Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL 2001 Veri Yapıları
Arasınav, 17.11.2016, 15:00, D-2, D-9
Süre : **90** Dakika

CEVAPLAR

```
void traverse(Node* v)
{
        cout << v->elt << " ";

        if (v->left != NULL)
        {                                       // A
                traverse(v->left);
                cout << v->elt << " ";
        }                                       // A

        if (v->right != NULL)
        {                                       // B
                traverse(v->right);
                cout << v->elt << " ";
        }                                       // B
}
```

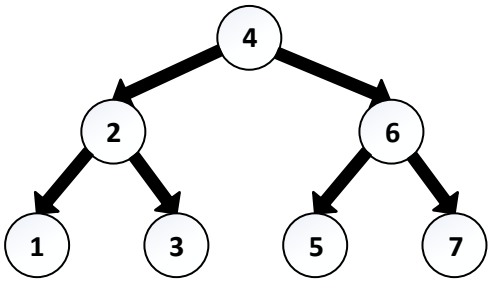
1. **traverse()** fonksiyonunun **main()**'de aşağıdaki ağacın **root**u ile çağrıldığı varsayıldığında:

a) `// A` ile temsil edilen kıvırcık parantezler kapatılırsa çıktı ne olur? **(15P)**

4 2 1 1 2 3 3 2 4 6 5 5 6 7 7 6 4

b) `// B` ile temsil edilen kıvırcık parantezler kapatılırsa çıktı ne olur? **(15P)**

4 2 1 1 2 3 3 2 4 6 5 5 6 7 7 6 4




```
void addBack(const string& e, const int& i)
{
   CircularlyNode* v = new CircularlyNode;
   v->elem = e;
   v->score = i;

   if (cursor == NULL)
   {
      v->next = v;
      cursor  = v;
   }
   else
   {
      v->next         =  cursor->next;
      cursor->next    =  v;
      cursor          =  cursor->next;
   }
}
```

2. Dairesel listenin sonuna düğüm ekleyen **addBack()** fonksiyonunu tamamlayınız. **(20P)**

```
void insertOrdered(const string& e, const int& i)
{
  DoublyNode* newNode = new DoublyNode;
  newNode->elem       = e;
  newNode->score      = i;

  DoublyNode* current = header;

  while (current->next != trailer)
  {
     if(newNode->score >= current->next->score)
        current = current->next;
     else
        break;
  }

  newNode->next           = current->next;
  newNode->prev           = current;
  current->next->prev     = newNode;
  current->next           = newNode;
}

int main()
{
  DoublyLinkedList list;

  list.insertOrdered("Paul",  720);
  list.insertOrdered("Rose",  590);
  list.insertOrdered("Anna",  660);
  list.insertOrdered("Mike", 1105);
  list.insertOrdered("Rob" ,  750);
  list.insertOrdered("Jack",  510);
  list.insertOrdered("Jill",  740);

}
```

3. **`insertOrdered()`** fonksiyonunu tamamlayınız. **(30P)**

```
#include <iostream>
#include <stack>
using namespace std;

void main()
{
	stack<int> stl_stack;

	int temp = 61;

	while (temp != 0)
	{
		stl_stack.push(temp % 2);
		temp = temp / 2;
	}

	while (!stl_stack.empty())
	{
		if (stl_stack.top() == 1)
			cout << '1';
		else
			cout << '0';

		stl_stack.pop();
	}

	getchar();
}
```

4. Yukarıdaki programın çıktısı nedir? **(20P)**

1 1 1 1 0 1