

# مقدمات فرترن برای محاسبات عددی

ریحانه دانش دوست

# FORTRAN

FORTRAN برگرفته شده از عبارت FORMula TRANslation است. فرترن اولین زبان سطح بالا است.

اگرچه نرم افزارهایی مانند "متلب"، امکانات گرافیکی خوبی دارند اما سرعت انجام محاسباتشان کم است؛ هنگامی که حجم محاسبات بالا است از زبان هایی مانند فرترن یا C++ استفاده می شود که کار با فرترن ساده تر است. علت استفاده از فرترن، سادگی کار با آرایه ها مخصوصا آرایه های چند بعدی است.

# ابزار لازم

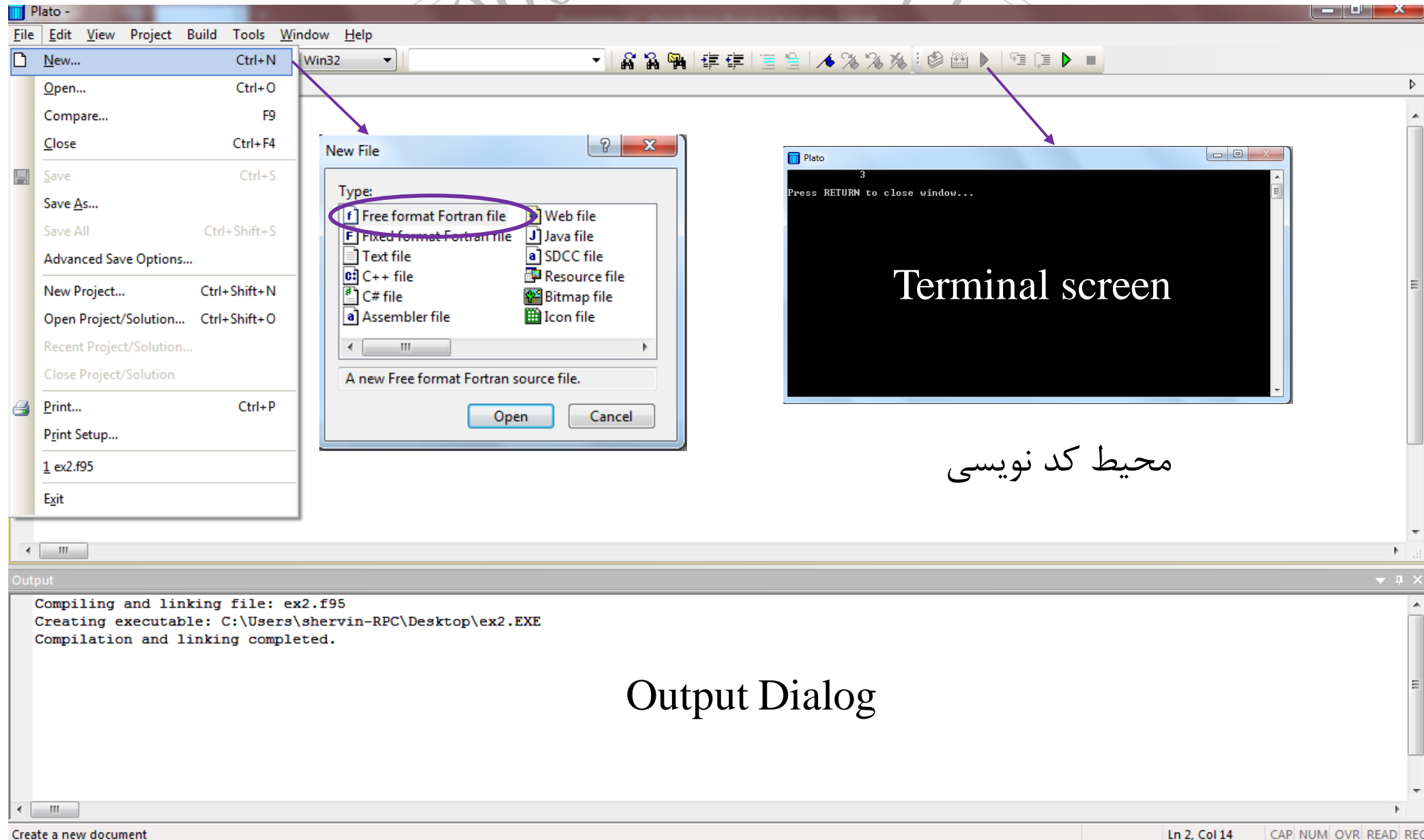
محیط برنامه نویسی

محیط برنامه نویسی	حجم اشغال کننده
Visual Studio	۳۰ گیگابایت
Plato	۲۰۰ مگابایت

کامپایلر

کامپایلرهای رایگان	کامپایلرهای تجاری
GFortran	SilverFrost FTN95
LFortran	Intel

# محیط Plato



Output Dialog

# نامگذاری متغیرها

- در نامگذاری از حروف a-z ، A-Z ، اعداد 0-9 و \_ می توان استفاده کرد.
- نام متغیر نباید بیشتر از ۳۱ کاراکتر باشد.
- نام متغیر حتما باید با حرف شروع شود.
- فرترن بین حروف بزرگ و کوچک فرقی قائل نیست.
- بین حروف نباید فاصله باشد مگر برای جداسازی دو کلمه کلیدی مجزا.
- از کلمات کلیدی فرترن مانند if،do،while و ... نباید به عنوان نام متغیر استفاده شود.

# انواع متغیرها

دستور	مثال	عملکرد
integer :: var	integer :: i,j	متغیرهای a و j را به صورت اعداد صحیح ذخیره می کند.
real :: var	real :: x,y	متغیرهای x و y را به صورت اعداد حقیقی تا ۵ رقم بعد از اعشار ذخیره می کند.
double precision :: var	double precision :: b	متغیر b را به صورت عدد حقیقی با تعداد رقم اعشار بیشتر (۱۱ رقم بعد از اعشار)، ذخیره می کند.
complex :: var	complex :: a	متغیر a را به صورت عدد مختلط ذخیره می کند.
logical :: var	Logical :: on,off	متغیرهای on و off را به صورت مقادیر منطقی، ذخیره می کند.
character :: var	Character :: name	متغیر name را به صورت حرف با طول ۱، ذخیره می کند. (فقط حرف اول متغیر را ذخیره می کند).
	Character (Len=2) :: name	متغیر name را به صورت رشته ای از حروف با طول ۲، ذخیره می کند. (فقط حروف m و a را ذخیره می کند).
	Character (Len=*),parameter :: name='me'	متغیر name را به صورت رشته ای از حروف با طول ثابت براساس طول کاراکتر وارد شده، ذخیره می کند. (طول کاراکتر وارد شده در اینجا، ۲ است).
