


```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84

```



=====  
 Hazirlayan : Aras. Gor. Ozgur Cobanoglu  
 Danisman : Prof. Dr. M. Nizamettin Erduran  
 -----  
 Her türlü oneri ve düzeltme için lütfen  
 aşağıdaki adresleri kullanın.  
 -----  
 İstanbul Üniversitesi, fen fakültesi, nükleer  
 fizik anabilim dalı, oda NF205, Vezneciler,  
 +90-212-455-57-00 dahili:15410 İSTANBUL.  
 -----  
 ozgur@nucleus.istanbul.edu.tr  
 Ozgur.Cobanoglu@cern.ch  
 ozgur\_cobanoglu@hotmail.com  
 =====

<-> Asal Fonksiyonlar (Intinsic **FUNCTION**)  
 -----  
 Fortran77 fortran komutlari da denebilecek olan asal  
 fonksiyonlarinin tanimlandigi bir kutuphaneye sahiptir ve bu  
 kutuphane icerisinde tanimlanmis olan fonksiyonlari kullanmak için  
 asagidaki gibi bir yol izlenir (simdiye kadar **WRITE** ve **READ**  
 komutlarinin kullanıminda oldugu gibi) :  
 -----  
 fonksiyonAdi({parametreListesi})  
 -----  
 Burada parametreListesi, fonksiyonAdi adli fonksiyonun islem  
 yapabilmek için bilmesi gerekenleri iceren degiskenler olarak  
 tanimlanir. Ornegin OPEN() fonksiyonu bir unite numarasina, dosya  
 ismine, STATUS ve FORM gibi degiskenlerin tuttuğu bilgilere ihtiyac  
 duyar ya da bunlari kullanir. Fonksiyonlarin aldigi bazı  
 parametreler zaruridir; ornegin UNIT numarası olmayan ya da dosya  
 adi olmayan bir OPEN fonksiyonu dusunulemez. Bazı parametreler ise  
 secime baglidir; ornegin FORM gibi.

Bir fonksiyonun kendisini cagirana, kendisine aktarilan  
 parametrelerin tuttuğu degerlere göre bir deger dondurur. Bazı  
 önemli asal fonksiyonlar (fortran komutlari) ABS, ACOS, COS,  
 DOT\_PRODUCT, EXP, INT, LEN, LOG, LOG10, MAX, MIN, MOD, SQRT olarak  
 verilebilir. Bunlar ile ilgili daha kapsamlı bilgiye ilerleyen  
 bolumlerde yer verilecektir.

Pek çok asal fonksiyon (fortran kutuphanesinde bulunan on tanimli  
 fortran komutlari) kullanilmasi kolay ve hep ayni bicimde  
 cagirilmaya göre tasarlanmistir; tum bu asal fonksiyonlar farkli  
 (ama karisik olmamak üzere) veri tipleri ile (**INTEGER**, **REAL** v.b.)  
 calisacak durumdadir (bu, fonksiyonlarin asiri yuklenmesi olarak  
 bilinir). Bazı fonksiyonlarin kullanimi ise sadece belirli veri  
 tiplerinin kullanimini gerektirir; ornegin SQRT(4) hatalidir ve  
 bunun yerine SQRT(4.0) kullanilmalidir. Ornek olarak bir fonksiyonun  
 calistirilmasi asagidaki gibi olacaktir :  
 -----  
 kareKok = SQRT(4.0)  
 -----  
 Burada SQRT() fonksiyonu 4.0 degerini parametre olarak alarak isleme  
 sokar. Islem sonucu olan 2.0 degeri SQRT() fonksiyonunun cagirana  
 dondurdugu degerdir ve bu deger = isareti araciligıyla kareKok adli  
 degiskene atanir; sonucta kareKok adli degisken 2.0 degerini  
 tutacaktır. kareKok adli degiskenin **REAL** olarak tanimlandigi  
 varsayilmistir. SQRT(1.0 \* birSayi) kullanimi birSayi degiskeni  
 hangi tipte olursa olsun fonksiyonun istenen degeri dondurmesini  
 saglayacaktır.

