Dipartimento di Scienze e Tecnologie
Centro Direzionale Isola C4 – 84143 Napoli - Italy

Università degli Studi di Napoli "Parthenope"
www.uniparthenope.it

Corso di Programmazione e Laboratorio di Programmazione 2 – A.A. 2022-23 – CdL in Informatica
Appello del 23/07/2024 – Traccia n. 2

Nome e cognome: _____ Matricola: _____

Esercizio n.1

Si supponga di disporre dell'implementazione della seguente struttura dati **`mylist`**:

```
template <class ItemType>
class mynode {
public:
        ItemType data;
        mynode<ItemType> *next;
        mynode(ItemType i): data(i), next(nullptr) {};
        ~mynode() {};
};

template <class ItemType>
class mylist {
        mynode<ItemType> *first, *last, *current;
public:
        mylist(): first(nullptr), last(nullptr), current(nullptr) {};
        ~mylist();
        bool isEmpty();   // vero se la lista è vuota
        bool isAtFirst(); // vero se current punta al primo nodo;
        bool isAtLast();  // vero se current punta all'ultimo nodo;
        mylist<ItemType> *goFirst();
                          // assegna current al primo nodo della lista (first);
        mylist<ItemType> *goNext();
                          // sposta current al nodo succecssivo (current->next);
        ItemType getCurrentData();
                          // restituisce l'item contenuto nel nodo corrente,
                          // SENZA ESTRARRE IL NODO
        mylist<ItemType> *insert(ItemType i);
                          // inserisce un nuovo nodo con item i in fondo alla lista e
                          // vi assegna il puntatore current;
        ItemType extract();
                          // estrae il primo nodo della lista e restituisce il
                          // dato in esso contenuto
};
```

Utilizzando ESCLUSIVAMENTE e in maniera coerente con la loro dichiarazione i metodi forniti, si scriva la funzione **`equals()`** che prende come argomenti i puntatori a due oggetti di classe **`mylist`** e che restituisca **`true`**, se le liste corrispondenti sono entrambe vuote oppure se contengono esattamente gli stessi dati (nello stesso ordine); oppure **`false`** altrimenti. L'applicazione di **`equals()`** non deve distruggere le liste. Si ponga attenzione all'uso corretto della sintassi dei template;

Esercizio n.2

Si scriva la classe **`simpleTree`** che implementa un albero binario utilizzando un *vettore posizionale* di interi positivi **`T[]`**. Si implementino:

1. un costruttore **`simpleTree(int numnodi)`** che consenta all'utente di allocare un albero di taglia *numnodi* arbitraria, e che inizializzi tutte le celle dell'array al valore -1 (nodo assente);
2. il metodo **`add(data, i)`** che permetta di aggiungere il nodo **`i`** con item **`data`** all'albero (se non già presente);
3. il metodo **`InorderTrav(i)`**, che effettui la visita *inorder* dell'albero radicato sul nodo **`i`**, enumerando i soli nodi presenti (**`T[i]!=-1`**).
4. *A discrezione del candidato:* ogni altro metodo ritenuto necessario per implementare quelli richiesti.