Università degli Studi di Trento Corso di Programmazione I – E3: 145430/145935 1° Appello – 11/01/2022 – a.a. 2021/2022								PC N°
COGNOME								
NOME								
MATRICOLA								
RITIRATO	F	IRMA P	ER RI	TIRO				

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 - 6punti] [A] Data la seguente sequenza di valori 4, 4, 6, 6, 2, 1, 3, 3, 7 costruire un albero binario di ricerca (BST), usando in ordine tali valori e mettendo i valori uguali nel sottoramo di SX.	

[B] Cosa calcola la seguente funzione?

Che valore restituisce passandogli il riferimento della root dell'albero BST appena creato?

```
int Calcola (Tnode * 1) {
  if (1 == NULL) return -1;
  int u = Calcola(1->DX);
  int v = Calcola(1->SX);
  if (u > v) return u+1;
  return v + 1;
Typedef struct Tnode {
   int dato;
   Tnode * DX;
   Tnode * SX;
} Tnode * SX;
} Tnode;
```

[C] Per l'albero così realizzato inoltre calcolare

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

[Domanda2 - 2punti]
Dire se le seguenti affermazioni sono corrette:
SI □ NO □ & è operatore di puntamento
SI □ NO □ & serve per accedere all'entità
SI □ NO □> serve per generare l'indirizzo di una entità
SI □ NO □> operatore di indirizzamento
SI NO operatore * serve per accedere al valore presente in memoria
SI □ NO □ operatore "freccia" -> serve per accedere ad una variabile
SI NO & serve per accedere ai campi di un record tramite puntatore
SI \square NO \square ** operatore di doppio indirizzamento serve per la doppia dereferenziazione
[Domanda3 - 4punti]
[A] Dare la definizione delle notazioni Θ Ω 0

[B] Esprimere la funzione $T(n) = n^4/13 - 55 n^3 + \log_2(n)$ nelle notazioni: $\Theta \Omega O$ motivando la soluzione

PARTE PRATICA: 21 PUNTI

[A - punti 8] Inserire i seguenti tipi in un file dati.h e implementare i metodi, costruttori e distruttori in dati.cpp.

Inserire nella definizione della struttura **TipoCodaFIFO**: costruttore specifico, costruttore a zero parametri e distruttore, metodi: stampa, codaIsEmpty, codaIsFull, get, put e eventualmente altri metodi necessario e/o utili.

```
#define DIM 3
 typedef enum Tfrutto { MELA, PESCA, PERA } Tfrutto;
 typedef struct Tproduzione {
  char piantagione[20]; unsigned int numeroPiante;
   float quantita;
  Tfrutto tipoFrutto;
  Tproduzione () { /*da implementare*/ }
                   { /*da implementare*/ }
  void stampa()
 } Tproduzione;
typedef struct TipoCodaFIFO {
   int head, tail; //posizione inserimento, posizione prelevamento
   int dim;
                    //numero elementi array==coda
   int n;
                    //numero elementi presenti nella coda
                   // array dinamico circolare
   Tproduzione* s;
} TipoCodaFIFO;
```

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

```
TipoCodaFIFO* pf[DIM]
Tproduzione p;
/*da implementare init array code FIFO pf dimensione 3 elementi*/
for(int i=0; i<6; i++) { newProduzione(&p); addProduzione(pf, DIM, p); }
stampaProduzioni(pf, DIM);
cout << estraiProduzioni(pf, DIM, MELA);</pre>
```

- [C punti 3] Implementare una funzione newProduzione che riceve in input l'indirizzo di una variabile di tipo Tproduzione pe la inizializza con:
 - campo piantagione letta da tastiera;
 - campo numeroPiante valore letto da tastiera, tra 1000 e 9999, con controllo input.
 - campo quantita valore casuale tra 100.00 e 2000.00;
 - campo tipoFrutto valore casuale tra MELA, PESCA, PERA.
- [D punti 4] Implementare una funzione addProduzione che riceve come input una variabile di tipo TipoCodaFIFO* produzioni[], la dimensione dell'array dim ed una variabile p di tipo Tproduzione. La funzione sceglie in modo casuale la coda FIFO ed inserisce l'elemento p. (Inserimento in testa head). Nel caso la coda fosse piena (codaIsFull) inserire la produzione in un file denominato "noProd.txt" nella forma (per l'enum stampare etichetta in modo opportuno):

```
piantagione (tipoFrutto) quantita numeroPiante
```

[E - punti 2] Implementare una funzione stampaProduzioni che riceve come input una variabile di tipo TipoCodaFIFO* produzioni[] e la dimensione dell'array dim e stampa il contenuto delle liste utilizzando opportunamente il metodo stampa di TcodaFIFO nella forma (per l'enum stampare etichetta in modo opportuno):

```
piantagione (tipoFrutto) quantita numeroPiante
```

[F - punti 3] Implementare una funzione estraiProduzioni che riceve come input una variabile di tipo TipoCodaFIFO* produzioni[], la dimensione dell'array dim e Tfrutto tf. La funzione elimina tutti gli elementi dall'array di code FIFO (cancellazione da testa – head) e restituisce il numero di produzioni con tipoFrutto pari a tf