<-> Asal Altprogramlar (Intinsic **SUBROUTINE**)  
 -----  
 Subroutines yani altprogramlar fonksiyonlara çok benzerler fakat  
 önemli bir farkları vardır. Fonksiyonlar sadece bir deger  
 dondururler ve kendilerine gelen parametreleri degistiremezler.  
 Altprogramlar ise fonksiyonlar gibi deger donduremezler fakat  
 kendilerine 'gecilen' parametreleri degistirme yetenegine  
 sahiptirler. Aynı zamanda altprogramlar birden çok degerin  
 dondurulmesi gereken durumlarda kullanilmalidir. Asal kelimesi  
 ontanimli fortran kutuphane fonksiyon ya da altprogramlarini  
 tanımlamak için kullanılmaktadır. Kullanıcılar kendi fonksiyon  
 ve/veya altprogramlarini tanımlama özgurlugüne sahiptir ve bu  
 fonksiyon ve altprogramlar **external** ya da hiçbir betimleyici  
 kullanmaksizin sadece altprogramlar ve fonksiyonlar olarak  
 isimlendirilir; kullanıcının tanımladığı fonksiyon ve  
 altprogramlarda ontanimli fonksiyon ve altprogramlar

```

85 kullanilabilir. Altprogramlar CALL (cagir) komutu kullanilarak
86 cagirilirlar :
87
88     CALL altProgramAdi(argumanListesi)
89
90 Bazi onemli asal altprogramlar (yani fortran kutuphanesi icinde
91 varolan ontanimli fortran komutlari) DATE_AND_TIME, MVBITS,
92 RANDOM_NUMBER, RANDOM_SEED, SYSTEM_CLOCK v.b. olarak verilebilir.
93
94 <-> Fortran Asal Fonksiyon ve Altprogramlari
95 -----
96 Asagida bazi -tamami degil- kullanisli fortran FUNCTION ve SUBROUTINE
97 tipi altprogramlari parametreleri ve kısa aciklamalari ile birlikte
98 gosterilmistir. Bu asal fonksiyonlarin tek tek denemesi onerilir.
99
100 <-> Sayisal Fonksiyonlar
101 -----
102 ABS(A) : tamsayi, reel ya da karmasik bir sayinin mutlak degerini dondurur
103 ACOS(X) : x' in ters cosinusunu dondurur - ArcCos(x)
104 AIMAG(Z) : Z karmasik sayisinin imajiner kismini dondurur
105 COS(X) : x' in kosinusunu dondurur
106 COSH(X) : hiperbolik kosinus
107 EXP(X) : exponansiyel x; e uzeri x
108 LOG(X) : dogal logaritma, ln(x), gosterime dikkat edin
109 LOG10(X) : 10 tabanina gore logaritma, log(x), gosterime dikkat edin
110 MAX(a1, a2, a3, ...) : argumanlarin en buyugunu dondurur
111 MIN(a1, a2, a3, ...) : argumanlarin en kucugunu dondurur
112 MOD(A, P) : A'yi P'ye boler ve kalan dondurur ornegin MOD(2.2, 2.0)
113     0.2 dondurur
114 SQRT(X) : kare kok x dondurur
115
116 <-> Karakter Isleme Fonksiyonlari
117 -----
118 Bu kisimdaki fonksiyonlar karakter katarlari ile ilgili bolumde
119 kullanilmistir; burada tekrar edilmeyecektir.
120
121 ACHAR(I) : I ASCII koduna sahip karakteri dondurur (Bkz. EK1, EK2)
122 IACHAR(C) : C karakterinin ASCII kodu (Bkz. EK1, EK2)
123 LGE(katar1, katar2) : katar1 ile katar2' yi karsilastirir, katar1
124     katar2'den sonra geliyorsa .TRUE. dondurur
125     yani .GT. karsilastirma operatorunun karakter
126     katarlarini karsilastiran halidir, karsilastirma
127     ASCII tablosundaki siraya gore yapilir (benzer
128     bicimde LLE, LLT, LGE komutlari da vardir)
129 LEN(katar) : katar bir karakter dizisi ise karakter sayisi, bir dizi
130     ise dizinin eleman sayisini dondurur
131 TRIM(katar) : katar'in sonundaki bosluklari yok edilmiş hali
132     dondururlur
133
134 <-> Bit Duzeyinde Islem Fonksiyonlari
135 -----
136 BTEST(I, yer) : I tamsayisinin yer' inci biti 1 ise .TRUE. dondurur
137 IAND(I, J) : I ve J tamsayilari uzerinde karsilikli tum bitler
138     arasinda mantiksal VE islemi gerceklestirerek sonucu
139     dondurur
140 IBITS(I, yer, uzunluk) : I tamsayisinde, yer'inci bitten baslayan
141     ve uzunluk kadar uzun olan bir bit dizisinin
142     onluk sistemdeki karsiligi; ornegin I=1[1001]010
143     ikilik sistemdeki sayisi icin IBITS(I, 3, 4)
144     islemi 9 sayisini dondurur ([] icinde gosterilmistir).
145 NOT(I) : I tamsayisinin tum bitleri degillenir ornegin I=1010 ikilik
146     sistemdeki sayisi I=101 halini alir.
147 CALL MVBITS(den, denYer, uzunluk, ya, yaYer) : den tamsayisinin denYer'inci
148     bitinden uzunluk kadar bir kisimi
149     alip, ya tamsayisinin
150     yaYer'inci bitinden itibaren
151     yerlestirir. (bir SUBROUTINE
152     olduguna dikkat edin)
153
154 Ozellikle IBITS() ve CALL MVBITS() fonksiyon ve altprogrami SORTing denen
155 nukleer ve yuksek enerji fizigi alanlarinda kullanılan bir veri isleme
156 yontemi esnasinda oldukca sik kullanilir. Verinin herhangi bir dosyadan
157 ya da cihazdan dogrudan ve ikilik sistemde okundugu durumlarda okunan
158 verinin islenmesi ancak bir duzeyinde islem yapilarak mumkun olur. Veri
159 dosyalarini ikilik sistemde saklamak su acidan olumludur : disk uzerinde
160 veri mumkun en kucuk yeri kaplar ve dosyanin kendisini yazan program
161 disinda degistirilmesi zorlasir (ikilik olarak kaydedilmiş dosyalar
162 karakter karakter veri okuyan uygulamalarca duzgun goruntulenemez; ornegin
163 bir kelime islemci ile acildiginda anlamsiz sekiller gorulur, dosyaya
164 ikilik sistemde erisim bu ders kapsamı disinda oldugundan bu kadari ile
165 yetinilmistir. Dosyaya bu turu erisim icin internetten aramak uzere
166 anahtar kelimeler binary file access ve direct binary access olabilir)
167
168 <-> Uygulama

