

FORTRAN Değişkenleri

- FORTRAN'daki değişken terimi, programın çeşitli yerlerinde veya safhalarında değişik değerler alabilen niceliklerin sembolik adı olup en fazla 6 karakterden oluşur (standart FORTRAN).
- İlk karakter mutlaka bir harf olmalı ve sadece alfabetik veya sayısal karakterlerden oluşmalıdır.
- Bir FORTRAN derleyicisi, bir değişken veya sabit isminin ilk harfini kullanarak (aksi belirtilmedikçe), değişkenin gerçel mi yoksa tam sayı mı olduğunu belirler.

VERİ TİPLERİ ve ARİTMETİK DEYİMLER

- TAM SAYI (kesirli kısmı olmayan),
- GERÇEL SAYI (kesirli kısmı olan),
- KOMPLEKS SAYI (sanal kısmı olan),
- KARAKTER (alfabetik)

Değişkenlerin isimlendirilmesi

- Maksimum 32 karakter
- Alfabetik karakter ile başlamalı
- Kullanılacak karakterler:
 - Harf olarak (A - Z, a - z)
 - Sayı olarak (0-9)
 - Boşluk

Değişkenlerin isimleri

➤ Örnekler:

doğru

AgirlikDegiskeni

HARDDISK

pesin_para

yanlış

Ornek Agirligi

Benim@bilgisayarım

TL olarak

Tam Sayı Değişkenleri

- Ondalık noktası taşımayan, bölündüklerinde yine ondalık hanesi olmayan pozitif ve negatif sayılardır.
- Bilgisayarın kabul edebileceği tam sayının büyüklüğü, işletim sistemine, derleyiciye ve bilgisayara bağlı olarak değişebilir.
-
- Örneğin IBM 360 veya F77L3 Lahey derleyicisi için tam sayı sabitindeki hane sayısı 10 iken, CDC 6000' de 18 'dir.
- Genellikle tamsayı değişkenleri için kullanılacak maksimum değeri 32 bitlik bir sayı değeri $2^{31}-1=2.147.483.647$ dir.

Tam Sayı Değişkenleri

- TAM SAYI DEĞİŞKENLERİNİN ilk harfi I, J , K , L , M veya N ile başlar

Örnek :

MATRIS, K25, I6M2P, I, J

Tam Sayı Değişkenleri

- Tamsayıların yanlış kullanımı bilgisayarın veya programın hatalı bir şekilde durmasına neden olur (overflow error).
- Tamsayıların kullanılmadığı yerlerde yani 0.25, 1.4 veya -3.2 gibi sayılar için gerçel sayılar (floating-point data type) tanımlanır. Bu ifade, Fortran da kullanılacak olan tamsayı değişkenlerin tanımlandığı satırdır.

Tam Sayı Değişkenleri

- Tamsayı değişken tanımlama aşamasında ilk atama işlemi yapılabilir. Aşağıda Fortran 90/95 de tamsayı tanımlanması yapılmaktadır:

INTEGER :: degisken adi[, degisken adi, vs.]

INTEGER :: i, j, sayac

INTEGER :: gun = 4


```
PROGRAM Bolu8_01
IMPLICIT NONEINTEGER :: I
REAL :: r
CHARACTER (1) :: c
LOGICAL :: l
COMPLEX :: cp
PRINT *, ' Tamsayı ', KIND(i)
PRINT *, ' Gerçel ', KIND(r)
PRINT *, ' Karakter ', KIND(c)
PRINT *, ' Mantiksal ', KIND(l)
PRINT *, ' Karmaşık ', KIND(cp)
END PROGRAM Bolu8_01
```

INTEGER*1 değişkeni bellekte 1 byte lık yer işgal eder ($n=1 \times 8=8$ bit)
en küçük tamsayı $= -2^{8-1} = -128$
en büyük tamsayı $= 2^{8-1} - 1 = 127$

7

6

0



INTEGER*2 değişkeni bellekte 2 byte lık yer işgal eder ($n=2 \times 8=16$ bit).

en küçük tamsayı $= -2^{16-1} = -32.768$

en büyük tamsayı $= 2^{16-1} - 1 = 32.767$

15

14

0



INTEGER*4 değişkeni bellekte 4 byte lık yer işgal eder ($n=4 \times 8=32$ bit).

en küçük tamsayı $= -2^{32-1} = -2.147.483.648$

en büyük tamsayı $= 2^{32-1} - 1 = 2.147.483.647$

31

30

0



Gerçel Sayı Değişkenleri

- ondalık noktası taşıyan rasyonel sayılardır.
- **127.56 \Rightarrow 0.12756E+03,**
- **3.6 x 10⁻⁸ \Rightarrow 0.36E-07**

GERÇEL SAYI DEĞİŞKENLERİ

- İlk harf, tam sayı değişkenini belirten harflerden başka bir harfle (A - H , O - Z) başlar.
- Örnek : AMAT, RK25, ZI6M2P, TOPLAM,
DIZI, ENERJİ, FOTON, YUKSEK,
HIZ, ABC, BAC, A, AB, BA

REAL :: degisken adi[, degisken adi, vs.]