	Character :: name*2	متغیر name را به صورت رشته ای از حروف با طول ۲، ذخیره می کند.
	Character,dimension(2) :: name Character :: name(2)	متغیر name را به صورت رشته ای از حروف با طول ۱ و یک بعد با ۲ المان، ذخیره می کند.

parameter-type,parameter :: para=const

مثال:

real,parameter :: pi=3.14

## پارامترها (ثوابت)

متغیرهایی هستند که مقدارشان در کل برنامه ثابت است.

# شروع و پایان برنامه

متغیر	نحوه مقداردهی
integer	var = initial value
real	var = initial value
double precision	var = initial value
complex	var = (value1,value2)
logical	var = .true. var = .false.
character	var = 'initial value'

program program-name

implicit none

declaration of variables

.

.

.

end program program-name

دستور implicit none

فرتن به صورت پیش فرض متغیرهایی که با حروف i, j, k, l, m و n شروع می شوند را از نوع integer و بقیه متغیرهایی که با حروفی غیر از این حروف (حروف a-h و o-z) شروع می شوند را از نوع real در نظر می گیرد. با این دستور، این پیش فرض ها از بین رفته و طبق آنچه که ما تعریف کنیم پیش می رود.

# ورودی و خروجی برنامه

دستور	نحوه فراخوانی	عملکرد
read (*,*) var	read (*,*) a	مقدار متغیر a را از صفحه ترمینال می گیرد.
read (unit,fmt) var	read (1,*) a	مقدار متغیر a را از فایل شماره ۱ با فرمت آزاد می گیرد.
write (*,*) var	write (*,*) a	مقدار متغیر a را در صفحه ترمینال با فرمت آزاد چاپ می کند.
write (*,*) 'var'	write (*,*) 'a'	عبارت a را در صفحه ترمینال با فرمت آزاد چاپ می کند.
write (unit,fmt) var	write (1,*) a	مقدار متغیر a را در فایل شماره ۱ با فرمت آزاد چاپ می کند.
write (unit,fmt) 'var'	write (1,*) 'a'	عبارت a را در فایل شماره ۱ با فرمت آزاد چاپ می کند.
print *,var	print *,a	مقدار متغیر a را در پرینتر با فرمت آزاد چاپ می کند.
print *,'var'	print *,'a'	عبارت a را در پرینتر با فرمت آزاد چاپ می کند.
print fmt,var	print 10,a	مقدار متغیر a را در پرینتر با فرمت ۱۰، چاپ می کند.
print fmt,'var'	print 10,'a'	عبارت a را در پرینتر با فرمت 10، چاپ می کند.

