

PARTE PRATICA: 21 PUNTI – PUNTEGGIO MINIMO 11 PUNTI

[A - punti 3] Inserire i seguenti tipi in un file **dati.h** e implementare i metodi di stampa ed i costruttori:

```
#define DIM 2
typedef enum Trichiasta { WEB, TELEFONO, UFFICIO } Trichiasta;
typedef struct Tordine{
    char materiale[20];
    int quantita;
    float prezzo;
    Trichiasta tipoRichiesta;
    Tordine () { /* da implementare */ }
    void stampa() { /* da implementare → formato: vedi punto E */ }
} Tordine;
typedef struct Tnodo {
    Tordine dato;
    Tnodo* next;
    Tnodo* prev;
    Tnodo () { /* da implementare */ }
    Tnodo (Tordine o, Tnodo* n, Tnodo* p) { /* da implementare */ }
    void stampa() { /* da implementare */ }
} Tnodo;
```

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

```
Tnodo* negozio[DIM];
for (int i=0; i<DIM; i++) { negozio[i] = NULL; }
Tordine o;
for (int i=0; i<5; i++) { newOrdine(&o); addOrdine(negozio, DIM, o); }
stampaNegozi(negozio, DIM);
o = cercaOrdine(negozio, DIM);
o.stampa();
```

[C - punti 4] Implementare una funzione **newOrdine** che riceve in input per riferimento una variabile di tipo **Tordine o** e la inizializza con:

- campo **tipoRichiesta** valore casuale tra WEB, TELEFONO, UFFICIO;
- campo **quantita** valore letto da tastiera, tra 1 e 20, con controllo input.
- campo **prezzo** valore casuale tra 10.00 e 99.99;
- campo **materiale** letta da tastiera.

[D - punti 4] Implementare una funzione **addOrdine** che riceve come input una variabile di tipo array **Tnodo* neg**, la dimensione dell'array **dim** e variabile di tipo **Tordine o** e la inserisce nella listaFIFO **neg[x]** (con **x** valore casuale tra 0 e 1) (**insertFirst**).

[E - punti 4] Implementare una funzione **stampaNegozi** che riceve in input una variabile di tipo array **Tnodo * neg**, la dimensione dell'array **dim** e stampa il contenuto delle liste FIFO utilizzando opportunamente il metodo **stampa di Tnodo**. Nella forma (per enum stampare etichetta):

materiale **QTA**=quantita (tipoRichiesta) **prezzo**=prezzo

[F - punti 5] Implementare una funzione **cercaOrdine** che riceve come input una variabile di tipo array **Tnodo* neg** e la dimensione dell'array **dim** e cerca l'ordine che ha il maggior costo medio (prezzo/quantita). La funzione, oltre a restituire l'ordine trovato, aggiunge l'ordine stesso ad un file denominato "**costoso.txt**" nella forma:

materiale **QTA**=quantita (tipoEstrazione) **prezzo**=prezzo

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