

```

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

```

Tarih : 21-12-2003

Hazirlayan : Aras. Gor. Ozgur Cobanoglu
Danisman : Prof. Dr. M. Nizamettin Erduran

Her turlu oneri ve duzeltme icin lutfen asagidaki adresleri kullanin.

Istanbul Universitesi, fen fakultesi, nukleer fizik anabilim dali, oda NF205, Vezneciler, +90-212-455-57-00 dahili:15410 ISTANBUL.

ozcoban@istanbul.edu.tr ozgur@nucleus.istanbul.edu.tr
Ozgur.Cobanoglu@cern.ch ozgur_cobanoglu@hotmail.com

1 - Asagidaki program girilen sayinin cift ya da tek olusuna gore ekrana 'tek' veya 'cift' yazar; ciktiesi asagidaki gibidir :

```

ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./ciftMee
Cift mi tek mi icin bi tam sayi girin :
89
Sayi tektir.
ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./ciftMee
Cift mi tek mi icin bi tam sayi girin :
42
Sayi cifttir.
ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./ciftMee
Cift mi tek mi icin bi tam sayi girin :
0
Sayi cifttir.
ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./ciftMee
Cift mi tek mi icin bi tam sayi girin :
-1
Sayi tektir.
ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>

```

Programda MOD(a,b) kutuphane fonksiyonu kullanilmistir; bu fonksiyon a' nin b' ye bolumunden kalani dondurur. Program cift sayilarin 2'ye bolunebilen sayilar oldugu tanimlana dayanilarak yazilmistir.

```

PROGRAM ciftMee
IMPLICIT NONE
INTEGER sayi
WRITE(*,*)'Cift mi tek mi icin bi tam sayi girin : '
READ(*,*)sayi
IF ( MOD(sayi,2).EQ.0 ) WRITE(*,*)'Sayi cifttir.'
IF ( MOD(sayi,2).NE.0 ) WRITE(*,*)'Sayi tektir.'
END

```

Yukaridaki program ile ayni isi yapan ve MOD(a,b) fonksiyonunu kullanmayan bir baska esdeger program asagidaki gibi olabilir :

```

PROGRAM ciftMeeII
IMPLICIT NONE
INTEGER sayi, i
REAL r
WRITE(*,*)'Cift mi tek mi icin bi tam sayi girin : '
READ(*,*)sayi
r = sayi/2.0
i = sayi/2.0
IF ( (r-i).EQ.0.0 ) WRITE(*,*)'Sayi cifttir.'
IF ( (r-i).NE.0.0 ) WRITE(*,*)'Sayi tektir.'
END

```

2 - Asagidaki program girilen iki pozitif sayidan byuk olani kucuk olana boler ve bolumun tam olup olmamasina gore ekrana 'tam' veya 'tem degil' yazar; MOD(a,b) fonksiyonunun kullanilmasi klaylik saglayacaktır. Ciktisi asagidaki gibidir :

```

ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./buyukBoluKucuk
iki sayi girin :
6
2
Sayilar tam bolunur...
ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./buyukBoluKucuk
iki sayi girin :