REAL :: degisken adi = deger

REAL :: uzaklik, zaman

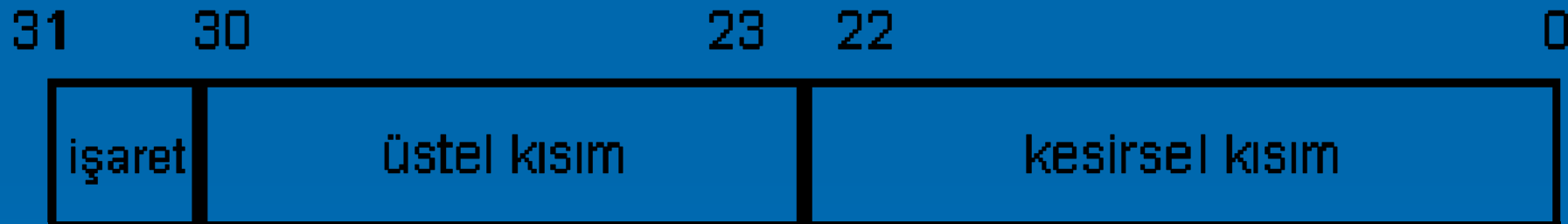
REAL :: uzaklik = 100

GERÇEL SAYI DEĞİŞKENLERİ

REAL*4 gerçel sayı değişkeni bellekte 4 byte lık yer işgal eder ve üç kısım olarak saklanırlar.

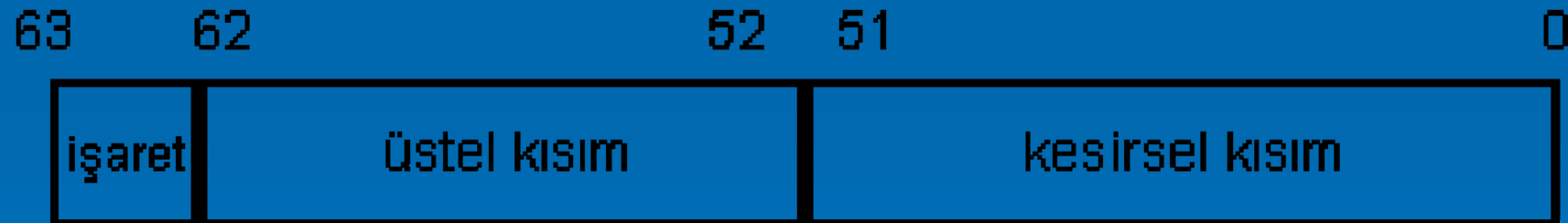
$-1.17549435 \times 10^{-38}$

$3.40282347 \times 10^{-38}$



GERÇEL SAYI DEĞİŞKENLERİ

REAL*8 gerçel sayı değişkeni bellekte
8 byte lık yer işgal eder.
-1.17549435E-38 ile 3.40282347E38



Mantıksal Değişkenler

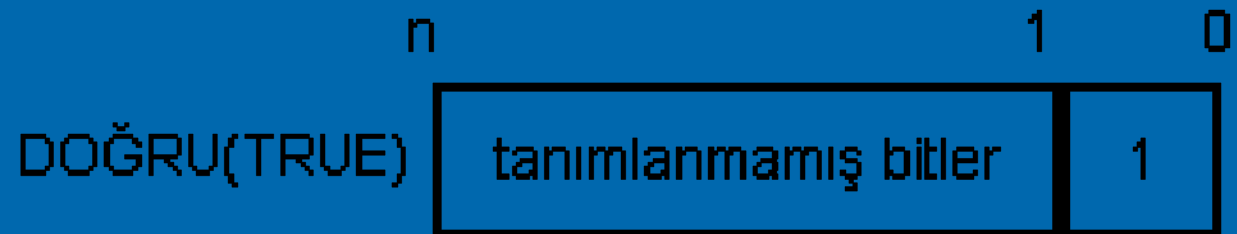
- Mantıksal değişkenlerin değerleri tamsayıların değerleri gibi işlem görür.
- LOGICAL*1 A, A değişkeni veya sabiti bellekte 1 bytelık yer işgal eder ve $-128 \leq A \leq 127$ değerlerini alır.
- LOGICAL*2 A, A değişkeni veya sabiti bellekte 2 bytelık yer işgal eder ve $-32768 \leq A \leq 32767$ değerlerini alır.
- LOGICAL*4 A, A değişkeni veya sabiti bellekte 4 bytelık yer işgal eder ve $-2,147,483,648 \leq A \leq 2,147,483,647$ değerlerini alır.

