Fondamenti di Informatica - A.A. 2021-2022

Prof. ssa Maristella Matera Appello del 26/01/2022



Cognome:	Nome:	Matricola:	Voto:	/30
				_, 00

Quesito	1	2	3	4.1	4.2	4.3	Tot
Punteggio Max	4	4	5	9	5	3	30
Valutazione							

Istruzioni:

- Il tempo massimo a disposizione per svolgere la prova è di 2 ore.
- È vietato consultare appunti e utilizzare calcolatrici, telefoni, PC o qualsiasi dispositivo elettronico.
- Il voto minimo per superare la prova è 18.

Quesito 1 (4 punti).

Dati i due numeri A = - 24_{16MS} e B = -30_{10MS}, codificarli in binario complemento a 2 utilizzando il **numero** minimo di bit necessari a rappresentarli entrambi. Si eseguano quindi le operazioni A+B e A-B, indicando esplicitamente se si verifica overflow e motivando la risposta. Mostrare i passaggi seguiti.

Quesito 2 (4 punti). Si scriva una **funzione ricorsiva** che, ricevuti come parametri un array di interi e altri parametri eventualmente ritenuti necessari, restituisce 1 se nell'array il numero di valori positivi è maggiore di quelli negativi, -1 altrimenti.

Quesito 3 (max 8 punti).

Si vuole definire una funzione che, ricevuto in ingresso un array contenente *n* valori interi, e qualsiasi altro parametro ritenuto strettamente necessario, verifichi se è possibile dividere l'array in due partizioni aventi somma degli elementi uguale.

ESEMPIO 1. Dato l'array {1, 2, 3, 10, 8, 8} è possibile individuare le due partizioni {1, 2, 3, 10} e {8, 8}, aventi entrambe somma degli elementi pari a 16.

ESEMPIO 2. Dato l'array {1, 2, 3, 5, 8, 8}, non è possibile alcun partizionamento.

Lo studente deve scegliere di svolgere una sola delle seguenti versioni della funzione.

Versione 1 (5 punti). Se il partizionamento è possibile (ESEMPIO 1), la funzione restituisce all'ambiente chiamante **l'indice del primo elemento della partizione destra** (4 nell'ESEMPIO 1). Restituisce -1 nel caso in cui il partizionamento non sia possibile (ESEMPIO 2).

Versione 2 (8 punti). Se il partizionamento è possibile (ESEMPIO 1), la funzione costruisce e restituisce all'ambiente chiamante **un nuovo array** che contiene la partizione destra ({8.8} nell'ESEMPIO 1) e la cui dimensione è tale da contenere esattamente il numero di valori nella partizione (2 in ESEMPIO 1). Restituisce valore NULL nel caso in cui il partizionamento non sia possibile (ESEMPIO 2).

Quesito 4. (17 punti).

Si vuole scrivere un programma in grado di analizzare un testo per determinare alcune statistiche sulla frequenza delle parole in esso contenute.

Punto 1 (9 punti). Si definisca una funzione, *frequenza*, che riceve in ingresso il puntatore al file in cui è memorizzato il testo (e ogni altro parametro ritenuto necessario) e restituisce una lista dinamica che memorizza le parole trovate e la frequenza con cui queste ricorrono nel testo. Le parole valide per il conteggio sono quelle di **lunghezza minima 2**. La lista dovrà essere costruita mantenendo **l'ordinamento alfabetico delle parole**. N.B.: È possibile usare funzioni predefinite per la gestione delle stringhe (funzioni della libreria *string* per il calcolo della lunghezza, per il confronto, per l'assegnamento).

Punto 2 (5 punti). Si definisca una seconda funzione, *statistiche*, che ricevuta in ingresso la lista costruita dalla funzione *frequenza* (e ogni altro parametro ritenuto necessario), calcoli e restituisca il numero totale di parole contenute nella lista, la parola con frequenza minima e il valore della frequenza, la parola con frequenza massima e il valore della frequenza.

Punto 3 (3 punti). Si definisca il main del programma (completo di opportune dichiarazioni per il tipo del nodo della lista e della lista stessa e di ogni altra dichiarazione necessaria a livello globale e locale), che:

- riceve da riga di comando (parametro del main) il nome del file da cui leggere il testo
- invoca la funzione frequenza;
- invoca la funzione *statistiche* e stampa il numero totale di parole, la parola con frequenza; minima e il valore della frequenza, la parola con frequenza massima e il valore della frequenza.