



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

Quesito:	1	2	3	4	5	6	7	Tot.
Max:	4	3	4	5	6	5	3	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, né comunicare;
- non è possibile utilizzare la calcolatrice o qualsiasi dispositivo elettronico;
- si può rispondere ai quesiti nell'ordine preferito;
- si può scrivere con qualsiasi colore, anche a matita, ad eccezione del **rosso**.
- tempo a disposizione: 2hr

**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  $A = -18_{10}$  e  $B = -25_{10}$  effettuare la conversione in base 2, notazione complemento a 2, sul numero minimo di bit necessari a rappresentare gli operandi. Si effettuino poi, in tale rappresentazione, 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 (3 punti)**

Punteggio ottenuto: ... /3

Dati il numero rappresentato in base due, notazione virgola mobile  $A = 010001010111000000001100000000$  convertirlo in base 10; va bene anche la forma  $(1 + M) \times 2^e$ , dove  $M$  è una somma di potenze (negative) di 2. Mostrare i passaggi.

**Quesito 3 (4 punti)**

Punteggio ottenuto: ... /4

Scrivere un programma che acceda al file di testo `fi.log` costituito da un numero di righe a priori non noto, ciascuna di al più 75 caratteri, così organizzate:

- exp** intero che rappresenta l'indicativo numerico dell'esperimento (non necessariamente in ordine);
- classificazione** due caratteri, ciascuno dei quali è '0' o '1', che indica il risultato dell'esperimento, e
- traccia** che è una sequenza di 65 caratteri, ciascuno dei quali è '0' o '1', che rappresenta la sequenza d'uscita dell'esperimento.

La classificazione ha il significato di seguito descritto. '0' '0': guasto non osservato non rilevato (NOND), '0' '1' guasto non osservato e rilevato (NOD), '1' '0' guasto osservato e non rilevato (OND), '1' '1' guasto osservato e rilevato (OD). Il programma analizza il file e visualizza: il numero degli esperimenti effettuati, il numero di esperimenti OD e OND e per essi il numero medio di '1' nella traccia.

**Esempio traccia**

```
125 01 01010001001001111100001011100000011010101010101010101010101111111
321 11 01010101001011100010100101110011100010010011110000101101011010111
1121 11 00010101010010101010101110011010111011111100100101110010001100111
1122 10 00000000000010111000110101101101011001110100001100111100000000000
```

**Quesito 4 (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 5 (6 punti)**

Punteggio ottenuto: .../6

Definire i seguenti tipi di dati:

- `contatto_t`: costituito da due campi: `alias` e `numTel`, stringhe rispettivamente di al più 20 e 15 caratteri;
- `data_t`: `giorno`, `mese` ed `anno`, `ora` e `minuto` in cui si effettua/riceve una chiamata (tutti interi);
- `durata_t`: per memorizzare il numero di ore, minuti e secondi (interi) della durata di una chiamata;
- `chiamataIN_t`: per memorizzare le informazioni relative ad una chiamata ricevuta;
- `chiamataOUT_t`: per memorizzare le informazioni relative ad una chiamata effettuata;
- `telefonino_t`: costituito dai seguenti campi (ognuno del tipo più opportuno tra quanti introdotti): `numTel`, stringa di al più 15 caratteri, `rubrica` per memorizzare le informazioni di 200 contatti, `registroIN` di 35 elementi per le chiamate ricevute e `registroOUT` di 35 elementi per le chiamate in uscita.

Si scriva un programma che dichiara una variabile `telefono` per la gestione delle informazioni di un telefonino e si chiede all'utente di inserirne il contenuto. Una volta **terminata l'acquisizione** il programma chiede all'utente mese e anno (due interi) e visualizza il numero di chiamate effettuate in quel periodo.

**Quesito 6 (5 punti)**

Punteggio ottenuto: .../5

In relazione all'architettura di un sistema di calcolo secondo il modello di Von Neumann ed in particolare alla CPU, quali elementi vengono coinvolti nell'esecuzione di un'istruzione aritmetica, tipo  $c = a - b$ , a partire dall'inizio del ciclo `fetch-decode-execute`? E quale ruolo svolge ciascuno di essi?

**Quesito 7 (3 punti)**

Punteggio ottenuto: .../3

Dire sinteticamente qual è la funzionalità svolta dal programma di seguito riportato. Le risposte che spiegano il codice istruzione per istruzione non saranno prese in considerazione).

Si scriva cosa visualizza il programma quando riceve in ingresso 1111011001.

```
#define L 10
```

```
int main(int argc, char *argv[])
{
    char    val[L+1], comp[L+1];
    int     i;
    int     notOK, first1;

    do {
        scanf("%s", val);
        for(i = 0, notOK=0; val[i] != '\0' && !notOK; i++)
            if(val[i] != '0' && val[i] != '1')
                notOK = 1;
    } while(notOK);

    comp[i] = val[i];
    for(i--, first1 = 0; i >= 0; i--)
        if(first1)
            if(val[i] == '0')
                comp[i] = '1';
            else
                comp[i] = '0';
        else {
            comp[i] = val[i];
            if(val[i] == '1')
                first1 = 1;
        }

    printf("\n%s > %s\n", val, comp);
}
```