LOGICAL :: değişken adi[, değişken adi vs.]

LOGICAL :: başlangıç, kontrol

LOGICAL :: kontrol = .false.

Mantıksal Değişkenler



n LOGICAL tanımındaki miktara göre 7, 15, 31 değerlerini alır.

Veri Tiplerinin Bellek Gereksinimi

➤ Tipi	Tür	Boyutu	Bilgi
➤ BYTE	1	1	INTEGER(1) gibi.
➤ INTEGER	4	4	INTEGER(4) gibi.
➤ INTEGER(1)	1	1	
➤ INTEGER(2)	2	2	
➤ INTEGER(4)	4	4	
➤ REAL	4	4	REAL(4) gibi.
➤ REAL(4)	4	4	
➤ DOUBLE PRECISION	8	8	REAL(8) gibi.
➤ REAL(8)	8	8	
➤ COMPLEX	4	8	COMPLEX(4).
➤ COMPLEX(4)	4	8	
➤ DOUBLE COMPLEX	8	16	COMPLEX(8).
➤ COMPLEX(8)	8	16	
➤ CHARACTER	1	1	CHARACTER ve CHARACTER(1).
➤ CHARACTER*n	1	<i>n</i>	<i>n</i> karakter katarının uzunluğu.
➤ LOGICAL	4	4	LOGICAL(4) gibi.
➤ LOGICAL(1)	1	1	
➤ LOGICAL(2)	2	2	
➤ LOGICAL(4)	4	4	

Karakter Değişkenleri

- Her karakter bellekte 1 byte yer işgal eder.
- Karakter uzunluğu 1 den 65535 e kadar değişebilir.
- CHARACTER(len=<uzunluk>) :: degisken adi[, degisken adi]

CHARACTER(<uzunluk>) :: degisken adi[, degisken adi]

CHARACTER :: deęişken adi[, deęişken adi]

CHARACTER*<uzunluk> :: deęişken adi

CHARACTER deęişken adi*<len>

Karakterler

```
PROGRAM Bolum7_A01
```

```
! Fortran90/95 : bu program bir isim girilmesini ve
```

```
! ismin ekrana yazdırılmasını sağlar
```

```
IMPLICIT NONE CHARACTER (20) :: Ad
```

```
PRINT *, 'Adınızı giriniz.'
```

```
PRINT *, '20 karaktere kadar'
```

```
READ *, Ad
```

```
PRINT *, Ad
```

```
END PROGRAM
```

Kompleks Sayı Sabitleri

- $a \pm i \cdot b$ yapısında olan kompleks sayılar FORTRAN 'da $(a , \pm b)$ formunda temsil edilir.
- a , hem de b gerçel sayı olmalıdır.
- a ve b nin her ikisi de aynı duyarlılıkta olmalıdır.

veri tipleri ve deęişken tanımlamaları

1. Bir deęişken isminin ilk karakteri alfabetik olmalıdır.
2. Bir deęişkende yer alan karakter sayısı altıyı geçmez.
3. İlk karakterden sonraki karakterler sayısal veya alfasayısal olabilir.
4. Özel karakterler bir deęişken içinde kullanılamaz. Bu özel karakterler \$, #, @, *, (,),+,=,/,<,> vb olabilir.
5. Özel anlamlı bazı isimler deęişken adı olarak kullanılamaz.
6. I,J,K,L,M,N harflerinden biriyle başlayan deęişkenler tamsayı, dięer harflerle başlayanlar ise kesirli sayı deęişkenleridir.
7. Pozitif ve negatif **tam sayılar** bellekte 1, 2, 4 veya 8 byte kadar yer işgal ederler.

veri tipleri ve deęişken tanımlamaları

8. 7. n bit sayısı olmak üzere: Enküçük tamsayı $=-2^{n-1}$ ve enbüyük tamsayı $=2^{n-1}-1$
9. 4 byte lık (n=32 bit) tamsayı için, enküçük tamsayı -2.147.483.648 enbüyük tamsayı 2.147.483.647 dir.
10. 1 bit 2 li sayı tabanına göre 0 veya 1 deęerlerinden biridir. Bu bitler birleřtirilerek byte lar oluşur.
11. İkili sayı sisteminde (genel olarak) en küçük digit grubu 8 bit ten oluşur ve 8 bit 1 byte tır.
12. Bit lerin oluşturduęu byte tan daha büyük gruplara **word** denir.
13. Tamsayıların yanlış kullanımı bilgisayarın veya programın hatalı bir şekilde durmasına neden olur (overflow error).

