**BURSA ULUDAĞ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ**

**2022 – 2023 EĞİTİM YILI BAHAR DÖNEMİ**

**VERİ YAPILARI DERSİ**

**PALİNDROME ÖDEVİ**

**Ad Soyad : Mert Gökmen**

**Numara : 032190004**

**Projede verilen tasarım koşulları : Stack ve Queue kullanarak stringin palindrome olup olmadığını göstermek.**

**Kullanılan veri modeli : Stack ve Queue.**

**Varsayımlar : isPalindrome fonksiyonuna verilen değerin veri tipinin string olması.**

**Örnek Çıktılar :**

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Kod (Bağlı liste kullanarak) :**

using System;

namespace Prog

{

class Program

{

class node

{

public char veri;

public node ileri;

public node geri;

public node(char x)

{

veri = x;

ileri = null;

geri = null;

}

}

class queue

{

public node bas;

public node son;

public queue()

{

this.bas = null;

this.son = null;

}

public void enqueue(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.geri = this.son;

this.son.ileri = node;

this.son = node;

}

}

public int dequeue()

{

if (this.size() == 1)

{

int x = this.bas.veri;

this.bas = null;

this.son = null;

return x;

}

else if (this.size() > 1)

{

int x = this.bas.veri;

this.bas = this.bas.ileri;

this.bas.geri = null;

return x;

}

else

{

Console.WriteLine("Listede silmek için eleman yok.");

return -1;

}

}

public void printQueue()

{

if (this.size() != 0)

{

node temp = this.bas;

while (temp != null)

{

Console.Write(temp.veri + " <--> ");

temp = temp.ileri;

}

Console.WriteLine();

}

}

public void printListReverse()

{

node temp = this.son;

while (temp != null)

{

Console.Write(temp.veri + " <--> ");

temp = temp.geri;

}

Console.WriteLine("NULL");

}

public int size()

{

int count = 0;

node temp = this.bas;

while (temp != null)

{

count++;

temp = temp.ileri;

}

return count;

}

public char peek()

{

if(this.size() > 0)

return this.son.veri;

return '0';

}

public void addtohead(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.ileri = this.bas;

this.bas.geri = node;

this.bas = node;

}

}

public void addtoend(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.geri = this.son;

this.son.ileri = node;

this.son = node;

}

}

public void addtomiddle(node node)

{

if (this.size() > 2)

{

node temp = this.bas;

for (int i = 0; i < this.size() / 2 - 1; i++)

{

temp = temp.ileri;

}

node.geri = temp;

node.ileri = temp.ileri;

temp.ileri = node;

}

else

{

Console.WriteLine("Ortaya eklemek için listenin uzunluğu en az 3 olmalıdır.");

}

}

}

class stack

{

public node bas;

public node son;

public stack()

{

this.bas = null;

this.son = null;

}

public void push(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.geri = this.son;

this.son.ileri = node;

this.son = node;

}

}

public int pop()

{

if (this.size() == 1)

{

int val = this.son.veri;

this.bas = null;

this.son = null;

return val;

}

else if (this.size() > 0)

{

int val = this.son.veri;

this.son = this.son.geri;

this.son.ileri = null;

return val;

}

else

{

this.bas = null;

this.son = null;

Console.WriteLine("Listede silmek için eleman yok.");

return -1;

}

}

public void printStack()

{

node temp = this.son;

while (temp != null)

{

Console.WriteLine(" | " + temp.veri + " | ");

temp = temp.geri;

}

Console.WriteLine();

}

public int size()

{

int count = 0;

node temp = this.bas;

while (temp != null)

{

count++;

temp = temp.ileri;

}

return count;

}

public char peek()

{

if (this.size() > 0)

return this.son.veri;

return '0';

}

public void addtohead(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.ileri = this.bas;

this.bas.geri = node;

this.bas = node;

}

}

public void addtoend(node node)

{

if (bas == null)

{

this.bas = node;

this.son = node;

}

else

{

node.geri = this.son;

this.son.ileri = node;

this.son = node;

}

}

public void addtomiddle(node node)

{

if (this.size() > 2)

{

node temp = this.bas;

for (int i = 0; i < this.size() / 2 - 1; i++)

{

temp = temp.ileri;

}

node.geri = temp;

node.ileri = temp.ileri;

temp.ileri = node;

}

else

{

Console.WriteLine("Ortaya eklemek için listenin uzunluğu en az 3 olmalıdır.");

}

}

public void removefrommiddle()

{

if (this.size() > 2)

{

node temp = this.bas;

for (int i = 0; i < this.size() / 2 - 1; i++)

{

temp = temp.ileri;