```

```

85 6
86 4
87 Sayilar tam bolunMuyor...
88 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
89 -----
90 PROGRAM buyukBoluKucuk
91 IMPLICIT NONE
92 REAL a, b, r
93 INTEGER i
94
95 WRITE(*,*)'Iki sayi girin : '
96 READ(*,*)a, b
97
98 IF (a.GT.b) THEN
99     r = a/b
100    i = a/b
101 ELSE IF (a.LT.b) THEN
102     r = b/a
103     i = b/a
104 ELSE
105     WRITE(*,*)'Girilen sayilar esit !..'
106 END IF
107
108 IF ((r-i).EQ.0.0) THEN
109     WRITE(*,*)'Sayilar tam bolunur...'
110 ELSE
111     WRITE(*,*)'Sayilar tam bolunMuyor...'
112 END IF
113
114 END
115 -----
116
117 3 - Asagidaki programi kendi icindeki aciklama ile degerlendirin;
118 ciktisi asagidaki gibidir :
119
120 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./abcdabcd
121 abcdabcd/x=tamsayi icin 0<x<50000 listesi
122 -----
123 Tam bolen bulundu : 73
124 Tam bolen bulundu : 137
125 Tam bolen bulundu : 10001
126 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
127 -----
128 c abcdabcd bicimindeki sayilari tam bolen
129 c sayilari belirli bir aralikta bulur. Boyle
130 c bir sayi 10001000a+1000100b+100010c+10001d
131 c biciminde yazilir ve bu sayinin a,b,c ve d
132 c her ne olursa olsun belirli bir sayiya
133 c bolunebilmesi demek 10001000, 1000100, 100010, 100010
134 c ve 10001 katsayilarinin hepsini ayni anda
135 c bolebiliyor olmasi demektir. Bolun isleminde
136 c kalan olup olmadigi, 10001/x ve 10001.0/x gibi
137 c biri REAL digeri INTEGER olan degerler arasindaki
138 c farklari toplamak ve bu toplami sifir ile sinamak
139 c yolu ile belirlenmistir. Toplam sifirdan farkli
140 c olursa tam bolunum olmadigi anlasilacaktir.
141
142 PROGRAM abcdeabcde
143 IMPLICIT NONE
144 INTEGER i1, i2, i3, i4, i5, x
145 REAL r1, r2, r3, r4, r5, f1, f2, f3, f4, f
146 WRITE(*,*)' abcdabcd/x=tamsayi icin 0<x<50000 listesi'
147 WRITE(*,*)'-----'
148 DO x=2,999999
149     r1=10001000.0/x
150     r2=1000100.0/x
151     r3=100010.0/x
152     r4=10001.0/x
153     i1=10001000/x
154     i2=1000100/x
155     i3=100010/x
156     i4=10001/x
157     f1=r1-i1
158     f2=r2-i2
159     f3=r3-i3
160     f4=r4-i4
161     f=f1+f2+f3+f4
162     IF (f.EQ.0.0) WRITE(*,*)' Tam bolen bulundu :',x
163 END DO
164 END
165 -----
166
167 Programin ciktisi abcdabcd biciminde herhangi bir sayi
168 gozune alinarak denenebilir; ornek olarak

```

```

169 56785678/73=777886, 23452345/137=171185 ve
170 98739873/10001=9873 bolumlerinin hepsinin tam olmasi
171 gerekmektedir.
172
173 4 - Programin FUNCTION tipi alt program kullanilarak yeniden
174 yazilmasi istenmektedir; cikti ve kaynak kod asagidaki gibi
175 olabilir :
176
177 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./comptonSacilmasi
178 Gelen fotonun enrjisini MeV olarak giriniz :
179 0.662
180 Sacilma acisini Radyan olarak giriniz :
181 3.14
182 Sacilan fotonun enerjisi = 0.184349939 MeV
183 Bu enerji icin Compton kenari = 0.477650046 MeV
184 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
185
186 PROGRAM comptonSacilmasi
187 IMPLICIT NONE
188 COMMON /degiskenler/Egelen, Esacilan, teta, comptonKenari
189 REAL Egelen, Esacilan, teta, comptonKenari, HESAPLA, a
190 WRITE(*,*)'Gelen fotonun enrjisini MeV olarak giriniz : '
191 READ(*,*)Egelen
192 WRITE(*,*)'Sacilma acisini Radyan olarak giriniz : '
193 READ(*,*)teta
194 a = HESAPLA()
195 WRITE(*,*)'Sacilan fotonun enerjisi =',Esacilan,' MeV'
196 WRITE(*,*)'Bu enerji icin Compton kenari =',comptonKenari,' MeV'
197 END
198
199 REAL FUNCTION HESAPLA()
200 COMMON /degiskenler/eg, es, t, ck
201 es = eg / (1+(eg/0.511)*(1-cos(t)))
202 ck = eg - es
203 END
204
205
206 5 - Asagidaki program iki degiskenin degerlerinin nasil deistokus
207 edilecegine bir ornek olarak verilmektedir. a ve b iki degisken
208 olmak uzere bu iki degiskenin degerlerini su sekilde birbirlerine
209 atayabiliriz : gecici bir g degiskeni tanimlayip a' nin degerini
210 g' ye atayip a' ya da b' yi atariz; bu durumda g artik a' nin eski
211 degerini tutar ve son olarak da b' ye g' yi atarsak a ve b' nin
212 degerleri yer degistirmis olur. Cikti ve kod asagidaki gibi olabilir :
213
214 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./degerDegistir
215 111 222 888 999
216 888 999 111 222
217 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
218
219
220 PROGRAM degerDegistir
221 IMPLICIT NONE
222 INTEGER a1, b1, a2, b2
223 COMMON/degerler/a1,a2,b1,b2
224 a1 = 111
225 a2 = 222
226 b1 = 888
227 b2 = 999
228 WRITE(*,*)a1,a2,b1,b2
229 CALL degistir()
230 WRITE(*,*)a1,a2,b1,b2
231 END
232
233 SUBROUTINE degistir()
234 INTEGER a1, b1, a2, b2
235 COMMON/degerler/a1,a2,b1,b2
236 INTEGER gecici
237 c a1 <-> b1 degisimi yapiliyor
238 gecici = a1
239 a1 = b1
240 b1 = gecici
241 c a2 <-> b2 degisimi yapiliyor
242 gecici = a2
243 a2 = b2
244 b2 = gecici
245 END
246
247
248 6 - Asagida kesikli bir fonksiyonun her kisminin ayri bir
249 FUNCTION tipi alt programda hesaplandigi ornek verilmistir.
250 Cikti ve kod asagidaki gibi olabilir :
251
252 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./kesikFonk