veri tipleri ve deęişken tanımlamaları

14. Tamsayıların kullanılmadığı yerlerde yani 0.25, 1.4 veya -3.2 gibi sayılar için gerçel sayılar (floating-point data type) tanımlanır.
15. Pozitif ve negatif gerçel deęerler bellekte 4 byte (32 bit) yer işgal ederler.
16. 4 byte lık gerçel sayılar (32 bit) taban ve kuvvetinden/üstel terimden (mantissa and exponent) oluşur. 32 bit lik bir gerçel sayıda 24 bit (3 byte) taban (mantissa) ve 8 bit (1 byte) üstel kısma ayrılır.
17. 4 byte lık gerçel sayı = 3 byte lık taban x 1 byte lık üstel terim = $m \times 2^e$
18. Gerçel sayılar ikili sayı sistemine çevrilirken veya tersi durumunda yuvarlama hataları olmaktadır (Round off error). Örneğin 1.99999 bazen 2.00000 olarak karşımıza çıkabilir.
19. 1 karakter bellekte 1 byte yer işgal eder.
20. Karakterler ASCII (American standard Code for Information Interchange) veya EBCDIC (IBM) standartlarına göre sınıflandırılır. Japon ve Çin dillerinde 4000 civarında karakter vardır.
21. 256 deęişik deęer 256 byte deęerini oluşturur. Her karakter Unicode da, bellekte 2 byte yer işgal eder. Bu da, $(256^2)=65536$ olası karakter kodunun yazılmasını sağlar

Aritmetik İşlemler

İşlem	Sembol	Matematiksel	Örnek
Toplama	+	$a+b$	$A+B$
Çıkarma	-	$a-b$	$A-B$
Çarpma	*	ab	$A*B$
Bölme	/	a/b	A/B
Üs alma	**	a^b	$A**B$

Koşullu Mantıksal İşlemciler

Yeni	Eski	Anlamı
= =	.EQ.	Eşit
/=	.NE.	Eşit değil
>	.GT.	Büyük
>=	.GE.	Büyük eşit
<	.LT.	Küçük
<=	.LE.	Küçük eşit

FORTRAN'da aritmetik işlem kuralları

1- İki aritmetik işlem sembolü yanyana kullanılmaz: $A*-B$ yanlış doğru $\Rightarrow A * (-B)$

2- Bütün işlemlerde mutlaka aritmetik işlem sembolü kullanılmalıdır. Örneğin parantez yalnızca işlem gruplarını ayırır; çarpma anlamına gelmez.
 $2X<=$ yanlış doğru $\Rightarrow 2*X$ veya $(A+B)C$ yanlış doğru $\Rightarrow (A+B)*C$

3- Aritmetik semboller kullanılarak yazılan bir deyimde işlemlerin öncelik sıralaması aşağıdaki gibidir :

- i) Parantez (parantez içeren işlemler en içteki parantezlerden dışarıya doğru yapılır),
- ii) Matematiksel fonksiyon
- iii) ** (Üstel işlemler sağdan sola doğru yapılır)
- iv) * ve / (Çarpma ve bölme soldan sağa doğru yapılır)
- v) + ve - (Toplama ve çıkarma soldan sağa doğru yapılır)
- vi) Koşul işlemcileri (==, /=, >, >=, <, <=) soldan sağa doğru yapılır,
- vii) Bütün .NOT. işlemleri yapılır
- viii) Bütün .AND. işlemleri soldan sağa doğru yapılır,
- ix) Bütün .OR. işlemleri soldan sağa doğru yapılır,
- x) Bütün .EQ. ve .NE. işlemleri soldan sağa doğru yapılır,

FORTRAN'da aritmetik işlem kuralları

- 4- Açılan ve kapanan parantez sayısı eşit olmalıdır. Bu sadece aritmetik işlemlerde değil, diğer deyimlerde de geçerlidir.
- 5- Üs alma ifadelerinde işlem sırası, parantezlerle belirlenmelidir. Aynı öncelik sırasına sahip iki işlemde, önce soldaki yapılır. İç içe parantezler varsa, önce en içteki parantez işleme girer.
- 6- İşlemler yapılırken bütün değişken ve sabitler aynı tipten seçilirse sonuçlar da aynı tipten çıkar. Farklı olmaları halinde istenmeyen yanlış sonuçlar oluşabilir. Bu yüzden FORTRAN'da aritmetik işlemlerde çoğunlukla reel değişken ve sabitler kullanılır. Tam sayılar ise genellikle sayma, numaralama ve indisleri belirtmede kullanılır.
- 7- Karakterlerin birbirine eklenmesi // işlemcisi ile yapılır. 'Adı//' Soyadı' ekleme işleminden sonra 'Adı Soyadı' şeklini alır.

