



Cognome	Nome	Matricola	Voto: ... /30
---------	------	-----------	---------------

Quesito:	1	2	3	4	5	6	7	Tot.
Max:	4	4	4	4	4	5	5	30
Punti:								

Istruzioni:

- un punteggio inferiore a 18/30 preclude la possibilità di sostenere la seconda prova in itinere;
- 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;
- **non è consentito l'utilizzo di funzioni di libreria.**

Quesito 1 (4 punti)

Punteggio ottenuto: ... /4

Dati i due numeri $A = -47_{10MS}$ e $B = 1010001_{2C2}$ 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 (4 punti)

Punteggio ottenuto: ... /4

Dato il numero $A = +1.765625_{10}$ convertirlo in base 2, notazione IEEE 754, singola precisione (non nella forma $(1+M) \times 2^e$, bensì nella forma estesa con tutti i bit). Codificare il valore convertito in base 16. Mostrare i passaggi.

Quesito 3 (4 punti)

Punteggio ottenuto: ... /4

Con riferimento all'architettura di Von Neumann, si elenchino i registri della CPU, indicando chiaramente per ciascuno di essi il ruolo che esso svolge, ed il dato in esso memorizzato.

Quesito 4 (4 punti)

Punteggio ottenuto: ... /4

Dato il seguente programma, si identifichino e si correggano gli errori in esso contenuti, sapendo che il suo scopo è quello di acquisire una stringa di al più 30 caratteri, invertire l'ordine dei suoi caratteri e visualizzarla;

```
#include <stdio.h>
#define N 30

int main(int argc, char * argv[])
{
    char parola[N], tmp;
    int i;

    printf("inserisci un vocabolo di al piu' 30 caratteri: ");
    scanf("%s", &parola);

    for(i = 0; i <= N; i++) {
        tmp = parola[i];
        parola[i] = parola[N-i];
        parola[N-i] = tmp;
    }
    printf("%d", parola);
    return 0;
}
```

Quesito 5 (4 punti)

Punteggio ottenuto: ... /4

Scrivere un programma che acquisisce i dati di una matrice quadrata di dimensione 10, quindi i dati di una matrice di dimensione 4. Il programma verifica se la seconda matrice è contenuta nella prima ed in tal caso visualizza 1, 0 altrimenti.

Esempio (con dimensioni rispettivamente 5 e 2):

Ingresso:

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

Uscita:

```
4 5 1
8 1
```

Quesito 6 (5 punti)

Punteggio ottenuto: ... /5

Scrivere un programma che ricevuto in ingresso un numero intero, conta il numero di occorrenze di ogni cifra, visualizzando l'informazione <cifra, occorrenze> in ordine crescente rispetto alle occorrenze (non importa l'ordine tra le cifre a parità di occorrenze), e solo se compare almeno una volta.

```
22111113    9341173
3 1          7 1
2 2          4 1
1 4          9 1
            1 2
            3 2
```

Quesito 7 (5 punti)

Punteggio ottenuto: ... /5

Supponendo che le leggi che regolano l'andamento della popolazione di fagiani e di volpi siano le seguenti:

$$fag_i = fag_{i-1} + 0.04 * fag_{i-1} - 0.0005 * fag_{i-1} * vol_{i-1} \quad (1)$$

$$vol_i = vol_{i-1} - 0.03 * vol_{i-1} + 0.00025 * fag_{i-1} * vol_{i-1} \quad (2)$$

dove fag_i e vol_i rappresentano rispettivamente i fagiani e le volpi vive al periodo i -esimo, si scriva un programma che chiede all'utente di inserire un valore intero n e la popolazione iniziale fag_0 e vol_0 e calcola il numero di fagiani e di volpi al periodo n (ossia vol_n e vol_n).

Esempio:

Ingresso:

```
70
300
40
```

Uscita:

```
fagiani: 28 volpi: 125
```