

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

Quesito	1	2	3	4	5	6.1	6.2	Tot
Punteggio Max	3	3	5	6	5	6	2	30
Valutazione								

**Istruzioni:**

- Il tempo massimo a disposizione per svolgere la prova è di 1 ora 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 (3 punti)**

1. Dati i due numeri  $A = -27_{10}$  e  $B = 51_{10}$ , 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 e perché. **Si motivi la risposta e si mostrino i passaggi eseguiti.**

**Esercizio 2. Funzioni Ricorsive (3 punti).** Si scriva una funzione ricorsiva che, ricevuto come parametro un array di interi (e ogni altro parametro ritenuto necessario), individui e restituisca al chiamante il numero di valori pari contenuti nell'array.

**Esercizio 3. Liste dinamiche (5 punti).** Scrivere una funzione che, ricevuta in ingresso una lista dinamica  $L$  che memorizza interi, modifichi la lista cancellando eventuali ripetizioni. Per esempio, data la lista:

4-> 3-> 1-> 3-> 7-> 6-> 7-|

la lista sarà modificata nel seguente modo:

4 -> 3-> 1-> 7-> 6--|

**Esercizio 4. Matrici (6 punti).** Scrivere un sottoprogramma che, ricevuta in ingresso una matrice quadrata di interi (e qualsiasi altro parametro ritenuto strettamente necessario), **verifichi se esistono almeno una riga e una colonna identiche**, cioè contenenti gli stessi valori e nello stesso ordine. In caso positivo il sottoprogramma restituisce 1 altrimenti 0.

**Esercizio 5. Array (5 punti).** Si vuole definire **una funzione che verifichi la validità di una password**. La funzione riceve in ingresso la stringa *passwd*, che rappresenta la password da verificare, un array di caratteri, *speciali*, che memorizza un insieme di caratteri speciali ammessi, e due interi, *min* e *max*. La funzione quindi verifica che la stringa:

- sia lunga almeno *min* caratteri e non superi *max* caratteri,
- contenga almeno uno tra i caratteri memorizzati nella stringa *speciali*.

Nel caso in cui la password sia valida, la funzione restituisce 1, altrimenti restituisce:

- -2 se la lunghezza è minore di min,
- -1 se è maggiore di max,
- 0 se non contiene caratteri speciali.

**Esercizio 6. Liste Dinamiche e File (8 punti).** Siano dati due file che memorizzano ognuno una sequenza ordinata di interi positivi con possibili ripetizioni,  $s1$  e  $s2$ . Il numero di valori nelle due sequenze non è noto a priori. A partire da  $s1$  ed  $s2$ , si vuole creare una nuova sequenza che fonda i valori di  $s1$  e  $s2$  in modo ordinato e senza ripetizioni.

Per esempio, se  $s1 = \{0\ 1\ 3\ 4\ 5\ 5\ 6\ 7\ 7\ 7\ 8\ 8\}$  ed  $s2 = \{3\ 3\ 3\ 3\ 3\ 3\ 4\ 5\ 9\}$ , allora la nuova sequenza sarà  $s3 = \{0\ 1\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}$ .

La nuova sequenza  $s3$  deve essere memorizzata su un nuovo file.

**Punto 1 (6 punti).** Si scriva una funzione che riceve come parametri i puntatori a tre file distinti: i primi due memorizzano le due sequenze  $s1$  e  $s2$ , sul terzo dovrà essere salvata la nuova funzione.

La funzione dovrà **calcolare e scrivere nel terzo file la nuova sequenza  $s3$** .

Restituirà all'ambiente chiamante: 1 se il salvataggio della nuova sequenza va a buon fine, 0 altrimenti.

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

- riceva da **riga di comando** (parametro del main) il nome dei tre file
- invochi la funzione definita al punto precedente
- stampi un messaggio che indichi se il salvataggio della nuova sequenza è andato a buon fine