Fortran da İfadelerin Sırası

- PROGRAM ifade
- IMPLICIT NONE ifade
- Tip tanımlamaları REAL, INTEGER, LOGICAL ve CHARACTER ifadeler
- İşlem yapan/içeren ifadeler Atama ifadeleri, READ ifadeleri, WRITE ifadeleri, STOP ifadeleri
- END PROGRAM ifade

Fortran da komut sıralaması

1.	PROGRAM
2.	PARAMETER
3.	IMPLICIT
4.	EXPLICIT
5.	DIMENSION
6.	COMMON
7.	DATA
8.	Satır fonksiyonları
9.	Programın gövdesi
10.	END
11.	SUBROUTINE ve FUNCTION alt programları

Fortran da Atama İşlemi



➤ <degisken = ifade>

➤ Örnek:



```
pi=3.141593
```

```
uzaklik=0.5*ivme*zaman**2
```

```
kenar=hipotenus*cos(teta)
```


END PROGRAM İfadesi

- END PROGRAM ifadesi Fortran programının son ifadesi olmalıdır. Bu satır, derleyiciye bu satırdan sonra başka satırın olmayacağı belirtir. Programın çalışması bu satırla karşılaşıncaya kadar biter. Program adının yazılması ise isteğe bağlıdır.
- END PROGRAM [ad]

IMPLICIT NONE İfadesi

- Bu ifade FORTRAN daki standart yazım şeklini kapatır. Bir program içinde kullanılırsa, program içinde kullanılacak bütün değişkenler başlangıçta tanımlanmalıdır.
- **IMPLICIT NONE**

PROGRAM İfadesi

- PROGRAM ifadesi ile programa bir isim verilir. Bu programın ilk satırında olmalıdır. Programın adı 1 ile 31 tane alfabetik karakterden, sayıdan ve alt çizgiden oluşabilir. Fakat ilk değer karakter olmalıdır.
- PROGRAM program_adi
- PROGRAM ilk_programim

STOP İfadesi

- Bir Fortran programının çalışmasını durdurur. STOP ifadesi programın değişik yerlerinde olabilir. Eğer program içerisinde END ifadesine bir STOP ifadesinden önce erişiliyorsa STOP ifadesini kullanmaya gerek yoktur.
- STOP

Örnek program

```
PROGRAM Zaman
IMPLICIT NONERREAL :: I_dakika, Uzaklik, G_sure
INTEGER :: Dakika, Saniye
REAL , PARAMETER :: I_Yil = 9.46E12
! I_Yil : isigin 1 yilda Km olarak aldigi yol
! I_dakika: isigin 1 dakikada Km olarak aldigi yol
! Uzaklik : Gunesten dunyaya olan uzaklik
! G_sure : 1 dakikada alinan yol icin gecen sure
! Dakika : Gecen surenin dakikasinin tam kısmi
! Saniye : Gecen surenin saniyesinin tam kısmi
I_dakika = I_Yil/(365.25*24.0*60.0)
Uzaklik = 150.0*10**6
G_sure = Uzaklik / I_dakika
Dakika = G_sure
Saniye = (G_sure - Dakika)*60
Print *, ' Işık gunesten dunyaya ', Dakika, ' dakika'
Print *, ' ', Saniye, ' Saniye'
Print *, ' sonunda ulasir'
END PROGRAM Zaman
```

Fortran da hazır (intrinsic) fonksiyonlar ve seri açılımları

Matematiksel Fonksiyon	FORTRAN İsmi	işlem	Seri Açılımı
Exponansiyel	EXP (X)	e^x	
Doğal logaritma	LOG (X)	$\ln(x)$	
Doğal logaritma	ALOG (X)	$\ln(x)$	
Doğal logaritma	ALOG (1+X)	$\ln(1+x)$	
Logaritma	LOG10 (X)	$\log(x)$	10 tabanına göre logaritma
Logaritma	ALOG10 (X)	$\log(x)$	10 tabanına göre antilogaritma
Karekök	SQRT (X)	\sqrt{x}	a değeri başlangıç değeri, x karekökü alınacak sayı
Sinüs Sinüs ⁻¹	SIN (X) ASIN (X)	$\sin(x)$	
Kosünüs Kosünüs ⁻¹	COS (X) ACOS (X)	$\cos(x)$	
Hiperbolik tanjant	TANH (X)	$\tanh(x)$	
Karekök	SQRT (X)	\sqrt{x}	
Tan ⁻¹	ATAN (X)	Arctan x	
Mutlak değer	ABS (X)	$ x $	

