

# YZM 2116

## Veri Yapıları

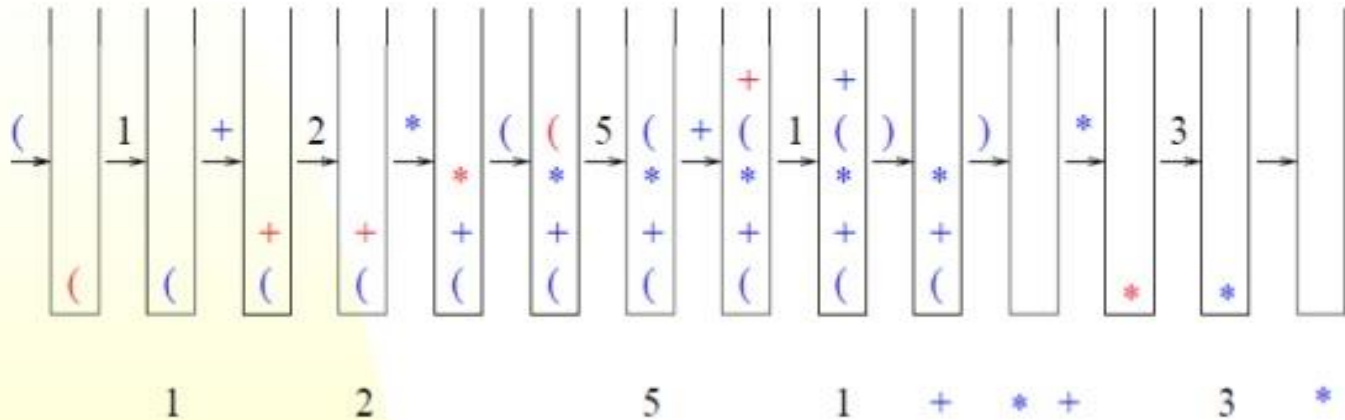
Yrd. Doç. Dr. Deniz KILINÇ  
Celal Bayar Üniversitesi  
Hasan Ferdi Turgutlu Teknoloji Fakültesi  
Yazılım Mühendisliği

# Infix to Postfix

## Stack Kullanarak Infix ifadeleri Postfix'e çevirme

### Infix to Postfix Expressions

- Example :  $(1+2*(5+1))*3$



postfix expression  $\rightarrow 1 2 5 1 + * + 3 *$

# Infix to Postfix Örnek 1

---

## Stack Kullanarak Infix ifadeleri Postfix'e çevirme

Infix:  $A-(B+C*D)/E$

Postfix:?

$ABCD^*+E/-$

# Infix to Postfix Örnek 2

---

## Stack Kullanarak Infix ifadeleri Postfix'e çevirme

Infix:  $((A+B)*(C-E))/(F+G)$

Postfix:?

$AB+CE-*FG+ /$

# Infix to Postfix

---

## Stack Kullanarak Infix ifadeleri Postfix'e çevirme

[http://scanftree.com/Data\\_Structure/prefix-postfix-infix-online-converter](http://scanftree.com/Data_Structure/prefix-postfix-infix-online-converter) linkini kullanarak cevaplarınızı kontrol edebilirsiniz.



# Stack Implementasyonu

---

Stack projesini indirip  
çalıştırın....

<http://www.kisa.link/FxH>

# Lab Uygulaması 1

---

## Stack Kullanarak String'i Tersten Yazma (Reverse)

`ArrayTypedStack` veri yapısını kullanarak verilen bir string ifadeyi ters çevirecek algoritmayı geliştirelim.

String: merhaba

ReverseString: abahrem

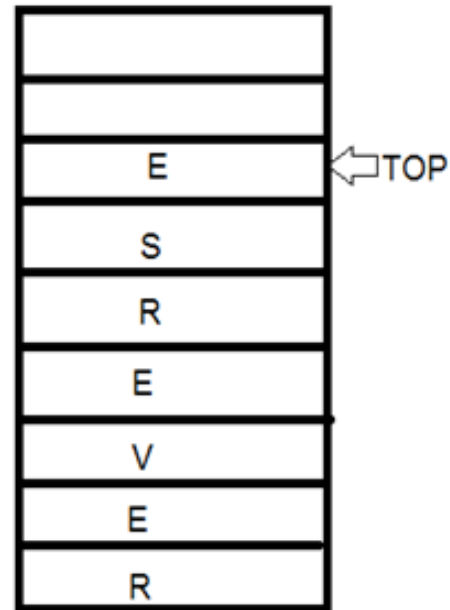
### Adımlar:

- String ifadedeki her bir karakter stack'e Push metodu ile aktarılır.
- Stack'deki her bir karakter Pop'ile stack'den okunur.

# Lab Uygulaması 1 (devam...)

STRING IS:

REVERSE



STACK



# Lab Uygulaması 1 (devam...)

---

```
private void btnReverseString_Click(object sender, EventArgs e)
{
    string test = txtOriginal.Text;

    char[] chrArr = test.ToCharArray();
    ArrayTypedStack ats = new ArrayTypedStack(chrArr.Length);

    for (int i = 0; i < chrArr.Length; i++)
    {
        ats.Push(chrArr[i]);
    }

    test = "";
    for (int i = 0; i < chrArr.Length; i++)
    {
        test += ats.Pop().ToString();
    }
    txtReverse.Text = test;
}
```

# Lab Uygulaması 2

---

## Parantez Eşleştirme Kontrolü (Matching Bracket Kontrol)

`ArrayTypedStack` veri yapısını kullanarak iç içe parantezler içeren bir ifadenin parantez eşleştirme kontrolünü gerçekleştiren algoritmayı geliştirelim.