ورودی

خروجی

Open (unit,file= 'file-name')  
Close(unit)

مثال:

Open(1,file='inputs.txt')  
Close(1)

برنامه با دستور open، فایل شماره ۱ با اسم inputs.txt را باز می کند و با دستور close، فایل را می بندد.



# دستور فرمت (format)

I: متغیر صحیح  
w: تعداد کل ارقام  
m: حداقل تعداد ارقامی که باید نشان داده شود.  
E: نمایش علمی عدد اعشار  
d: تعداد ارقام بعد از ممیز (دقت)  
e: تعداد ارقام توان  
F: متغیر اعشار با اعشار مشخص  
G: مانند F، به دقت بستگی دارد.  
L: متغیر منطقی  
A: متغیر کاراکتری

فرمت	متغیر
Iw, Iw.m	integer
Ew.d , Ew.dEe, Fw.d, Gw.dEe	real
Lw	Logical
A, Aw	character

کاربرد	کاراکتر
انتقال به خط بعدی	/
یک رشته کاراکتری را نشان می دهد.	'any string'
n خانه به سمت راست می برد.	nx

خروجی	عدد	فرمت
***	3456	I3
345	345	I3
□34	34	I3
233	233	I3.3
□15.50	15.5	F6.2
*****	2115.25	F6.2
□0.121E+03	121.454	E10.3
□□T	.true.	L3
star	star	A
sta	star	A3
□star	star	A5

# آرایه ها

نحوه عملکرد	مثال	دستور
متغیر <b>b</b> را به صورت یک آرایه ۱ بعدی (بردار) با ۵ المان در نظر می گیرد که کران پایین آن به صورت پیش فرض ۱ و کران بالای آن ۵ است.	<code>integer :: b(5)</code>	<code>array-type :: array()</code>
متغیر <b>b</b> را به صورت یک آرایه ۲ بعدی (ماتریس) در نظر می گیرد که تعداد المان های بعد اول (سطر)، ۵ و تعداد المان های بعد دوم (ستون)، ۳ است.	<code>integer :: b(5,3)</code>	<code>array-type :: array(,)</code>
متغیر <b>b</b> را به صورت یک آرایه ۱ بعدی با <b>shape</b> نامشخص در نظر می گیرد.	<code>allocatable :: b(:)</code>	<code>allocatable :: array(deferred-shape)</code>

مفهوم	ترم
تعداد ابعاد است.	rank
کران بالا و پایین است که کران پایین آن به صورت پیش فرض، ۱ است.	bounds
تعداد المان ها در هر بعد است.	extent
تعداد کل المان ها است.	size
پرانتز و عبارت داخل آن همان <b>shape</b> است.	shape

نحوه عملکرد	مثال	دستور
مقدار المان های ۵۰ تا ۱۰۰ آرایه <b>b</b> را برابر با ۳ می گیرد.	<code>real :: b(50:100) =3</code>	<code>array-type :: array(:)= value</code>
مقدار المان های ۵۰ تا ۱۰۰ آرایه <b>b</b> را با گام ۲ برابر با ۳ می گیرد.	<code>real :: b(50:100:2) =3</code>	<code>array-type :: array(::)= value</code>
مقدار المان های ۱ تا ۶ در بعد اول و مقدار المان های ۱ تا ۸ در بعد دوم آرایه <b>b</b> را برابر با ۳ می گیرد.	<code>integer :: b(1:6,1:8) =3</code>	<code>array-type :: array(:,)= value</code>
مقدار المان های ۱ تا ۶ در بعد اول با گام ۲ و تعداد المان های ۱ تا ۸ در بعد دوم با گام ۲ آرایه <b>b</b> را برابر با ۳ می گیرد.	<code>integer :: b(1:6:2,1:8:2) =3</code>	<code>array-type :: array(:,,:)= value</code>
مقدار ۲ را به آرایه <b>b</b> اختصاص داده و <b>b</b> را یک آرایه یک بعدی با دو المان در نظر می گیرد.	<code>allocate (b(2))</code>	<code>allocate (array(value))</code>

# نمایش ماتریس

```
program main
implicit none
integer :: i,j
integer :: ncmx,nout,nrmax,nrow,ntype,ncol
parameter(nrmax=3,nout=4,ncmx=3)
double precision :: a(nrmax,ncmx),dj
double precision :: ten
parameter(ten=10.0d+00)
character :: title*20
logical :: fileit
parameter(fileit=.false.)
external dsplay
intrinsic dble
do j=1,ncmx
    dj=dble(j)
    do i=1,nrmax
        a(i,j)=ten*dble(i)+dj
    end do
end do
ncol=3
nrow=3
title= 'i format'
ntype=1
Call dsplay(ncmx,ncol,nout,nrmax,nrow,ntype,a,title,fileit)
end program main
```

