

Università degli Studi di Trento Corso di Programmazione I – E3: 145430/145935 2° Appello – 08/02/2019 – a.a. 2018/2019		PC N°						
COGNOME NOME MATRICOLA <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>								
RITIRATO <input type="checkbox"/> FIRMA.....								

Completare le voci **Cognome**, **Nome** e **Matricola**, se tali voci saranno mancanti o non-leggibili elaborato **NON** verrà corretto.

NON è consentito utilizzo di alcun strumento cartaceo, elettronico, informatico etc. durante la prova.
 Fa parte della valutazione la leggibilità del codice C/C++.

Scrivere le risposte su questo foglio e riconsegnare. I risultati verranno pubblicati su DOL
Tempo a disposizione: 120 minuti.

PARTE TEORICA: 12 PUNTI – PUNTEGGIO MINIMO 6 PUNTI

[Domanda 1 – punti 3] Data la seguente funzione:

```
int elabora(int n1, int n2) {
    int a, b;
    a = n1 + 1;
    b = n2 - 1;
    cout << "a=" << a << " b=" << b << endl;
    if (b <= 0) {
        return a;
    } else {
        return elabora(a,b);
    }
}
```

Scrivere nel riquadro cosa viene stampato dalla funzione in seguito all'invocazione:

```
cout << elabora(4,3);
```

[Domanda 2 - punti 2] Dire (mostrando tutti i passaggi) a quale valore decimale corrisponde il numero binario 101101_2 in caso di [A] numero intero senza segno [B] numero intero con segno [C] numero intero codificato in CA2

[Domanda 3 - punti 5]

[A] Data la seguente sequenza di valori:

4, 1, 8, 2, 7, 3, 5, 5, 1

costruire un albero binario di ricerca (BST),
inserendo in ordine i valori della sequenza e
mettendo i valori uguali nel sottoramo di DX.

[B] Scrivere una funzione che conta quante foglie con valore pari sono presenti nell'albero BST.
Quale è la complessità dell'algoritmo della funzione così realizzata?

[C] Per l'albero così realizzato inoltre calcolare

Cammino									
Altezza									
Visita pre-ordine									
Visita post-ordine									
Visita in-ordine									
Visita level-ordine									

[Domanda 4 - punti 2] È dato un vettore di N elementi, composto da numeri 0 e 1.

Definire una funzione **cerca** a cui passare opportunamente il vettore ed un parametro intero **d**.

Verificare se esiste una sequenza di elementi di valore 0 adiacenti la cui lunghezza è maggiore del valore **d** ed in questo caso ritornare TRUE altrimenti ritornare FALSE.

PARTE PRATICA: 21 PUNTI – PUNTEGGIO MINIMO 11 PUNTI
--

[A - punti 5] Inserire i seguenti tipi in un file **dati.h** e implementare i metodi di stampa ed i costruttori:

```
#define DIM 3
typedef enum Tpista { NERA, ROSSA, BIANCA } Tpista;
typedef struct Timpianto{
    char nomeImpianto[20];    unsigned int lunghezza;
    float costo;
    Tpista tipoPista;
    Timpianto ()    { /* da implementare */ }
    void stampa() { /* da implementare → formato: vedi punto E */ }
} Timpianto;
typedef struct Tcoda {
    Timpianto* is; // array con allocazione dinamica
    int head, tail; int dim, n;
    Tcoda (int _dim) { /* da implementare */ }
    void stampa() { /* da implementare */ }
    void put(Timpianto d) { /* da implementare */ }
    Timpianto get() { /* da implementare */ }
    bool isFull() { /* da implementare */ }
    bool isEmpty() { /* da implementare */ }
} Tcoda; // CODA realizzata con array
```

[B - punti 1] Modificare il file **main.cpp** includendo il file **dati.h** ed inserendo il seguente codice nel **main**:

```
Tcoda* impiantiRisalita = new Tcoda(DIM); Timpianto imp;
for (int i=0; i<5; i++) {
    newImpianto(&imp);
    addImpianto(impiantiRisalita, imp);
}
stampaImpianti(impiantiRisalita);
cout << cancellaEconta (impiantiRisalita, NERA);
```

[C - punti 3] Implementare una funzione **newImpianto** che riceve in input l'indirizzo di una variabile di tipo **Timpianto** **imp** e la inizializza con:

- campo **tipoPista** valore casuale tra NERA, ROSSA, BIANCA;
- campo **lunghezza** valore letto da tastiera, tra 500 e 2400, con controllo input.
- campo **costo** valore casuale tra 25.00 e 35.50;
- campo **nomeImpianto** letta da tastiera.

[D - punti 5] Implementare una funzione **addImpianto** che riceve come input una variabile di tipo **Tcoda*** **cod**, ed una variabile **imp** di tipo **Timpianto** inserisce nella coda usando il metodo **put**. Se la coda è piena (usare **isFull**) aggiunge l'impianto ad un file denominato **"impiantiExtra.txt"** nella forma:

nomeImpianto (tipoPista) **LUN**=lunghezza **costo**=costo

[E - punti 2] Implementare una funzione **stampaImpianto** che riceve in input una variabile di tipo array **Tcoda *** **cod**, e stampa il contenuto della coda utilizzando opportunamente il metodo **stampa** di **Tcoda**. Nella forma (per l'enum stampare etichetta in modo opportuno):

nomeImpianto (tipoPista) **LUN**=lunghezza **costo**=costo

[F - punti 5] Implementare una funzione **cancellaEconta** che riceve come input una variabile di tipo array **Tcoda*** **cod** e **Tpista** **tp**. Estrae tutti gli elementi della coda (tramite funzione **get**), conta tutti gli impianti con tipoPista uguale a **tp** e restituisce tale valore.

È possibile creare strutture, funzioni, metodi aggiuntivi per realizzare gli esercizi proposti.
Il codice commentato NON verrà corretto!