risposta e proponendo una possibile correzione che sia il più possibile aderente all'attuale implementazione della funzione.

```
int somma(int A[], int dim)
{
  int &sum = 0; * p= A+dim;
  while (p > A)
  {
    sum += p;
    p-=1;
  }
  return sum;
}
```



Esame del 04/09/2013 - Seconda Prova

Esercizio 5.

Data una matrice \mathbf{M} di dimensione $\mathbf{m} \times \mathbf{n}$, una *banda orizzontale* di dimensione \mathbf{x} , con $1 \leq \mathbf{x} \leq \mathbf{m}$, è un insieme di \mathbf{x} righe consecutive tutte uguali fra loro; si noti che una riga diversa sia dalla precedente che dalla successiva, fa parte di una banda di dimensione 1. Una banda orizzontale che include la prima riga della matrice viene detta *banda pivot*. Una matrice si dice a bande orizzontali, se tutte le bande individuabili sono di dimensione pari a quella della banda pivot. Si scriva in C++ una funzione che ricevuto come parametro (almeno) una matrice \mathbf{M} di dimensione $\mathbf{m} \times \mathbf{n}$, verifichi se la matrice è a bande orizzontali, e restituisca la dimensione delle bande. **NOTA:** Per la verifica è necessario dapprima calcolare la dimensione della banda pivot. Inoltre, una matrice in cui tutte le righe sono diverse tra loro, è una matrice a bande orizzontali di dimensione 1. **ESEMPIO:** la matrice 7×6 riportata in basso a sinistra, non è una matrice a bande orizzontali: infatti, è possibile individuare, oltre alla banda pivot di dimensione 2, altre 2 bande orizzontali di dimensione 1 e 3 rispettivamente (entrambe di dimensione diversa da quella della banda pivot). Al contrario, la matrice 7×6 riportata in basso a destra è a bande orizzontali di dimensione 2.

1	6	3	4	8	3	9
1	6	3	4	8	3	9
2	7	3	5	5	4	2
5	2	7	8	4	5	1
5	2	7	8	4	5	1
5	2	7	8	4	5	1

Banda di dimensione 2
(PIVOT)

Banda di dimensione 1

Banda di dimensione 3

1	6	3	4	8	3	9
1	6	3	4	8	3	9
2	7	3	5	5	4	2
2	7	3	5	5	4	2
5	2	7	8	4	5	1
5	2	7	8	4	5	1

Banda di dimensione 2
(PIVOT)

Banda di dimensione 2

Banda di dimensione 2