i format

3 3

11	12	13
21	22	23
31	32	33

2

Rows 1 to 3, Columns 1 to 3

ccyyymmdd=20210925, hhmmss=170001

# تولید اعداد تصادفی

دستور	نحوه فراخوانی	عملکرد
random_number()	call random_number(a)	تولید یک عدد یا مجموعه ای از اعداد تصادفی. a می تواند یک عدد یا یک آرایه باشد.
random_seed()	call random_seed(size=n)	تولید مجدد یک عدد یا مجموعه ای از اعداد با استفاده از n عدد صحیح.
	call random_seed(put=seed)	تولید مجدد یک عدد یا مجموعه ای از اعداد با مقدار اولیه دادن به seed.
	call random_seed(get=seed)	تولید مجدد یک عدد یا مجموعه ای از اعداد با گرفتن مقدار فعلی seed.

# توابع کاربردی

دستور	نحوه فراخوانی	عملکرد
matmul( , )	matmul(a,b)	دو ماتریس a و b را در هم ضرب می کند.
dot_product( , )	dot_product(a,b)	دو بردار a و b را در هم ضرب داخلی می کند.
transpose()	transpose(a)	تشکیل ترانپاده ی ماتریس a.
maxval()	maxval(v) maxval(a) maxval(a,dim=1) maxval(a,dim=2) maxval(v,mask=a. LT .n) maxval (a,mask=a. LT .n)	بزرگترین مولفه بردار v را نمایش می دهد. بزرگترین مولفه ماتریس a را نمایش می دهد. بزرگترین مولفه در هر ستون ماتریس a را نمایش می دهد. بزرگترین مولفه در هر سطر ماتریس a را نمایش می دهد. بزرگترین مولفه بردار v که کوچک تر از عدد n است را نمایش می دهد. بزرگترین مولفه ماتریس a که کوچک تر از عدد n است را نمایش می دهد.
minval()	minval(v) minval(a) minval(a,dim=1) minval(a,dim=2) minval(v,mask=a. LT .n) minval (a,mask=a. LT .n)	کوچکترین مولفه بردار v را نمایش می دهد. کوچکترین مولفه ماتریس a را نمایش می دهد. کوچکترین مولفه در هر ستون ماتریس a را نمایش می دهد. کوچکترین مولفه در هر سطر ماتریس a را نمایش می دهد. کوچکترین مولفه بردار v که کوچک تر از عدد n است را نمایش می دهد. کوچکترین مولفه ماتریس a را که کوچک تر از عدد n است را نمایش می دهد.
maxloc()	maxval(v) maxval(a) maxval(a,dim=1) maxval(a,dim=2) maxval(v,mask=a. LT .n) maxval (a,mask=a. LT .n)	موقعیت بزرگترین مولفه بردار v را نمایش می دهد. موقعیت بزرگترین مولفه ماتریس a را نمایش می دهد. موقعیت بزرگترین مولفه در هر ستون ماتریس a را نمایش می دهد. موقعیت بزرگترین مولفه در هر سطر ماتریس a را نمایش می دهد. موقعیت بزرگترین مولفه بردار v که کوچک تر از عدد n است را نمایش می دهد. موقعیت بزرگترین مولفه ماتریس a را که کوچک تر از عدد n است را نمایش می دهد.
minloc()	maxval(v) maxval(a) maxval(a,dim=1) maxval(a,dim=2) maxval(v,mask=a. LT .n) maxval (a,mask=a. LT .n)	موقعیت کوچکترین مولفه بردار v را نمایش می دهد. موقعیت کوچکترین مولفه ماتریس a را نمایش می دهد. موقعیت کوچکترین مولفه در هر ستون ماتریس a را نمایش می دهد. موقعیت کوچکترین مولفه در هر سطر ماتریس a را نمایش می دهد. موقعیت کوچکترین مولفه بردار v که کوچک تر از عدد n است را نمایش می دهد. موقعیت کوچکترین مولفه ماتریس a را که کوچک تر از عدد n است را نمایش می دهد.

