

2013-2014 Bahar Yarıyılı
Veri yapıları ve Algoritmalar Dersi 3. Ödevi

Konu : Yığın Veri Yapısı Kullanarak Aritmetik İfade Çözümü

Problem: Bu ödevde, verilen bir aritmetik işlemin sonucunu **yığın(stack)** veri yapısını kullanarak hesaplayan bir sistem tasarlanacaktır.

Ödev 4 ana bölümden oluşmaktadır :

1. Yığın işlemlerine ait fonksiyonların yazılması:

Yığına eleman **push** etme ve yığından eleman **pop** etme için gerekli fonksiyonları yazınız.

2. Infix – Postfix Dönüşümü :

Infix formunda verilen aritmetik ifadeyi, **aritmetik operatörler için yığın kullanarak** postfix formuna çeviren fonksiyonu yazınız.

İşlem Adımları :

1. Aritmetik işlemi soldan sağa doğru değerlendiriniz.

Eğer o anda bakılan karakter:

- **sol parantez ise :** sol parantez yığına **push** edilir.
 - **sağ parantez ise :** sol parantez çıkana kadar yığından **pop** işlemi yapılır. Alınan işlem işareti postfix ifadeye eklenir. Sol parantez görüldüğünde pop işlemine son verilir. Sol parantez postfix'e eklenmez.
 - **sayı ise :** postfix ifadeye eklenir.
 - **işlem işareti ise :** yığının en üstünde sol parantez varsa veya en üstteki işaretin önceliği bu işareten düşük ise işlem işareti yığına **push** edilir. Bu işaretin önceliği daha düşük ise yığındaki bu işareten yüksek öncelikli işaretler için **pop** işlemi yapılır. Stackten pop edilen işaretler postfix ifadeye eklenir. İşlem işareti yığına **push** edilir.
 - **ifadeler bittiğinde :** yığındaki işaretler sıra ile **pop** edilerek postfix ifadeye eklenir.
2. Okuma yaparken girişteki **sayılar ve işlem işaretleri arasında bir boşluk** bırakınız.
 3. Postfix ifadeyi oluştururken **her ifade arasında bir boşluk** bırakınız.
 4. Giriş bilgisini string olarak okuyunuz.
 5. Giriş stringindeki sayı olan karakterlerin sayı karşılığını elde etmek için bir fonksiyon yazınız. Aritmetik işlemlerde ve işlem sonuçlarında **9'dan büyük sayılar olmayacağı** varsayılmaktadır.
 6. Kullanılan işlem işaretleri şunlar olabilir : **+ - * / ()**
 7. Yanlış giriş yapılmadığı varsayılmaktadır.

Örnek:

Giriş : 8 / 4 * (3 + 1)

İşlemler: 8 sayı olduğu için postfix'e eklenir:

Postfix : 8 Yığın : Boş

/ yığın boş olduğu için yığına push edilir:

Postfix : 8 Yığın : /

4 sayı olduğu için postfix'e eklenir:

Postfix : 8 4 Yığın : /

* yığının en üstündeki ifadenin işlem önceliği ile karşılaştırılır. Önceliği aynı olduğu için / işareti yığından pop edilerek postfix'e eklenir, * işareti yığına push edilir:

Postfix : 8 4 / Yığın : *

(yığına push edilir:

Postfix : 8 4 / Yığın : * (

3 sayı olduğu için postfix'e eklenir:

Postfix : 8 4 / 3 Yığın : * (

+ yığının en üstünde (olduğu için yığına push edilir:

Postfix : 8 4 / 3 Yığın : * (+

1 sayı olduğu için postfix'e eklenir:

Postfix : 8 4 / 3 1 Yığın : * (+

) geldiği için ('e kadar olan işaretler yığından pop edilip postfix'e eklenir. (yığından pop edilir:

Postfix : 8 4 /3 1 + * Yığın : Boş

Giriş string'i bittiği için yığında kalan elemanlar sıra ile çekilerek postfix'e eklenmelidir. Fakat yığın boş olduğu için bu işlem yapılmaz.