```

```

253 f(x) = | x**2      , x <= 0 |
254       | Sqrt(x)   , 0 < x < 1 |
255       | x         , x >= 1 |
256       | cikis     , x = 555 | fonksiyonu icin x girin :
257 0.09
258 f(x)= 0.300000012
259 Bi daa hesap yapmak istiyo musun ? (e/h)
260 h
261 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
262
263 PROGRAM kesikFonk
264 IMPLICIT NONE
265 REAL x, kendisi, kareKoku, karesi
266 CHARACTER cevap
267 2 PRINT*, 'f(x) = | x**2      , x <= 0 | '
268 PRINT*, '          | Sqrt(x)   , 0 < x < 1 | '
269 PRINT*, '          | x         , x >= 1 | '
270 PRINT*, '          | cikis     , x = 555 | fonksiyonu icin x girin : '
271 READ(*,*)x
272 c
273 IF (x.LE.0.0) THEN
274 PRINT*, 'f(x)=', karesi(x)
275 GO TO 1
276 ELSE IF ( (x.GT.0.0).AND.(x.LT.1.0) ) THEN
277 PRINT*, 'f(x)=', kareKoku(x)
278 GO TO 1
279 ELSE IF ( (x.GE.1.0).AND.(x.NE.555) ) THEN
280 PRINT*, 'f(x)=', kendisi(x)
281 GO TO 1
282 ELSE IF (x.EQ.555) THEN
283 STOP 'Programdan cikildi !...'
284 END IF
285
286 1 PRINT*, 'Bi daa hesap yapmak istiyo musun ? (e/h) '
287 READ(*,*)cevap
288 IF (cevap.EQ.'e') GOTO 2
289 555 END
290
291 REAL FUNCTION karesi(x)
292 IMPLICIT NONE
293 REAL x
294 karesi = x*x
295 END
296
297 REAL FUNCTION kareKoku(x)
298 IMPLICIT NONE
299 REAL x
300 kareKoku = SQRT(x)
301 END
302
303 REAL FUNCTION kendisi(x)
304 IMPLICIT NONE
305 REAL x
306 kendisi = x
307 END
308
309
310 Yukaridaki programda FUNCTION tipi alt programlarin tanimlanmasina
311 dikkat ediniz : REAL FUNCTION kareKoku(x) yani dondurduklari veri
312 tipi cinsinden tanimlaniyorlar. kareKoku(x) fonksiyonunun bir REAL
313 veri dondurdugune ve ana programda da REAL bir degiskenmis gibi
314 tanimlandigina dikkat ediniz.
315
316 7 - Ucuncu dereceden bir fonksiyonun koklerinin hesaplanmasi
317 icin Newton yontemi kullanilmistir; cikti ve kod asagidaki
318 gibidir :
319
320 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./newtonMethodII
321 Iterasyona baslamak icin ilk degeri gir :
322 99
323 66.5005722 1853670.
324 44.8345718 549232.
325 30.3909969 162733.031
326 20.7625885 48215.5898
327 14.3446102 14285.0322
328 10.0674009 4231.88818
329 7.21809244 1253.41687
330 5.32180214 371.065094
331 4.06248808 109.732895
332 3.23028445 32.3711166
333 2.68655014 9.49540043
334 2.34080029 2.74781585
335 2.13543773 0.768399179
336 2.03379011 0.195436627