Fortran da hazır (intrinsic) fonksiyonlar ve seri açılımları

Tip değiştirme	INT (X) IFIX (X) REAL (K) FLOAT (K)	tamsayı tamsayı gerçel gerçel	X-gerçel sayı tamsayıya döner X- gerçel sayı tamsayıya döner K-tamsayı gerçel sayıya döner K-tamsayı gerçel sayıya döner
Maksimum Minimum	MAX (X, Y, Z, ...) MAX0 (I, J, K, ...) MAX1 (X, Y, Z, ..) MIN (X, Y, Z, ...) MIN0 (I, J, K, ...) MIN1 (X, Y, Z, ..)	maksimum maksimum maksimum minimum minimum minimum	Belirtilen değerler arasında enbüyük ve enküçük değerler bulunur.
Kesme Yuvarlama	AINT (X) ANINT (X) NINT (X)	kesme yuvarlama	Alt sayıya yuvarlama yapılır Üst sayıya yuvarlama yapılır
Kalan	MOD (I, J) AMOD (X, Y)	kalan	Bölme sonunda kalan değer hesaplanır
Karmaşık sayı	CMPLX (X, Y) REAL (X) AIMAG (Y)	Kompleks	CMPLX (1.2, 3.2) - karmaşık sayı ifade edilir.

İşlemlerde öncelik sırası

- $A/B * C \Rightarrow (A/B) * C$
- $A * B / C \Rightarrow (A * B) / C$
- $I - J + K \Rightarrow (I - J) + K$
- $(A - B) / C + 2 \Rightarrow ((A - B) / C) + 2$
- $a^{b^x} \Rightarrow A ** (B ** X),$
- $(ab)x \Rightarrow (A ** B) ** X$

Aktarma işlemleri ve veri tipleri

İşlem

Sonuç

$$K = 6/4$$

$K > 1$ (K-değişkeni tam sayıları saklar)

$$R = 6/4$$

$R = 1.0$ (R-değişkeni gerçel sayıları saklar)

$$K = 6./4.$$

$K = 1$ (K-değişkeni tam sayıları saklar)

$$R = 6./4.$$

$11 = 1.5$ (R-değişkeni gerçel sayıları saklar)

$$S = 4^{**}(2/3)$$

$S = 1.0$ (S-değişkeni gerçel sayıları saklar)

```

C      IKINCI DERECEDEDEN BIR POLINOMUN KOKLERININ BULUNMASI
C      FORTRAN I
      READ 100,A,B,C
100    FORMAT(3F12.4)
      DISCR = B**2-4*A*C
      IF (DISCR) 10,20,30
10     X1=(-B)/2.*A
      X2=SQRTF(ABSF(DISCR))/2.*A
      PRINT 110,X1,X2
110    FORMAT(5H X = ,F12.3,4H +i ,F12.3)
      PRINT 120,X1,X2
120    FORMAT(5H X = ,F12.3,4H -i ,F12.3)
      GOTO 40
20     X1=(-B)/2.*A
      PRINT 130,X1
130    FORMAT(11H X1 = X2 = ,F12.3)
      GOTO 40
30     X1=( (-B)+SQRTF(ABSF(DISCR)) ) / (2.*A)
      X2=( (-B)-SQRTF(ABSF(DISCR)) ) / (2.*A)
      PRINT 140,X1
140    FORMAT(6H X1 = ,F12.3)
      PRINT 150,X2
150    FORMAT(6H X2 = ,F12.3)
40     CONTINUE
      STOP
      END

```

```

PROGRAM QUAD4
C
C Bu program aşağıdaki formdaki gibi
C ikinci dereceden bir polinomun katsayılarını okur
C      A * X**2 + B * X + C = 0,
C ve polinomun köklerini hesaplar (FORTRAN 77 stili).
C ikinci dereceden polinomun katsayılarının girilmesi
WRITE (*,*) 'A, B ve C katsayılarını giriniz: '
READ (*,*) A, B, C
C
C katsayıların girilmesi istenir
C
WRITE (*,100) 'Katsayılar : ', A, B, C
100 FORMAT (1X,A,3F10.4)
C
C diskriminantın kontrol edilmesi ve köklerin hesaplanması.
C
DISCR = B**2 - 4*A*C
IF ( DISCR .LT. 0) THEN
  WRITE (*,*) ' polinom sanal köklere sahiptir : '
  WRITE (*,*) ' X = ', -B/2*A, ' +i ',
    .          SQRT(ABS(DISCR))/2*A
  WRITE (*,*) ' X = ', -B/2*A, ' -i ',
    .          SQRT(ABS(DISCR))/2*A
ELSE IF ( (B**2 - 4.*A*C) .EQ. 0) THEN
  WRITE (*,*) ' polinomun gerçel kökleri aynıdır : '
  WRITE (*,*) ' X = ', -B/2*A
ELSE
  WRITE (*,*) ' polinomun iki farklı kökü vardır: '
  WRITE (*,*) ' X = ', (-B + SQRT(ABS(DISCR)))/2*A
  WRITE (*,*) ' X = ', (-B - SQRT(ABS(DISCR)))/2*A
END IF
END