Ara Çıkış : 8 4 /3 1 + *

3. Postfix İfadenin Çözülmesi :

İlk aşamanın sonunda elde edilen postfix ifadeyi **sayılar için yığın veri yapısı kullanarak** çözen işlemi yapan fonksiyonu yazınız.

İşlem Adımları :

Postfix ifade soldan sağa doğru değerlendirilir.

Eğer o anda bakılan karakter:

- **sayı ise** : sayı yığına **push** edilir.
- **işlem işareti ise** : yığının üstündeki iki değer **pop** edilerek aralarında bu işlem yapılır. İşlem sonucu yığının en üstüne yerleştirilir.

Örnek:

Ara Çıkış : 8 4 /3 1 + * için aritmetik ifade aşağıdaki gibi çözülür:

İşlemler: 8 sayı olduğu için yığına push edilir:

Yığın : 8

4 yığına push edilir:

Yığın : 8 4

/ işlem işareti olduğu için yığının en üstündeki iki sayı pop edilir. Aralarında bölme işlemi yapılır (8/4).

İşlem sonucu yığına push edilir.

Yığın : 2

2 yığına push edilir:

Yığın : 2 3

1 yığına push edilir:

Yığın : 2 3 1

+ işlem işareti olduğu için yığının en üstündeki iki sayı pop edilir. Aralarında toplama yapılır (3+1).

İşlem sonucu yığına push edilir.

Yığın : 2 4

* işlem işareti olduğu için yığının en üstündeki iki sayı pop edilir. Aralarında çarpma yapılır (2*4).

İşlem sonucu yığına push edilir.

Yığın : 8

Postfix ifade bittiği için yığındaki sayı pop edilir. Bu sayı işlem sonucunu verir:

Çıkış : 8

4. Aritmetik işlemin yapılması :

İşlem işaretleri karakter tipinde verilmektedir. İşlem işareti için uygun aritmetik işlemi yaparak sonucunu döndüren fonksiyonu yazınız.

Algoritmanızın programını C dilinde yazınız.

Teslim İşlemleri:

Ödevler **5 Mayıs 2014** haftası yapılacak laboratuvarda gösterilecektir. Laboratuvar planını lab. sorumlusu asistanların sayfasından takip ediniz.

Ödev raporunuzu ve program kodunu laboratuvarda programınız çalıştırarak yapacağınız sunumdan sonra kontrolü yapan araştırma görevlisi tarafından verilecek USB belleğe kopyalanacaktır.

- **ÖğrenciNumarası_Ad_Soyad** olan bir **klasör** hazırlayınız.
- Bu klasör içinde ayrıca **rapor** ve **kod** klasörleri olmalıdır.
- Rapor, **ÖğrenciNumarası_Rapor.pdf** şeklinde adlandırılmalıdır.
- Kod klasöründe **kaynak kod** yer almalıdır.

Laboratuvar Sunumu: Laboratuvar sunumunda size verilecek olan bir aritmetik ifadenin **postfix sonucunu ve işlem sonucunu** göstermeniz istenecektir.

Değerlendirme:Ödeviniz aşağıdaki gibi değerlendirilecektir:

Algoritma Tasarımı ve Programın Çalışması: (%80)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Programda gerekli alt modüller belirlenerek her modül ayrı fonksiyon olarak yazılmalıdır.
4. Program hatasız çalışmalıdır.
5. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%20)

1. Raporun kapak sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Yöntem, uygulama ve sonuç bölümlerinde, yukarıda açıklaması verilen bilgiler anlatılmalıdır.
3. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
4. Değişken ve fonksiyon(veya metod) isimleri anlamlı olmalıdır.
5. Her fonksiyonun (veya metodun) yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
6. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
7. Gereksiz kod tekrarı olmamalıdır.
8. Kaynak kodun formatı düzgün olmalıdır.