# Lab Uygulaması 2

---

## Kurallar:

- Açılan ve kapanan toplam parantez sayısı eşit olmalıdır. Aç“(“ve kapa “)” parantezlerin eşitliğine bakılır.
- Kapanan her parantezden önce bir parantez açılmış olmalıdır. Her “)” için bir “(“olup olmadığına bakılır.

“((A+B)” ve “A+B(“ →1. şarta uymaz.

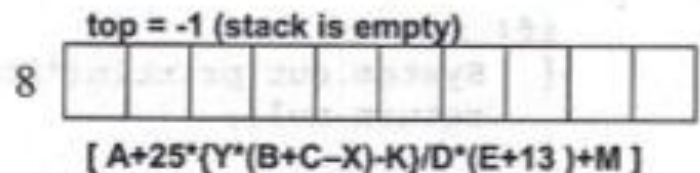
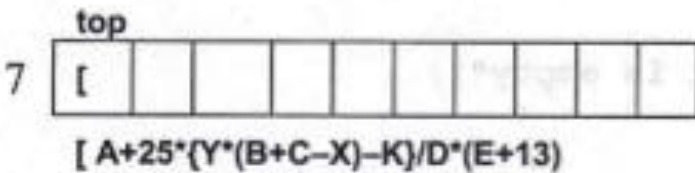
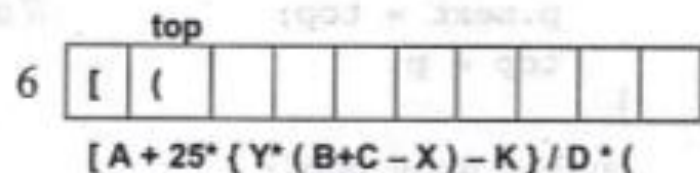
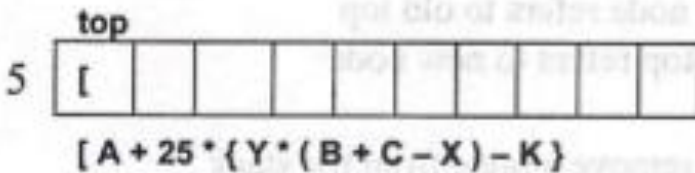
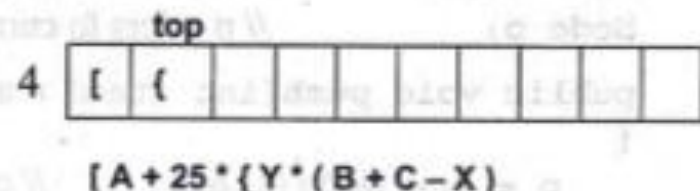
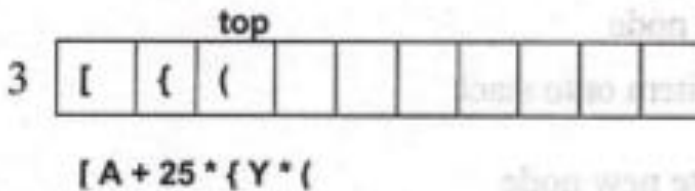
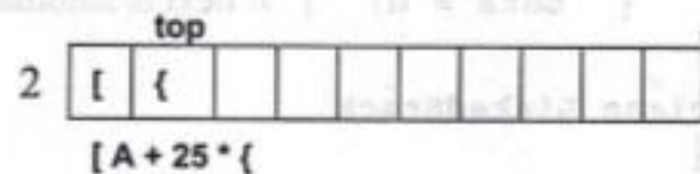
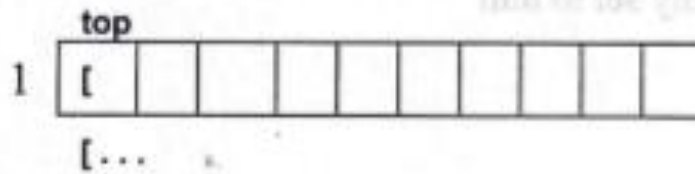
“)A+B(-C” ve “(A+B))- (C+D)” →2. şarta uymaz.

# Lab Uygulaması 2<sub>(devam...)</sub>

- Algoritma?  $[A+25*\{Y*(B+C-X)-K\}/D*(E+13)+M]$ 
  - Bir açılış parantezi ile **karşılaştığında** “(”, “{”, “[” stack’e **Push** edilir.
  - Bir kapanış parantezi ile **karşılaştığında** “)”, “}”, “]” stack’e bakılır, **stack boş değilse** stack’ten bir eleman **Pop** edilerek, *doğru karşılık olup olmadığı kontrol edilir.*
    - ❑ **Doğruysa** işlem sürdürülür.
    - ❑ **Doğru değilse** ifade geçersizdir.
  - Stack sonuna ulaşıldığında **stack boş olmalıdır.**
    - ❑ Aksi halde **açılmış** ama **kapanmamış** parantez olabilir.

# Lab Uygulaması 2 (devam...)

**Test:**  $[A+25*\{Y*(B+C-X)-K\}/D*(E+13)+M]$



# Lab Uygulaması 3

---

## Stack'e nesne ekleme

`ArrayTypedStack` veri yapısını kullanarak `Musteri` nesnesini stack'e ekleme işlemini gerçekleştirelim.

Musteri sınıfı oluşturunuz.

- Ad
- Soyad
- Yas
- Telefon

# İYİ ÇALIŞMALAR...