**BSM-438 DERLEYICI TASARIMI FİNAL ÖDEVİ**

Aşağıda açıklanan final ödevi, birbiri ile bağlantılı toplam 4 adet sorudan oluşmaktadır. Her soru, bir önceki soruda geliştirilen çözümün sonuçlarını kullanmaktadır. Ödevde kullanılacak araçlar Unix/Linux ortamında Lex ve Yacc ve C programlama dili olarak belirlenmiştir. Geliştirdiğiniz programlar, dışarıdan (terminalden) “.c” uzantılı bir C programını girdi olarak alarak işlem yapmalıdır.

Tablo 1 Tanınacak olan Token listesi

|  |  |  |  |
| --- | --- | --- | --- |
| Ayrıştırıcılar | Operatorler | Anahtar kelimeler (Keyword) | Tanımlayıcılar (Identiﬁer): |
| Noktali\_virgul (“;”) | **Atama** (“=”) | **IF** (“if”) | **ID**: C dilinde tanımlanabilen tanımlayıcılar (derste yapılan ID tanımları gibi) |
| Parantez\_ac (“(“) | **Kucuk** (“<”) | **ELSE** (“else”) | **Yorum:** “/\*” ve “\*/” sembolleri arasındaki ifadeler. |
| Parantez\_kapa (“)”) | **Buyuk** (“>”) | **WHILE** (“while”) |  |
| Virgul (“,”) | **Esit** (“==”) | **INT** (“int”) |  |
| SParantez\_ac (“{”) | **Arti** (“+”) | **FLOAT** (“ﬂoat”) |  |
| SParantez\_kapa (“}”) | **Eksi** (“-”) |  |  |
| Kparantez\_ac (“[”) | **Carpma** (“\*”) |  |  |
| Kparantez\_kapa (“]”) | **Bolme** (“/”) |  |  |
|  | **Degil** (“!”) |  |  |
|  | **Ve** (“&&”) |  |  |
|  | **Veya** (“||”) |  |  |

**Soru 1)** C dilindeki herhangi bir kod parçasını tarayan bir tarayıcı (scanner) program geliştiriniz. Taranan “token”lar, sembol tablosu olarak terminalde yazdırılarak kullanıcıya sunulmalıdır. Taranacak olan “token” listesi Tablo 1’de verilmektedir.

**Soru 2)** Tarayıcı tarafından belirlenen Tablo 1’deki “token”ları ayrıştıran bir ayrıştırıcı (parser) program geliştiriniz.

Soru 1 ve 2 tamamlandığında oluşan programınız, değişken tanımlamalarını (int a=0; şeklindeki başlangıç değeri atamalarını desteklemesine gerek yoktur.), operatör içeren ifadeleri, parantezler, if-else ve while deyimlerini tanımalıdır.

**Soru 3)** Ayrıştırıcı programınızı Abstract Syntax Tree oluşturacak şekilde geliştiriniz. AST sonucu dersteki örneklerdeki gibi tablo şeklinde terminale yazdırılmalıdır.

**Soru 4)** Programınız artık AST ağacını çıktı olarak vermekte ve bu ağacı öğrencino\_soru3.txt şeklinde kaydedebilmelidir. Şimdi bu programı makine koduna dönüşüm yapacak şekilde geliştiriniz. Bunun için C dilini tercih ettiyseniz;

Soru 3’teki programı batch olarak çağırıp öğrencino\_soru3.txt çıktı dosyasını üreten, ardından bu txt’yi okuyup makine kodu üreten ve öğrencino\_soru4.txt şeklinde kaydeden bir C programı geliştiriniz.

Programlarınızı test etmek için aşağıdaki kod parçasını test.c isimli bir dosyaya kaydedip deneyebilirsiniz. Bu kod geçerli bir program parçasıdır.

float a[100][100], b[100][100], c[100][100];

int i, j, k;

i = 0;

while (i < 100) {

j = 0;

while (j < 100) {

if (!(c[i][j] == 0.0))

c[i][j] = 0.0;

k = 0;

while (k < 100) {

c[i][j] = c[i][j] + a[i][k] \* b[k][j];

k = k + 1;

}

j = j + 1;

}

i = i + 1;

}

Hatırlanması gereken bazı hususlar:

1. Programlar yazılırken mutlaka koyu yazıyla belirtilen değişken isimleri kullanılmalıdır.
2. Ödevler arasındaki benzerlik yüzdesine bakılacak olup, kopya ödevler değerlendirmeye alınmayacaktır.
3. Her sorunun değerlendirilmesi için doğru/yanlış olarak yazılmış C kod parçaları denenecektir. Dolayısıyla her soruda geliştirilen programlar ayrı ayrı gönderilmeli, toplamda 4 adet klasör isimleri soru1, soru2, soru3, ve soru4 olacak şekilde teslim edilmelidir.
4. Geliştirilen 4 programın **her biri** için MUTLAKA “a.out”, “lex.yy.c”, “y.tab.c”, .l uzantılı lex dosyası, .y uzantılı yacc dosyası, kullanıldı ise kütüphane dosyaları, çıktı \*.txt dosyaları ve son soru C dilinde yapıldı ise bu c dosyası gönderilmelidir.
5. Her soruda geliştireceğiniz program, yaptığı işlemin sonucunu öğrencino\_soru1.txt, öğrencino\_soru2.txt, öğrencino\_soru3.txt ve öğrencino\_soru4.txt şeklinde kaydetmelidir.
6. Her soru bir önceki cevabın geliştirilmesiyle yapıldığından, örneğin 4. sorudaki programınız çalıştığında tarayıcı, ayrıştırıcı, ast oluşturup, en son makine kodu üretmelidir. Benzer şekilde 3. sorudaki program da tarayıcı ve ayrıştırıcı oluşturup ardından ast ağacını üretmelidir.

Başarılar Dileriz

Prof. Dr. Nejat Yumuşak

Dr. Kevser Ovaz Akpınar