```

```

169 -----
170 1 - Asagidaki program girilen sayinin tek mi cift mi oldugunu belirler.
171 Cift bir sayinin 2' ye bolumunden kalan yoktur ve tek bir tamsayinin
172 ikiye bolumunden kalan her zaman 1 olur. Bu bilgi kullanilarak bir
173 sayinin tek mi cift mi oldugu belirlenebilir. Bunu belirlemek bazi
174 durumlarda gereklidir; ornegin ayna cekirdeklerin ortalama baglanma
175 enerjisinin bulunmasi probleminde girilen cekirdegin aynasinin
176 notron ve proton sayilarinin belirlenmesi -bunun bilgisayara yaptirilmasi-
177 bu yolla saglanir. Cikti asagidaki gibi olur.
178
179 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ciftMee
180 Cift mi tek mi icin bi tam sayi girin :
181 3
182 Sayi tektir.
183 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ciftMee
184 Cift mi tek mi icin bi tam sayi girin :
185 4
186 Sayi cifttir.
187 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler>
188 -----
189 PROGRAM ciftMeee
190 IMPLICIT NONE
191 INTEGER sayi
192 WRITE(*,*)'Cift mi tek mi icin bi tam sayi girin : '
193 READ(*,*)sayi
194 IF ( MOD(sayi,2).EQ.0 ) WRITE(*,*)'Sayi cifttir.'
195 IF ( MOD(sayi,2).NE.0 ) WRITE(*,*)'Sayi tektir.'
196 END
197 -----
198
199 2 - Asagidaki program girilen tamsayinin ikilik sistemdeki karsiligini
200 ekrana basar; ciktiesi asagidadir. sayi degiskeni INTEGER olarak
201 tanimlanmistir ve INTEGER, 2 sekizli (byte) uzunluktadir; yani 16
202 bitten olusur. Bitler 0-15 olarak isimlendirilir dolayisi ile
203 DO dongusu 0'dan baslar ve 15'te biter. BTEST fonksiyon ile her
204 bitin 1 olup olmadigi sinanir ve 1 olmasi durumunda 1 ve 0 olmasi
205 durumunda 0 ekrana basilir. Alt alta degil de yanyana cikti verilmesi
206 icin degisik bir yol dusununuz.
207
208 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ikilikGosterim
209 Ikilik sisteme cevrilecek tamsayiyi girin :
210 235
211 -----
212 1
213 1
214 0
215 1
216 0
217 1
218 1
219 1
220 0
221 0
222 0
223 0
224 0
225 0
226 0
227 0
228 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler>
229 -----
230 PROGRAM ikilikGosterim
231 IMPLICIT NONE
232 INTEGER i, sayi
233 LOGICAL birMi
234 c
235 WRITE(*,*)'Ikilik sisteme cevrilecek tamsayiyi girin : '
236 READ(*,*)sayi
237 WRITE(*,*)'-----'
238 c
239 DO i=0,15
240 birMi = BTEST(sayi, i)
241 IF ( birMi ) THEN
242 WRITE(*,*) '1'
243 ELSE
244 WRITE(*,*) '0'
245 END IF
246 END DO
247 c
248 END
249 -----
250
251 3 - Bir oncekinin aynisidir. Yanyana cikti icin farkli bir yolu
252 gostermektedir. 0'dan 15'e kadar elemanlari olan bir dizi