```

```

337      2.00295949  0.0372925885
338      2.00002599  0.00298581738
339      2.  2.59896515E-05
340      2.  0.
341      2.  0.
342      2.  0.
343      2.  0.
344      TAMAM : Yakinsadi !..
345      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
346
-----
347      c bir fonksiyonun kokunu bulan iteratif program
348      c ciktinin solunda kok ve saginda hata gorulur
349
350      PROGRAM newtonMethodII
351      IMPLICIT NONE
352      INTEGER i
353      REAL x, maxHata, f, df
354      LOGICAL yakinsadiMi
355
356      c sonucun sahip olmasini istedigimiz en buyuk hata
357
358      PARAMETER (maxHata = 1e-6)
359      yakinsadiMi = .FALSE.
360
361      c ihtiyac duyulan deger okunuyor
362
363      WRITE(*,*)'Iterasyona baslamak icin ilk degeri gir : '
364      READ(*,*)x
365
366      c iterasyona baslaniyor f : fonksiyon ve df : f'in turevi
367
368      DO i=0,20
369      f = 2*x**3 - 9*x**2 + 13*x - 6
370      df = 6*x**2 - 18*x + 13
371      x = x - f / df
372      WRITE(*,*)x, f
373      IF (ABS(f).GE.maxHata) yakinsadiMi = .TRUE.
374      END DO
375
376      c iterasyon bitiyor ve sonuclar goruntuleniyor
377
378      IF (yakinsadiMi.EQV..TRUE.) WRITE(*,*)'TAMAM : Yakinsadi !..'
379      IF (yakinsadiMi.EQV..FALSE.) WRITE(*,*)'OLMADI : Yakinsamadi !..'
380
381      c bitti
382
383      END
384
-----
385
386      Yukaridaki program sadece bir kok bulur -verilen ilk degerden
387      iterasyona baslayarak- ve durur. Tum reel koklerin bulunmasi isteniyor
388      ise yukaridaki program asagidaki bicimde degistirilebilir. ABS(x)
389      fonksiyonu x' in mutlak degerini alır. Cikti ve kod asagida verilmistir :
390
391      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./newtonMethodIV
392      f(X) = AX^3 + BX^2 + CX + D = 0 icin katsayilar :
393      1 -6 11 -6
394      1. kok : 3.
395      2. kok : 1.
396      3. kok : 2.
397      ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
398
-----
399      PROGRAM newtonMethodIV
400      IMPLICIT NONE
401      INTEGER i, j
402      REAL x, maxHata, f, df, kok(3)
403      REAL a, b, c, d, da, db, dc
404      LOGICAL yakinsadiMi
405
406      c sonucun sahip olmasini istedigimiz en buyuk hata
407
408      PARAMETER (maxHata = 1e-6)
409      x=1000
410
411      c fonksiyonun katsayilari okunuyor, turev katsayilari hesaplaniyor
412
413      WRITE(*,*)'f(X) = AX^3 + BX^2 + CX + D = 0 icin katsayilar : '
414      READ(*,*)a, b, c, d
415      da = 3*a
416      db = 2*b
417      dc = c
418
419      c iterasyona baslaniyor f : fonksiyon ve df : f'in turevi
420

```

```

421     DO j=1,3
422         yakinsadiMi = .FALSE.
423         DO i=0,200
424             f = a*x**3 + b*x**2 + c*x + d
425             df = da*x**2 + db*x + dc
426             x = x - f / df
427             IF (ABS(f).GE.maxHata) yakinsadiMi = .TRUE.
428         END DO
429         IF (yakinsadiMi.EQV..TRUE.) THEN
430             kok(j)=x
431             WRITE(*,*)j, '. kok : ',kok(j)
432         END IF
433         x=-1000
434         IF (j.EQ.2) x=(kok(1)+kok(2))/2+0.001
435     END DO
436
437 c bitti
438
439     END

```

Yukaridaki programda iterasyona 1000 ve hemen ardindan -1000 degerlerinden baslanmaktadır bu sayede koke x uzerinde once sagdan ve sonra soldan yaklasilmektedir. Ucuncu kokun bulunmasi icin gerekli ilk deger bulunan bu iki kokun ortalamasi olarak alinmakta ve ucuncu kok bulunmektedir. Ornek olarak su fonksiyon kullanilmistir : $(x-1)(x-2)(x-3)=0$ yani $x^3-6x^2+11x-6$