# توابع کاربردی

دستور	نحوه فراخوانی	عملکرد
sum()	sum(v) sum(a) sum(a,dim=1) sum(a,dim=2) sum(a,mask=a.LT.n)	مجموع تمام مولفه های بردار v را نشان می دهد. مجموع تمام مولفه های ماتریس a را نشان می دهد. مجموع تمام مولفه های هر ستون ماتریس a را نشان می دهد. مجموع تمام مولفه های هر سطر ماتریس a را نشان می دهد. مجموع تمام مولفه های کوچک تر از عدد n در ماتریس a را نشان می دهد.
lbound()	lbound(a) lbound(a,dim=1) lbound(a,dim=2)	کران پایین ماتریس a را نمایش می دهد. کران پایین هر ستون ماتریس a را نشان می دهد. کران پایین هر سطر ماتریس a را نشان می دهد.
ubound()	ubound(a) ubound(a,dim=1) ubound(a,dim=2)	کران بالای ماتریس a را نمایش می دهد. کران بالای هر ستون ماتریس a را نشان می دهد. کران بالای هر سطر ماتریس a را نشان می دهد.
max()	max(n1,n2,...)	ماکزیمم مجموعه ای از اعداد را نمایش می دهد.
min()	min(n1,n2,...)	مینیموم مجموعه ای از اعداد را نمایش می دهد.
huge()	huge(x)	بزرگترین مقدار مثبتی که کامپایلر می تواند داشته باشد را نشان می دهد.
epsilon()	epsilon(x)	کوچکترین مقدار مثبتی که کامپایلر می تواند داشته باشد را نشان می دهد.

# توابع و عملگرهای ریاضی

عملکرد	نحوه فراخوانی	دستور
مقدار دقیقی یا همان قدرمطلق $X$ را نشان می دهد.	<code>abs(x)</code>	<code>abs()</code>
مقدار باقی مانده تقسیم $X$ بر $y$ نمایش می دهد.	<code>mod(x,y)</code>	<code>mod( , )</code>
مقدار جذر $X$ را نمایش می دهد.	<code>sqrt(x)</code>	<code>sqrt()</code>
کوچک ترین جزء بزرگتر یا مساوی $X$ را نمایش می دهد.	<code>ceiling(x)</code>	<code>ceiling()</code>
بزرگترین جزء صحیح کمتر یا مساوی $X$ را نمایش می دهد.	<code>floor(x)</code>	<code>floor()</code>
جزء کسری $X$ را نمایش می دهد.	<code>fraction(x)</code>	<code>fraction()</code>
جزء صحیح $X$ را نمایش می دهد.	<code>int(x)</code>	<code>int()</code>
نزدیک مقدار به $X$ را نمایش می دهد.	<code>nint(x)</code>	<code>nint()</code>
مقدار نمایی $X$ را نمایش می دهد.	<code>exp(x)</code>	<code>exp()</code>
مقدار لگاریتم طبیعی $X$ را نمایش می دهد.	<code>log(x)</code>	<code>log()</code>
عدد $X$ را به یک عدد مختلط تبدیل می کند. دو عدد $X$ و $y$ را به اعداد مختلط تبدیل می کند.	<code>cmplx(x)</code> <code>cmplx(x,y)</code>	<code>cmplx()</code>
مقدار سینوس $X$ را بر حسب رادیان نشان می دهد.	<code>sin(x)</code>	<code>sin()</code>
مقدار کسینوس $X$ را بر حسب رادیان نشان می دهد.	<code>cos(x)</code>	<code>cos()</code>
مقدار تانژانت $X$ را بر حسب رادیان نشان می دهد.	<code>tan(x)</code>	<code>tan()</code>
مقدار سینوس هایپربولیک $X$ را بر حسب رادیان نشان می دهد.	<code>sinh(x)</code>	<code>sinh()</code>
مقدار کسینوس هایپربولیک $X$ را بر حسب رادیان نشان می دهد.	<code>cosh(x)</code>	<code>cosh()</code>
مقدار تانژانت هایپربولیک $X$ را بر حسب رادیان نشان می دهد.	<code>tanh(x)</code>	<code>tanh()</code>

عملکرد	نحوه فراخوانی	دستور
مقدار سینوس وارون $X$ را بر حسب رادیان نشان می دهد.	<code>asin(x)</code>	<code>asin()</code>
مقدار کسینوس وارون $X$ را بر حسب رادیان نشان می دهد.	<code>acos(x)</code>	<code>acos()</code>
مقدار اتنژانت وارون $X$ را بر حسب رادیان نشان می دهد.	<code>atan(x)</code>	<code>atan()</code>

کاربرد	عملگر
توان	<code>**</code>
ضرب	<code>*</code>
تقسیم	<code>/</code>
جمع	<code>+</code>
تفریق	<code>-</code>

