

Cognome: \_\_\_\_\_ Nome: \_\_\_\_\_ Matricola: \_\_\_\_\_ Voto: \_\_\_\_/30

| Quesito       | 1 | 2 | 3 | 4.1 | 4.2 | Tot |
|---------------|---|---|---|-----|-----|-----|
| Punteggio Max | 5 | 5 | 8 | 8   | 4   | 30  |
| Valutazione   |   |   |   |     |     |     |

**Istruzioni:**

- Il tempo massimo a disposizione per svolgere la prova è di 1ora e 45 minuti.
- È vietato consultare appunti e utilizzare calcolatrici, telefoni, PC o qualsiasi dispositivo elettronico.
- Il voto minimo per superare la prova è 18.

**Esercizio 1. Codifica Binaria (5 punti).** 1. Dati i due numeri  $A = -51_{10}$  e  $B = -98_{10}$ , effettuare la conversione in base 2, notazione complemento a 2, **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. **Si motivi la risposta e si mostrino i passaggi eseguiti.**

**Esercizio 2. Funzioni Ricorsive (5 punti).** Si supponga di avere, già presente in memoria (non è quindi necessario crearla), una lista che memorizza una sequenza di interi. I nodi della lista sono definiti in base al seguente tipo:

```
typedef struct nodo {  
    int val;  
    struct nodo * next;  
} Nodo;
```

Si scriva una **funzione ricorsiva in C** che, ricevuta come parametro la testa della lista, modifichi i valori nella lista sommando a ogni elemento il valore dell'elemento precedente. Il primo elemento resta invariato.

Es. : Data la lista 1-> 2-> 10-> 3-> 6 la funzione la modificherà in 1 -> 3 -> 12 -> 13 -> 9

**Esercizio 3. Array e memoria dinamica (8 punti).** Si supponga di avere, già presente in memoria, l'array **iniziale** di dimensione massima predefinita **N**, che memorizza una sequenza di caratteri. Tale sequenza può includere sotto-sequenze delimitate da parentesi tonde.

Esempio di sequenza memorizzata dall'array **iniziale[N]** :

**A X ( c d c d c g c ) b e ( b a b a d e a f a k a ) F ( x x ) j**

Si scriva una funzione in C che, ricevuto come parametro l'array **iniziale**, crei e restituisca all'ambiente chiamante un secondo array, **finale**, in cui a ogni sotto-sequenza delimitata da parentesi tonde si sostituisca **il carattere che nella sotto-sequenza compare più frequentemente**.

Il nuovo array deve avere una **dimensione strettamente necessaria a contenere i suoi elementi**.

Esempio: dato l'array iniziale riportato sopra, il nuovo array creato dalla funzione memorizzerà la sequenza

**A X ( c ) b e ( a ) F ( x ) j**

e la sua dimensione sarà 15.

Si può supporre che la sequenza di caratteri memorizzata nella lista sia ben formata e cioè che: i) per ogni parentesi che apre una sotto-sequenza ce ne sia una successiva che la chiude; ii) non ci siano intersezioni tra coppie di parentesi.

**Esercizio 4. Matrici e file (11 punti).** Sia data una matrice quadrata di dimensione costante **N**, **mat[N][N]**, che memorizza valori interi.

**Punto 1 (8 punti).** Si definisca una funzione in grado di individuare e segnalare eventuali picchi, cioè elementi con valori che superano quelli degli 8 elementi che li circondano di almeno 10 unità.

Esempio: data la seguente matrice

|   |           |    |    |    |
|---|-----------|----|----|----|
| 1 | 3         | 10 | 18 | 17 |
| 2 | <b>21</b> | 8  | 8  | 8  |
| 8 | 6         | 7  | 7  | 7  |
| 4 | <b>22</b> | 5  | 9  | 20 |
| 9 | 12        | 11 | 1  | 0  |

gli elementi evidenziati sono picchi perché più grandi di almeno 10 unità rispetto agli elementi sul loro contorno. N.B. Per l'analisi dei picchi non sono da considerare gli elementi sul contorno della matrice (righe e colonne di indice 0 e N-1).

La funzione:

- riceve come parametri la matrice da analizzare e il puntatore a un file di testo per memorizzare l'output, oltre a eventuali altri parametri ritenuti necessari.
- Scrive sul file le coordinate dei picchi individuati, un picco per ogni riga nel formato <indice riga, indice colonna>. Per esempio, per la matrice d'esempio riportata sopra, la funzione scriverà nel file le due righe:

1 1

3 1

Si suppone che la matrice analizzata abbia sempre almeno un elemento che sia picco.

**Punto 2 (4 punti).** Si definisca il main di un programma (completo di opportune dichiarazioni a livello globale e locale) che:

- legga da riga di comando (parametri del main) il nome del file su cui memorizzare le coordinate dei picchi
- gestisca l'immissione dei valori della matrice da parte dell'utente (input da tastiera)
- invochi la funzione definita al punto precedente, passando opportuni parametri tra cui la matrice letta in input e il file su cui scrivere le coordinate dei picchi.