



<b>Cognome</b>	<b>Nome</b>	<b>Matricola</b>	<b>Voto: ... /30</b>
----------------	-------------	------------------	----------------------

Quesito:	1	2	3	4	5	6	Tot.
Max:	5	5	5	5	5	5	30
Punti:							

INIZIARE LA SOLUZIONE DI OGNI  
ESERCIZIO SU UNA PAGINA NUOVA

RESTITUIRE COMPILATO ANCHE  
NEL CASO IN CUI CI SI RITIRA

**Istruzioni:**

- non è possibile consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del rosso.
- tempo a disposizione: 2h 00m

**Stile del codice C:**

- non è necessario inserire direttive `#include`;
- i commenti non sono necessari, ma potrebbero essere utili nel caso di errore;
- è consentito l'utilizzo di funzioni di libreria.

**Quesito 1 (5 punti)**

Punteggio ottenuto: .../5

Dati i due numeri  $A = +57_{10MS}$  e  $B = +3A_{16MS}$  effettuare la conversione in base 2, notazione complemento a 2 (2C2), sul numero minimo di bit necessari a rappresentare entrambi gli operandi. Si effettuino quindi le operazioni  $A+B$  e  $A-B$  indicando esplicitamente se si verifica overflow o meno, e motivando la risposta. Mostrare i passaggi fatti.

**Quesito 2 (5 punti)**

Punteggio ottenuto: .../5

Scrivere un programma per il calcolo di statistiche sui caratteri presenti in un file di testo il cui nome è specificato come primo parametro sulla riga di comando. Il programma considera tutti i caratteri ad eccezione degli spazi e fornisce in uscita le seguenti statistiche:

- il numero di righe presenti nel file (ogni riga è al più di 100 caratteri),
- il numero totale di caratteri (esclusi quelli di spaziatura),
- il numero massimo e medio di caratteri di riga,
- la riga più lunga presente nel file.

Dato il file con il seguente testo:

```
Prova di contenuto di file
per avere un esempio.
Non e' importante quello che c'e' scritto
presente.
```

il programma visualizza:

```
4
84
35 21
Non e' importante quello che c'e' scritto
```

**Quesito 3 (5 punti)**

Punteggio ottenuto: .../5

Scrivere un sottoprogramma ricorsivo che ricevuto in ingresso un numero intero restituisce la cifra più alta della rappresentazione decimale. Per esempio, se il valore ricevuto in ingresso è 1232 il sottoprogramma restituisce 3. **La versione iterativa del sottoprogramma vale al più 3 punti.**

**Quesito 4 (5 punti)**

Punteggio ottenuto: .../5

Scrivere un sottoprogramma che riceve in ingresso due liste (il tipo degli elementi della lista è `list_t`) e restituisce una nuova lista creata alternando gli elementi delle due liste, e poi mettendo in coda tutti quelli della lista più lunga. Ad esempio, se le liste in ingresso sono fatte come segue,

```
1 3 5 7 9 11
2 4 6 8
```

la lista creata sarà fatta come segue

```
1 2 3 4 5 6 7 8 9 11
```

**Quesito 5 (5 punti)**

Punteggio ottenuto: .../5

Scrivere un programma che acquisisce i dati per popolare due array monodimensionali di 10 interi ciascuno. Il programma verifica quindi se, pur di fare uno scorrimento circolare del secondo array, i due array siano identici. Nello scorrimento circolare, la cifra che si trova in ultima posizione diventa quella in prima posizione dopo lo scorrimento. In caso affermativo il programma visualizza di quante posizioni deve essere fatto scorrere il secondo array verso destra. In caso negativo, visualizza -1.

**Esempio 1****Ingresso:** 0248217829 1748272616**Uscita:** -1**Esempio 2****Ingresso:** 5498301934 9345498301**Uscita:** 7**Quesito 6 (5 punti)**

Punteggio ottenuto: .../5

Scrivere un programma che trova e visualizza tutti i vocaboli di un dizionario (file di testo ASCII `dizionario.txt`) in cui tutte le vocali presenti compaiono in ordine lessicografico crescente. Non è necessario che compaiano tutte le vocali e la stessa vocale può comparire più volte. Nel file, ogni vocabolo compare su una riga nuova, ed ogni vocabolo è al più lungo 35 caratteri. I caratteri sono tutti minuscoli e non accentati. Per esempio, i vocaboli `alta`, `arte` vengono visualizzati, il vocabolo `idea no`.