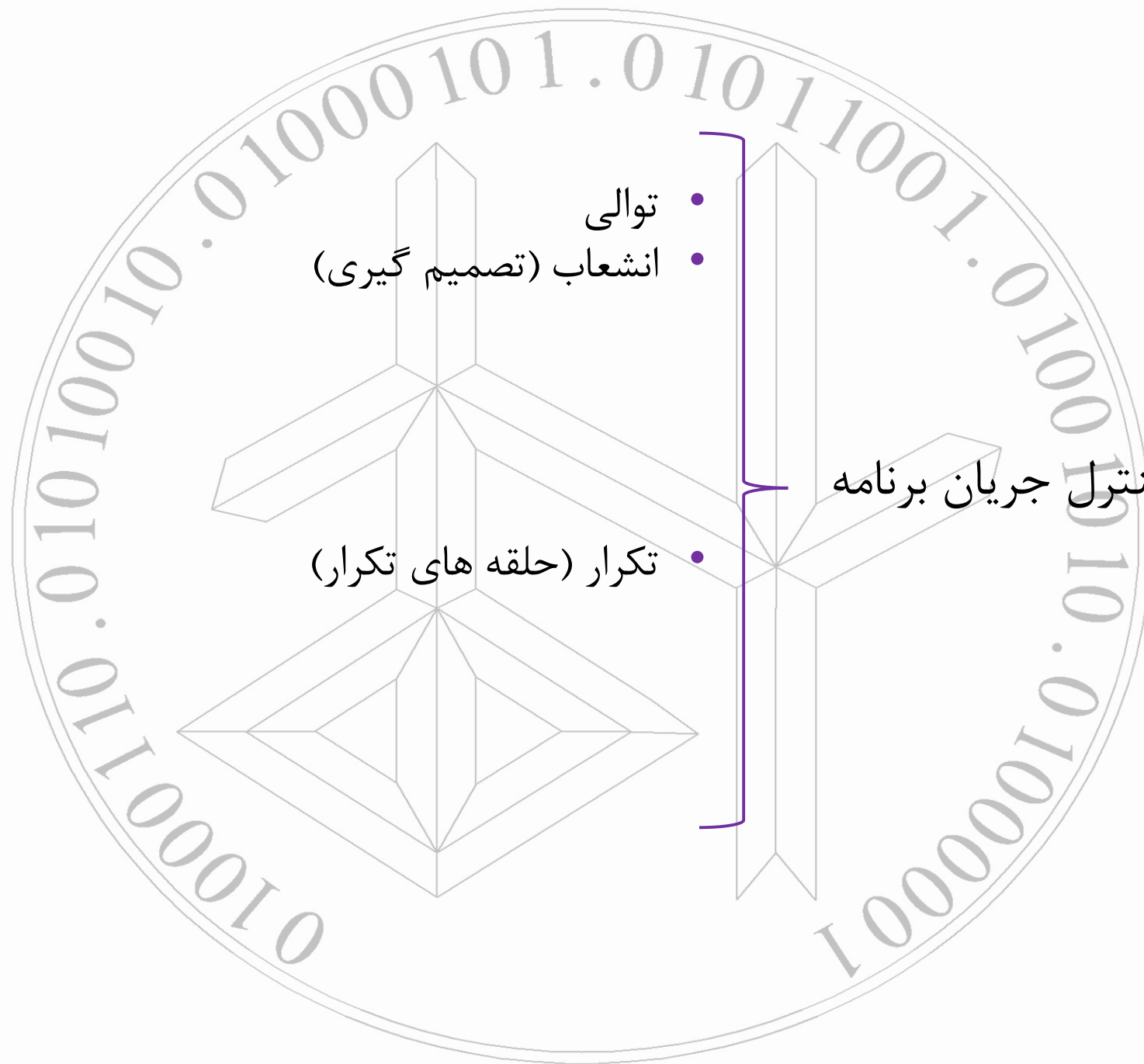
# تعريف تابع

تعريف تابع خارج از برنامه اصلي	تعريف تابع داخل برنامه اصلي
<code>Program</code> program_name	<code>Program</code> program_name
<code>Implicit none</code>	<code>Implicit none</code>
Declarations of variables	Declarations of variables
<code>End program</code> program_name	<code>Contains</code>
Output_type, <code>function</code> function_name (inputs)	Output_type, <code>function</code> function_name (inputs)
Declarations of inputs	Declarations of inputs
Definition of function	Definition of function
<code>End function</code> function_name	<code>End function</code> function_name
	<code>End program</code> program-name



# تعریف سابروتین (زیر برنامه)

تعریف سابروتین خارج از برنامه اصلی	تعریف سابروتین داخل برنامه اصلی
<code>Program</code> program_name	<code>Program</code> program_name
<code>Implicit none</code>	<code>Implicit none</code>
Declarations of variables	Declarations of variables
<code>Call</code> subroutine_name (inputs,outputs)	<code>Call</code> subroutine_name (inputs,outputs)
<code>End program</code> program_name	<code>Contains</code>
<code>subroutine</code> subroutine_name (inputs)	<code>subroutine</code> subroutine_name (inputs)
Declarations of inputs & outputs	Declarations of inputs & outputs
Definition of subroutine	Definition of subroutine
<code>End subroutine</code> subroutine_name	<code>End subroutine</code> subroutine_name
	<code>End program</code> program-name



- توالی
- انشعاب (تصمیم گیری)

