

| | | | | | | | | |
|--|--|-------|--|--|--|--|--|--|
| Università degli Studi di Trento Corso di Programmazione I – E3: 145430/145935 2° Appello – 08/02/2022 – a.a. 2021/2022 | | 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 <table border="1" style="display: inline-table; vertical-align: middle; width: 40px; height: 20px;"></table> FIRMA PER RITIRARSI | | | | | | | | |

NON è consentito utilizzo di alcun strumento elettronico, informatico etc. durante la prova.
 Fa parte della valutazione la leggibilità *del codice C/C++*.
Tempo a disposizione: 100 minuti.

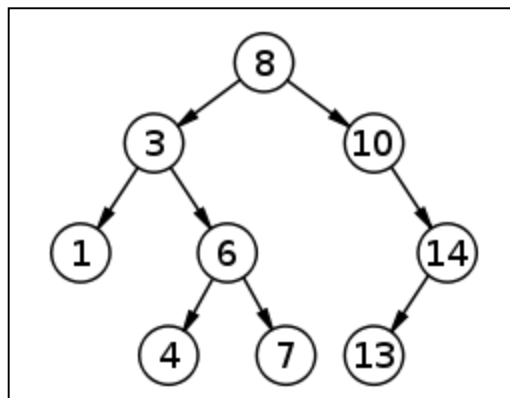
| |
|--------------------------------|
| PARTE TEORICA: 12 PUNTI |
|--------------------------------|

[Domanda1 – 4 punti]

[A] Dato il seguente tipo di dato e la funzione **costruisci**:

```
typedef struct Tnodo {
    int dato;
    Tnodo *DX, *SX;
    Tnodo(int d, Tnodo *sx, Tnodo *dx){
        dato=d; SX=sx; DX=dx;
    }
} Tnodo;

Tnodo* costruisci(int d, Tnodo* sx, Tnodo* dx
){ Tnodo* root;
  root = new Tnodo(d, sx, dx);
  return root;
}
```



Scrivere il codice che invocando opportunamente la funzione **costruisci** crei l'albero di figura facendo in modo che la variabile **a** dichiarata nel main punti alla radice dell'albero creato.

```
void main(){
    Tnodo* a;
    //...
}
```

[Domanda2 - 3punti]

Si scriva l'output (istruzione **printf**), nel riquadro a fianco, che viene prodotto durante l'esecuzione del programma se venisse invocata la funzione **MergeSort(A, 0, 5)** con A un array di interi.

```
void MergeSort (int A[], int p, int r){
    if (p < r) {
        q = floor((p+r)/2);
        printf("(p=%d, r=%d, q=%d)", p, r, q);
        MergeSort (A, p, q);
        MergeSort (A, q+1, r);
        Merge(A, p, q, r);
    }
}
```

[Domanda 3 - punti 3]

Quanto vale 1AD (esadecimale, base 16) in base 10? In base 8? E in base 2? (mostrare il procedimento, tutti i passaggi in dettaglio!)

[Domanda 4 - punti 2]

[4A] Un puntatore è

- SI ☐ NO ☐ ...Un simbolo usato nel linguaggio C per puntare all'indirizzo di un valore
SI ☐ NO ☐ ...Un indirizzo di memoria inizializzato automaticamente dal compilatore
SI ☐ NO ☐ ...Un tipo di dato usato per memorizzare variabili
SI ☐ NO ☐ ...Una variabile che memorizza un indirizzo di una variabile

[4B] Date tre variabili booleane A, B e C si consideri una funzione booleana f(A, B, C) che vale 1 quando esattamente una delle tre variabili vale 1. Si descriva la funzione f(A, B, C) usando gli operatori booleani and, or e not. Tale descrizione è:

- SI ☐ NO ☐ ... $(A \text{ and not } B \text{ and not } C) \text{ or } (\text{not } A \text{ and } B \text{ and not } C) \text{ or } (\text{not } A \text{ and not } B \text{ and } C)$
SI ☐ NO ☐ ... $(A \text{ or not } B \text{ or not } C) \text{ and } (\text{not } A \text{ or } B \text{ or not } C) \text{ and } (\text{not } A \text{ or not } B \text{ or } C)$
SI ☐ NO ☐ ... $A \text{ and } B \text{ and } C$
SI ☐ NO ☐ ... $\text{not } A \text{ or not } B \text{ or not } C$

PARTE PRATICA: 21 PUNTI

[A - punti 5] Inserire i seguenti tipi in un file **dati.h** e implementare i metodi di stampa ed i costruttori indicati ed altri metodi, costruttori o distruttori considerati utili in **dati.cpp**.

```
#define DIM 2
typedef enum Tbattello { ALISCAFO, CROCIERA, TRAGHETTO } Tbattello;
typedef struct Tnave{
    char nomeNave[20]; unsigned int annoVaro;
    float stazza;
    Tbattello tipoBattello;
    Tnave () { /* da implementare */ }
    void stampa() const { /* da implementare → formato: vedi punto E */ }
} Tnave;
typedef struct Tnodo {
    Tnave nave;
    Tnodo *next, *prev;
    Tnodo (Tnodo *n, Tnodo *p, Tnave _nave) { /* da implementare */ }
    Tnodo () { /* da implementare */ }
    void stampa() const { /* da implementare */ }
} Tnodo; // LISTA FIFO doppiamente concatenata
```

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

```
Tnodo* porti[DIM];
Tnave nave;
/* inizializzare la variabile porti in modo opportuno*/
for (int i=0; i<5; i++) {
    newNave(&nave);
    addNave(porti, DIM, nave);
}
stampaPorti(porti, DIM);
rimuoviEContaNavi (porti, DIM);
```

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

- campo **tipoBattello** valore casuale tra ALISCAFO, CROCIERA, TRAGHETTO;
- campo **annoVaro** valore letto da tastiera, tra 1995 e 2022, con controllo input.
- campo **stazza** valore casuale con 2 decimali tra 15000,00 e 50000,00;
- campo **nomeNave** letta da tastiera.

[D - punti 5] Implementare una funzione **addNave** che riceve come input un array di tipo **Tnodo*** **por**, la sua dimensione effettiva **dim** ed una variabile **nave** di tipo **Tnave**. La funzione inserisce nella lista **por[x]** (con x posizione casuale nell'array) l'elemento **nave** (modalità **insertLast**).

[E - punti 2] Implementare una funzione **stampaPorti** che riceve in input un array di tipo **Tnodo*** **por** e la dimensione effettiva dell'array **dim**. Stampa il contenuto delle liste utilizzando opportunamente il metodo **stampa** di **Tnodo** nella forma (per l'enum stampare etichetta in modo opportuno):

nomeNave (tipoBattello) stazza **anno varo**=annoVaro

[F - punti 5] Implementare una funzione **rimuoviEContaNavi** che riceve come input un array di tipo **Tnodo*** **por**, la sua dimensione effettiva **dim**. La funzione elimina tutti gli elementi dalle liste FIFO (usando la modalità **removeFirst**) e salva in un file denominato "**navi.txt**" l'indice dell'array con il numero di navi eliminate, nella forma:

CODA FIFO POS= indice **NUMNAVI**= valore

Esempio: **CODA FIFO POS**= 0 **NUMNAVI**= 3
 CODA FIFO POS= 1 **NUMNAVI**= 2