8 - Herbir seri elemaninin **FUNCTION** tipi alt programda hesaplanmasinin sagladigi kolayligi gostermek icin asagidaki **program** veilmistir. Cikti ve kod asagidaki gibidir :

```

452
453 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./seriAcilimi
454 g(x) = 1 + 1/x^2 + 1/x^4 + 1/x^6 + ... + 1/x^2n
455 bicimindeki toplamın x ve n girilerek hesaplanması
456 x ve n giriniz :
457 2
458 5
459 g= 1.33300781
460 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./seriAcilimi
461 g(x) = 1 + 1/x^2 + 1/x^4 + 1/x^6 + ... + 1/x^2n
462 bicimindeki toplamın x ve n girilerek hesaplanması
463 x ve n giriniz :
464 2
465 10
466 g= 1.33333302
467 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>

```

```

468
469 PROGRAM seriAcilimi
470 IMPLICIT NONE
471 INTEGER i, n
472 REAL*8 x, g, f
473 PRINT*, 'g(x) = 1 + 1/x^2 + 1/x^4 + 1/x^6 + ... + 1/x^2n'
474 PRINT*, 'bicimindeki toplamın x ve n girilerek hesaplanması'
475 PRINT*, 'x ve n giriniz : '
476 READ(*,*)x, n
477 g = 1
478 DO i=1,n
479     g = g + f(x,i)
480 END DO
481 PRINT*, 'g=',g
482 END
483
484 REAL*8 FUNCTION f(x,i)
485 IMPLICIT NONE
486 REAL*8 x
487 INTEGER i
488 f = 1/x**(2*i)
489 END

```

Yukaridaki programda **REAL** yerine **REAL*8** kullanilmistir ve x, f isimli **FUNCTION** tipi alt **program** ile bu alt **program** icinde kullanılan x degiskenleri de **REAL*8** tanimlanmistir. Hassasligi artirmak icin bu yapilmistir. Bir **FUNCTION** ana programda **REAL*8** tanimlandiginda ya da yukaridaki ornekte oldugu gibi aldigi parametre **REAL*8** tipinde oldugunda **FUNCTION** tipi alt **program** tanimlanirken de bu tipler kullanilmalidir aksi halde **program** derlenemeyectir.

9 - Asagidaki **program** $k!/x**k$ ($k=0..n$) seri toplamini hesaplamak icin yazilmistir. Alt programlar birbirleri icinden cagirilabilirler ve **FUNCTION** tipi alt programlar sadecekendilerini cagiran ana ve/veya diger alt programların icinde sanki birer degiskenlermis gibi tanimlanmalidirlar. Cikti ve kod asagidaki gibidir :

```

505
506 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./zorSeri
507 k!/x**k (k=0..n) seri toplami icin x ve n girin :
508 2
509 5
510 Sonuc = 8.
511 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./zorSeri
512 k!/x**k (k=0..n) seri toplami icin x ve n girin :
513 2
514 10
515 Sonuc = 4468.625
516 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari> ./zorSeri
517 k!/x**k (k=0..n) seri toplami icin x ve n girin :
518 2
519 20
520 Sonuc = 539134.062
521 ozgur@olmak:~/Documents/g77/kaynakKodlar/ornekler/odevCevaplari>
522
523 PROGRAM zorSeri
524 IMPLICIT NONE
525 INTEGER n
526 REAL seriyiTopla, x
527 WRITE(*,*)'k!/x**k (k=0..n) seri toplami icin x ve n girin : '
528 READ(*,*)x,n
529 WRITE(*,*)'Sonuc =',seriyiTopla(x,n)
530 END
531
532 REAL FUNCTION seriyiTopla(x,n)
533 IMPLICIT NONE
534 INTEGER n, k, faktoriyel
535 REAL x
536 seriyiTopla = 0
537 DO k=0,n
538   seriyiTopla = seriyiTopla + faktoriyel(k)/x**(k)
539 END DO
540 END
541
542 INTEGER FUNCTION faktoriyel(a)
543 IMPLICIT NONE
544 INTEGER a, i
545 faktoriyel = 1
546 DO i=2,a
547   faktoriyel = faktoriyel * i
548 END DO
549 END
550
551
552 Alt program kullaniminin ana programi ne kadar kisalttigina ve hata
553 denetimini ne kadar kolaylastirdigina dikkat ediniz.

```