- تکرار (حلقه های تکرار)

ساختارهای کنترل جریان برنامه

# عملگرهای منطقی و رابطه ای

A.and.B		
ورودی		خروجی
T	T	T
T	F	F
F	T	F
F	F	F

A.or.B		
ورودی		خروجی
T	T	T
T	F	T
F	T	T
F	F	F

A.eqv.B		
ورودی		خروجی
T	T	T
T	F	F
F	T	F
F	F	T

A.neqv.B		
ورودی		خروجی
T	T	F
T	F	T
F	T	T
F	F	F

.not.A	
ورودی	خروجی
T	F
F	T

عملگر رابطه ای		مفهوم
==	.EQ.	مساوی
/=	.NE.	نامساوی
<	.LT.	کوچکتر
<=	.LE.	کوچکتر مساوی
>	.GT.	بزرگتر
>=	.GE.	بزرگتر مساوی

# if

```
if (cond.) then
```

```
    statements if condition was true
```

```
end if
```

```
if (cond.) then
```

```
    statements if condition was true
```

```
else
```

```
    statements if condition was false
```

```
end if
```

```
if (cond.1) then
```

```
    statements 1
```

```
elseif (cond.2) then
```

```
    statements 2
```

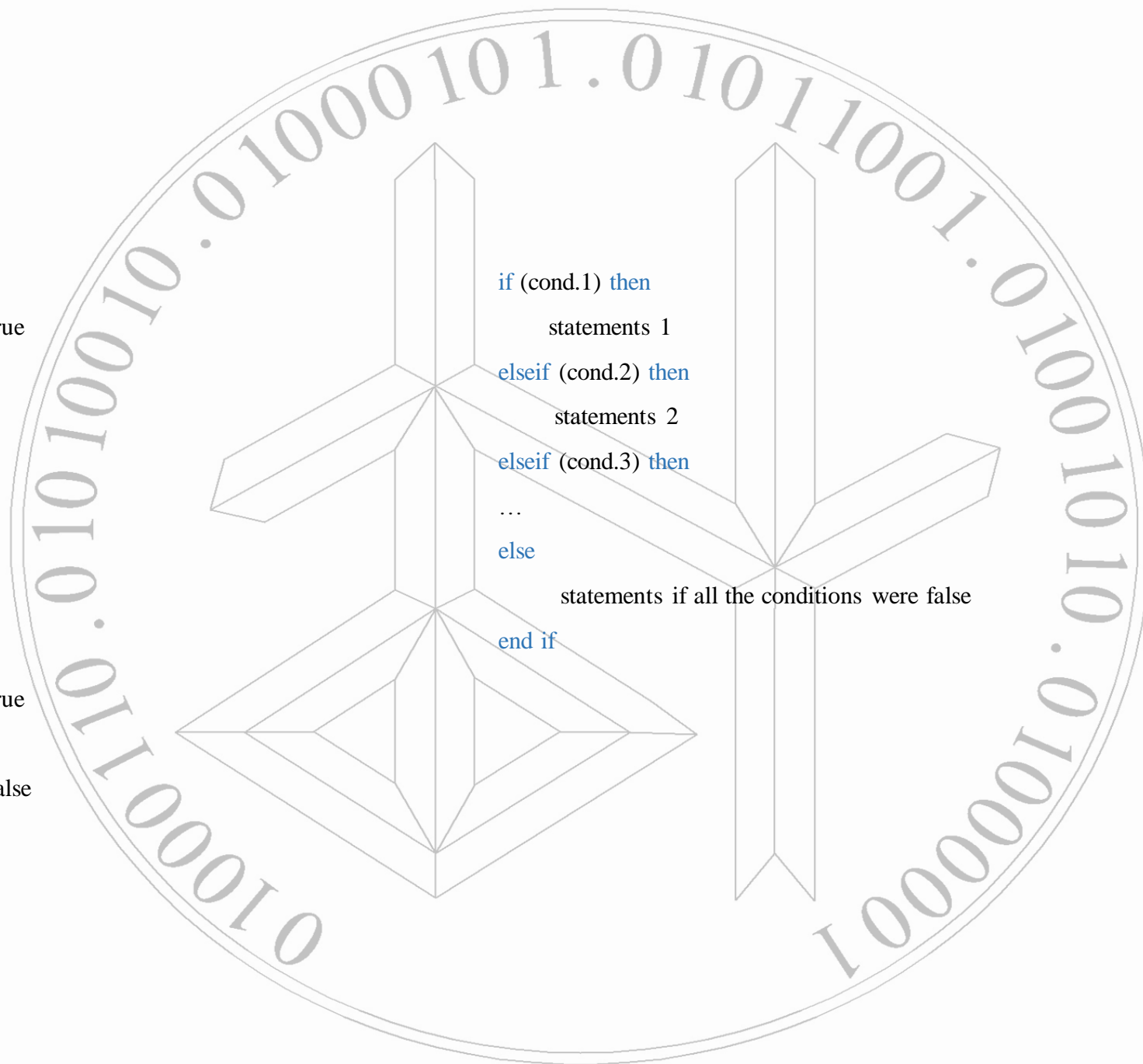
```
elseif (cond.3) then
```

```
...
```

```
else
```

```
    statements if all the conditions were false
```

```
end if
```