```

```

PROGRAM roots
! Amaç :
!   Bu program  $A * X^{**2} + B * X + C = 0$  şeklindeki bir polinomun
!   köklerini hesaplar (Fortran 90 stili).
!
IMPLICIT NONE
! programın kullanacağı değişkenler tanımlanır
REAL :: a           !  $X^{**2}$  terimini katsayısı
REAL :: b           !  $X$  teriminin katsayısı
REAL :: c           ! polinomdaki sabit değer
REAL :: discriminant ! Diskriminant
REAL :: imag_part   ! Sanal kökün sanal kısmı
REAL :: real_part   ! Sanal kökün gerçel kısmı
REAL :: x1          ! Gerçel kök
REAL :: x2          ! Gerçel kök
! eşitliğin katsayıları klavyeden girilmesi istenir
WRITE (*,*) 'bu program 2nci dereceden polinomları çözer'
WRITE (*,*) 'polinom  $A*X^{**2}+B*X+C=0.0$  şeklindedir. '
WRITE (*,*) ' A, B, ve C katsayıları giriniz:'
READ (*,*) a, b, c
! katsayılar ekrana yeniden yazılır.
WRITE (*,*) ' A, B, ve C katsayıları : ', a, b, c
! diskriminant hesaplanır
discriminant = b**2 - 4. * a * c
! diskriminanta bağlı olarak denklem çözülür.
IF ( discriminant > 0. ) THEN ! iki gerçel kök vardır, bunlar:
  X1 = ( -b + sqrt(discriminant) ) / ( 2. * a )
  X2 = ( -b - sqrt(discriminant) ) / ( 2. * a )
  WRITE (*,*) 'Eşitliğin iki kökü vardır:'
  WRITE (*,*) 'X1 = ', x1
  WRITE (*,*) 'X2 = ', x2
ELSE IF ( discriminant == 0. ) THEN ! özdeş iki kök vardır, bunlar
  x1 = ( -b ) / ( 2. * a )
  WRITE (*,*) 'Polinomun kökleri aynıdır :'
  WRITE (*,*) 'X1 = X2 = ', x1
ELSE ! kökler sanaldır,...
  real_part = ( -b ) / ( 2. * a )
  imag_part = sqrt ( abs ( discriminant ) ) / ( 2. * a )
  WRITE (*,*) 'Eşitliğin kökleri sanaldır:'
  WRITE (*,*) 'X1 = ', real_part, ' +i ', imag_part
  WRITE (*,*) 'X2 = ', real_part, ' -i ', imag_part
END IF
END PROGRAM

```

Programlama Öğütleri

1. Fortran programlarınızı serbest formatta yazınız,
2. Anlaşılır (fakat türkçe karakter içermeyen) değişkenler kullanınız. Örneğin adi, soyadi, pi, toplam vs.
3. Daima IMPLICIT NONE ifadesini kullanınız. Bu ifade derleme aşamasında hata noktasını yakalamanızı sağlar.
4. Her programınız için veri sözlüğü oluşturun. Yani programda kullandığınız değişkenleri açıkça deklare ediniz ve tanımlayınız. Fiziksel niceliklere uygun değişkenlerle verilerinizi isimlendiriniz (yol, enerji, is, saat, zaman vs).
5. Sabitleri program içinde değişik olarak yazmamaya dikkat edin. Örneğin π sabiti için programın bir yerinde 3.14 başka bir yerinde 3.141593 gibi bir değer kullanmayınız.
6. Sabitler için kullanabileceğiniz en fazla duyarlı hane sayısını kullanınız. Örneğin π için 3.14 yerine 3.141593

Programlama Öğütleri

7. Programınızda gerçel dünyadaki değişkenler tamsayı kullanmanızı gerektiyorsa tamsayı, gerçel sayı kullanmanızı gerektiriyorsa gerçel sayı kullanınız. Örneğin zaman veya yol için gerçel sayı değişkenleri, sayaç için tamsayı değişkenleri kullanınız.
8. Tamsayı ve gerçel sayılarla ilgili işlemler yapacaksanız tamsayıları gerçel sayıya veya gerçel sayıları tamsayıya çevirerek yapınız. Bunu REAL, İNT ve NINT fonksiyonları ile yapınız.
9. Formülün okunmasını kolaylaştıracaksa fazla parantez kullanınız.
10. Klavyeden girilecek verilerin ne olduğunu kullanıcıya ekrandan yansıtınız.
11. Değişkenlere, onları kullanmadan önce mutlaka başlangıç atamalarını yapınız. Atama işlemlerini READ veya deklarasyon satırlarında yapınız.
12. İşlem çıktılarında fiziksel birim gerekiyorsa kullanınız. Bu program çıktılarının kolay değerlendirilmesini sağlar.