```

```

253      tanımlanmaktadır. Bu dizi girilen sayı için yapılan her bir
254      karşılama işlemi sonucunda 1 ya da 0 sayılarını tutacak
255      biçimde kullanılır. Sonra bu dizi, tek satır DO döngüsü
256      aracılığı ile ekrana basılır. Diziye atama yapılırken i
257      yerine 15-i indislerinin kullanılmasının nedeni ikilik
258      sistemdeki bir sayının ilk bitinin en sağda bulunmasıdır;
259      aksi halde ikilik sistemdeki sayı ekrana ters yazılır.
260      (MSB - Most Significant Bit, LSB - Least Significant Bit)
261
262      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ikilikGosterimII
263      İkilik sisteme çevrilecek tamsayıyı girin :
264      1
265      -----
266      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
267      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ikilikGosterimII
268      İkilik sisteme çevrilecek tamsayıyı girin :
269      2
270      -----
271      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
272      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ikilikGosterimII
273      İkilik sisteme çevrilecek tamsayıyı girin :
274      222
275      -----
276      0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0
277      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler>
278
279      PROGRAM ikilikGosterimII
280      IMPLICIT NONE
281      INTEGER i, sayi, dizi(0:15)
282      LOGICAL birMi
283
284      c
285      WRITE(*,*)'İkilik sisteme çevrilecek tamsayıyı girin : '
286      READ(*,*)sayi
287      WRITE(*,*)'-----'
288
289      c
290      DO i=0,15
291      birMi = BTEST(sayi, i)
292      IF ( birMi ) THEN
293      dizi(15-i)=1
294      ELSE
295      dizi(15-i)=0
296      END IF
297      END DO
298
299      c
300      WRITE(*,*) (dizi(i),i=0,15)
301      END
302
303      -----
304
305      4 - Aşağıdaki program bir tamsayıyı değerler ve sonuçları
306      ekrana basar. Karşılıklı bitlerin birbirlerinin değeri
307      olduklarına dikkat ediniz.
308
309      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler> ./ikilikGosterimIII
310      Değillenecek tamsayıyı girin :
311      255
312      Girilen tamsayı : 255
313      0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
314      Değillenen hali : -256
315      1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
316      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler>
317
318      PROGRAM ikilikGosterimIII
319      IMPLICIT NONE
320      INTEGER i, sayi, dizi(0:15)
321      LOGICAL birMi
322
323      c
324      WRITE(*,*)'Değillenecek tamsayıyı girin : '
325      READ(*,*)sayi
326
327      c
328      DO i=0,15
329      birMi = BTEST(sayi, i)
330      IF ( birMi ) THEN
331      dizi(15-i)=1
332      ELSE
333      dizi(15-i)=0
334      END IF
335      END DO
336
337      c
338      PRINT*,'Girilen tamsayı : ',sayi
339      WRITE(*,*) (dizi(i),i=0,15)
340
341      c
342      DO i=0,15
343      birMi = BTEST(sayi, i)
344      IF ( birMi ) THEN

```

```
337         dizi(15-i)=0
338         ELSE
339         dizi(15-i)=1
340         END IF
341     END DO
342 c
343     PRINT*,'Degillenmis hali :',NOT(sayi)
344     WRITE(*,*)(dizi(i),i=0,15)
345 c
346     END
347 _____
```