



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

Quesito:	1	2	3	4	5	6	Tot.
Max:	4	5	5	8	4	4	30
Punti:							

**Quesito 1 (4 punti)**

Punteggio ottenuto: ... /4

Dati i due numeri  $A = -33_{16}$  e  $B = -28_{10}$  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 (5 punti)**

Punteggio ottenuto: ... /5

Scrivere un sottoprogramma che ricevuta in ingresso una stringa, restituisce un numero intero  $k$  che caratterizza la stringa. Una stringa è una  $k$ -stringa se in essa ci sono esattamente  $k$  caratteri adiacenti tali per cui  $s[i+1] < s[i]$ . Il sottoprogramma deve individuare il  $k$  massimo per la stringa ricevuta in ingresso. La stringa contiene senz'altro caratteri minuscoli ('a', 'b', ..., 'z').

**Esempio 1**

**Ingresso:** intercarpedine  
**Uscita:** 4

**Esempio 2**

**Ingresso:** arcobaleno  
**Uscita:** 3

**Quesito 3 (5 punti)**

Punteggio ottenuto: ... /5

Scrivere un programma che riceve in ingresso il nome di un file (al più 30 caratteri, inclusi percorso, estensione, ecc.) in cui sono contenuti 10 valori interi positivi  $a_i$ . Il programma calcola il numero magico  $n$  come il numero di divisori interi di  $a_1 \times a_2 \times \dots \times a_{10}$  e ne visualizza la cifra meno significativa. Gestire il caso d'errore in cui il file indicato non esiste o non è accessibile, mediante un opportuno messaggio; se il file c'è, contiene senz'altro 10 valori interi positivi.

**Esempio**

**Ingresso:** dati.txt (contenuto del file: 1 2 6 1 3 1 1 1 2 1)  
**Uscita:** 2

**Quesito 4 (8 punti)**

Punteggio ottenuto: ... /8

Scrivere un sottoprogramma che ricevuta in ingresso una stringa restituisce una struttura dati opportuna che contiene per ogni carattere presente nella stringa, il numero di volte che compare. La struttura dati creata e restituita deve essere fatta "su misura" in base alle informazioni da restituire al sottoprogramma chiamante, a partire dal contenuto specifico della stringa ricevuta in ingresso. Negli esempi riportati, ciò che viene mostrato è la struttura dati visualizzata, non una stampa di alcuni elementi di una struttura dati più grande.

**Esempio 1**

**Ingresso:** aiuole  
**Uscita:** a (1) i (1) u (1) o (1) l (1) e (1)

**Esempio 2**

**Ingresso:** erre  
**Uscita:** e (2) r (2)

**Quesito 5 (4 punti)**

Punteggio ottenuto: ... /4

Si consideri un sottoprogramma che riceve in ingresso una matrice di numeri interi, un intero  $i$  e qualsiasi altro parametro ritenuto strettamente necessario. La matrice contiene senz'altro solamente 0 e 1. Il sottoprogramma visualizza tutti gli elementi di quelle righe che nella  $i$ -esima colonna hanno valore 1, e tutti i valori di quelle colonne che nella  $i$ -esima riga hanno valore 0.

La dichiarazione della matrice nel chiamante è la seguente:

```
...
int mat[N][M];
...
```

**Quesito 6 (4 punti)**

Punteggio ottenuto: .../4

Sia dato il seguente programma:

```
int main(int argc, char * argv[])
{
    pid_t * v, son;
    int status, inorder;
    int n, i;

    n = atoi(argv[1]);
    if(v = (pid_t *)malloc(sizeof(pid_t)*n)){
        for(i = 0; i < n; i++){
            *(v+i) = fork();
            if(*(v+i) == 0){
                sleep(2);
                exit(0);
            }
        }
        inorder = 1;
        for(i = 0; i < n; i++){
            son = wait(&status);
            if(son != *(v+i))
                inorder = 0;
        }
        printf("%d\n", inorder);
    } else
        printf("ERRORE: impossibile allocare memoria per %d dati", n);
    return 0;
}
```

Rispondere alle domande elencate sotto:

1. quando il programma visualizza 1?
2. è possibile che questa condizione non si verifichi mai?
3. a quanti processi dà luogo questo programma (incluso il processo "principale") quando l'utente lo esegue con la chiamata `prog 5`?