Programlama Öğütleri

13. Programınızın dökümanı şu özellikleri içermelidir:

- AD : programın içerildiği dosya adı
- PROGRAMCI : Programı yazan kişi
- TARİH : programın yazıldığı tarih
- SÜRÜM : Programın sürüm numarası
- AMAÇ : Problemin tanımı
- GİRDİ : Giriş verileri ve formatları
- ÇIKTI : Çıkış verileri ve formatları
- PARAMETRE : Parametre isimleri, anlamları, ve fonksiyonları
- STRATEJİ : Programın probleme yaklaşımı
- ALTPROGRAM: Altprogramların listesi ve tanımları
- SINIRLARI ve KISITLAMALARI : Programın yapabilecekleri ve yapamayacakları
- HATA KONTROLÜ : Oluşabilecek hataların listesi açıklamaları
- DEĞİŞKEN ve SABİTLER : Değişkenlerin ve sabitlerin listesi ve kullanım amaçları

Sorular

1. Aşağıdaki aritmetik işlemlerin sonuçlarını yazınız.

a) $8/2$ b) $6/10$ c) $(6+8)/5$ d) $4*5/3/2$ e) $(4*5)/(3/2)$ f) $(4*(5/3))/2$

2. Aşağıdaki sayıları üstel formda gösteriniz.

a) 10.3 b) 2345.7 c) 378.66 d) 0.453 e) 0.00056 f) 0.000678

3. Aşağıdaki değişkenleri FORTRAN veri tiplerine göre ayırd ediniz.

a) JACK b) PI c) ALAN d) KARE e) MATRIS f) HACIM

5. Kenar uzunlukları 5.5, 7.25 ve 3.75 olan dikdörtgen prizmanın hacmini, toplam yüzey alanını hesaplayan algoritmayı yazınız.

6. Yükseklik ve çap değerleri aşağıda verilen bir silindirin hacmini hesaplayan algoritmayı yazınız.

Çap (cm)37.420 Yükseklik (cm)45.512

7. Paralel kenar uzunlukları $a=4.75$ ve $b=10.25$ birim olan ve yüksekliği $h=8.375$ birim olan bir yamuğun alanını $A=0.5h(a+b)$ denklemi ile hesaplayan algoritmayı yazınız.

8. Aşağıdaki değişkenlerde saklanan değerleri belirtiniz.

$A=2*6+1$ (A-gerçel sayı değişkeni)

$A=2/3$ (A-gerçel sayı değişkeni)

$I=2*10/4$ (I-tam sayı değişkeni)

$A=2.*6./4$ (A-gerçel sayı değişkeni)

$A=6.0*(1.0/6.0)$ (A-gerçel sayı değişkeni)

$I=2*(10/4)$ (I-tam sayı değişkeni)

$I=100*(99/100)$ (I-tam sayı değişkeni)

$I=19/4+5/4$ (I-tam sayı değişkeni)

9. Aşağıdaki ifadeleri Fortran da matematiksel formül olarak yazınız.

a) BETA değerine 2 ekleyiniz ve sonucu DELTA değişkenine aktarınız.

b) A nın karekökünü alıp, buna B nin karesini ekleyiniz. Bunların toplamının karekökünü alıp bunu C değişkenine aktarınız.

c) A nın değerini -0.1 ile çarpıp sonucu tekrar A değişkenine aktarınız.

d) THETA değişkenini $\pi/180.0$ ile çarpıp bunun kosünüs değerini hesaplayan ve sonucu RHO değişkeni ile çarpınız.

10. a. Okunan değerler : A,B,C
Yazılan değerler : A,B,C,X1,X2
 $X1=(-B+(B**2-4*A*C)**0.5)/(2* A)$
 $X2=(-B-(B**2-4*A*C)**0.5)/(2* A)$

b. Okunan değerler : A, B, C, X
Yazılan değerler : A, B, C, X, R
 $R=B*C(6*X**2*(1-X/A)**2+B**2*(1-X/A)**2)/12$

11. $Y(X)=16.7*X+0.92*X**2-1.02*X**3$ fonksiyonunu X in 1.0 dan 9.9 değerine kadar 0.1 aralıklarla hesaplayan, X ve Y değerlerini ekrana yazan bir algoritma hazırlayınız.

12. Bir üçgenin kenar uzunlukları A, B, C girildikten sonra üçgenin alanını $S=(A+B+C)/2$ ve $ALAN=(S(S-A)(S-B)(S-C))**0.5$ formülü ile hesaplayan bir algoritma geliştiriniz.