}

temp.ileri.ileri.geri = temp;

temp.ileri = temp.ileri.ileri;

}

else

{

Console.WriteLine("Ortadan silmek için listenin uzunluğu en az 3 olmalıdır.");

}

}

public void removefromhead()

{

if (this.size() > 0)

{

this.bas = this.bas.ileri;

this.bas.geri = null;

}

else

{

Console.WriteLine("Listede silmek için eleman yok.");

}

}

public void removefromend()

{

if (this.size() > 0)

{

this.son = this.son.geri;

this.son.ileri = null;

}

else

{

Console.WriteLine("Listede silmek için eleman yok.");

}

}

}

public static bool isPalindrome(string str)

{

stack s = new stack();

queue q = new queue();

string str1 = str;

//Verilen stringi küçük harflere çeviriyorum

//Verilen stringdeki boşlukları kapatıyorum

str = str.ToLower();

str = str.Replace(" ", "");

//Hem kuyruğa hem de stacke harflere atıyorum

for (int i = 0; i < str.Length; i++)

{

q.enqueue(new node(str[i]));

s.push(new node(str[i]));

}

//Tek tek harfleri stackten ve kuyruktan çıkartıp aynı mı diye bakıyorum

for (int i = 0;i< str.Length; i++)

{

if (q.dequeue() != s.pop())

{

Console.WriteLine("'" + str1 + "'" + " isnt palindrome.");

return false;

}

}

Console.WriteLine("'"+str1+"'"+" is palindrome.");

return true;

}

static void Main()

{

queue queue = new queue();

isPalindrome("ey edip adaNadA PiDe ye");

isPalindrome("AdDa");

isPalindrome("MessSi");

}

}

}

**Kod (Dizi kullanarak) :**

using System;

namespace Prog

{

class Program

{

class stack

{

private char[] ele;

private int top;

private int max;

public stack(int size)

{

ele = new char[size];

top = -1;

max = size;

}

public void push(char item)

{

if (top == max - 1)

{

Console.WriteLine("stack dolu.");

return;

}

else

{

ele[++top] = item;

}

}

public int pop()

{

if (top == -1)

{

Console.WriteLine("stack bos.");

return -1;

}

else

{

return ele[top--];

}

}

public void printStack()

{

if (top == -1)

{

Console.WriteLine("stack bos.");

return;

}

else

{

for (int i = top; i > -1; i--)

{

Console.WriteLine(" | " + ele[i] + " | ");

}

Console.WriteLine();

}

}

public char peek()

{

if (top != -1)

return ele[top];

return '1';

}

public char first()

{

return ele[0];

}

public char last()

{

return ele[top];

}

public int size()

{

return top;

}

}

class queue

{

public char[] ele;

private int top;

private int max;

public queue(int size)

{

ele = new char[size];

top = -1;

max = size;

}

public void enqueue(char value)

{

if (top == max - 1)

{

Console.WriteLine("stack dolu.");

return;

}

else

{

ele[++top] = value;

}

}

public char dequeue() {

if (top == -1)

{

Console.WriteLine("stack bos.");

return '0';

}

else

{

char[] arr = this.ele;

char x = arr[0];

for (int i = 1; i < this.max; i++)

{

arr[i - 1] = arr[i];

}

for (int i = 0; i < arr.Length; i++)

{

ele[i] = arr[i];

}

this.top--;

return x;

}

}

public void printQueue()

{

if (top == -1)

{

Console.WriteLine("stack bos.");

return;

}

else

{

for (int i = top; i > -1; i--)

{

Console.Write(ele[i] + " -> ");

}

Console.WriteLine();

}

}

}

public static bool isPalindrome(string str)

{

stack s = new stack(str.Length);

queue q = new queue(str.Length);

string str1 = str;

//Verilen stringi küçük harflere çeviriyorum

//Verilen stringdeki boşlukları kapatıyorum

str = str.ToLower();

str = str.Replace(" ", "");

//Hem kuyruğa hem de stacke harflere atıyorum

for (int i = 0; i < str.Length; i++)

{

q.enqueue(str[i]);

s.push(str[i]);

}

//Tek tek harfleri stackten ve kuyruktan çıkartıp aynı mı diye bakıyorum

for (int i = 0; i < str.Length; i++)

{

if (q.dequeue() != s.pop())

{

Console.WriteLine("'" + str1 + "'" + " isnt palindrome.");

return false;

}

}

Console.WriteLine("'" + str1 + "'" + " is palindrome.");

return true;

}

static void Main()

{

isPalindrome("AMdDma");

isPalindrome("ey edip aDanADa Pide ye");

}

}

}