# select case

```
select case (variable_name)
```

```
  case (cond.1)
```

```
    statements1
```

```
  case (cond.2)
```

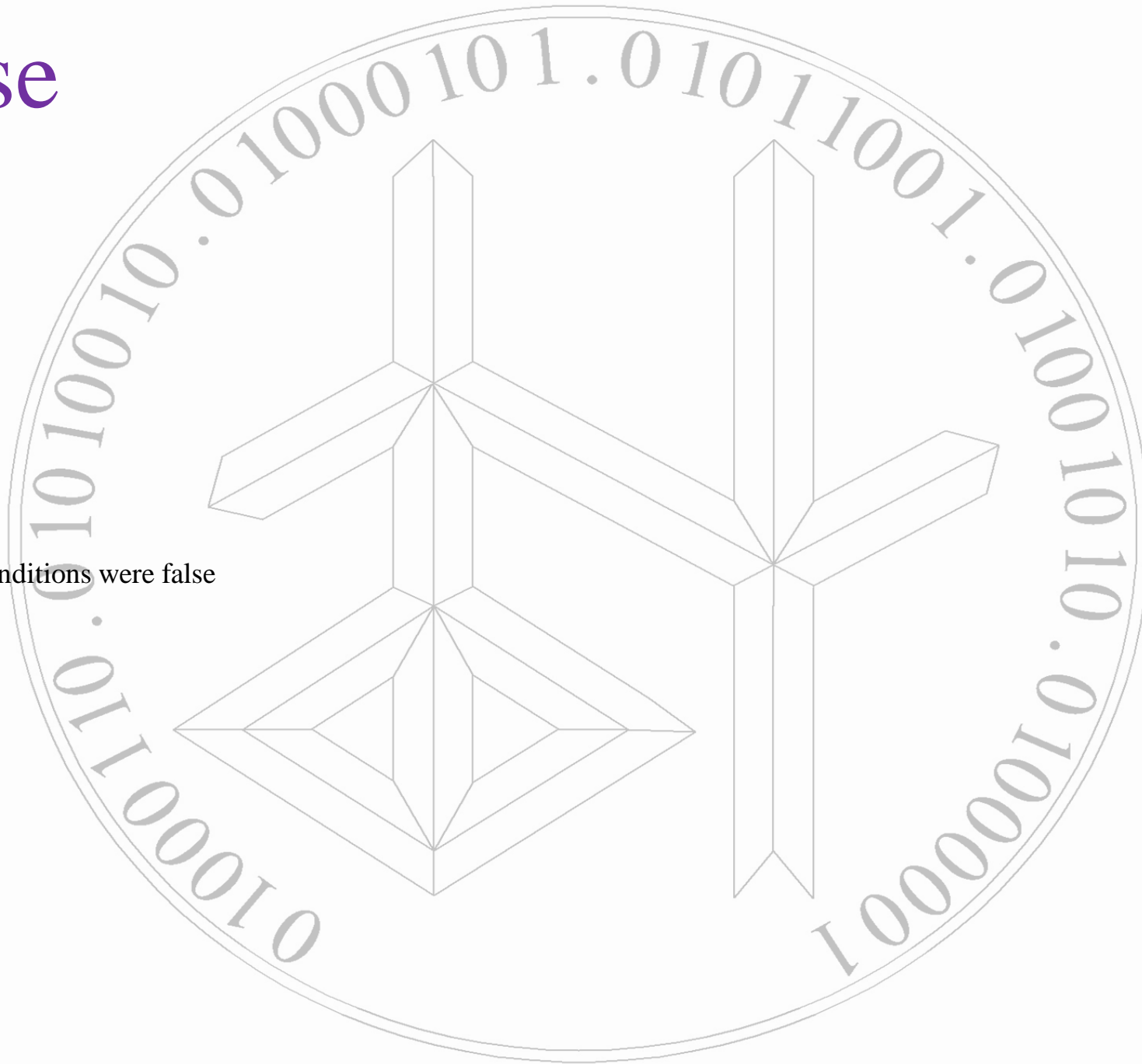
```
    statements 2
```

```
  ...
```

```
Case default
```

```
  statements if all the conditions were false
```

```
end select
```

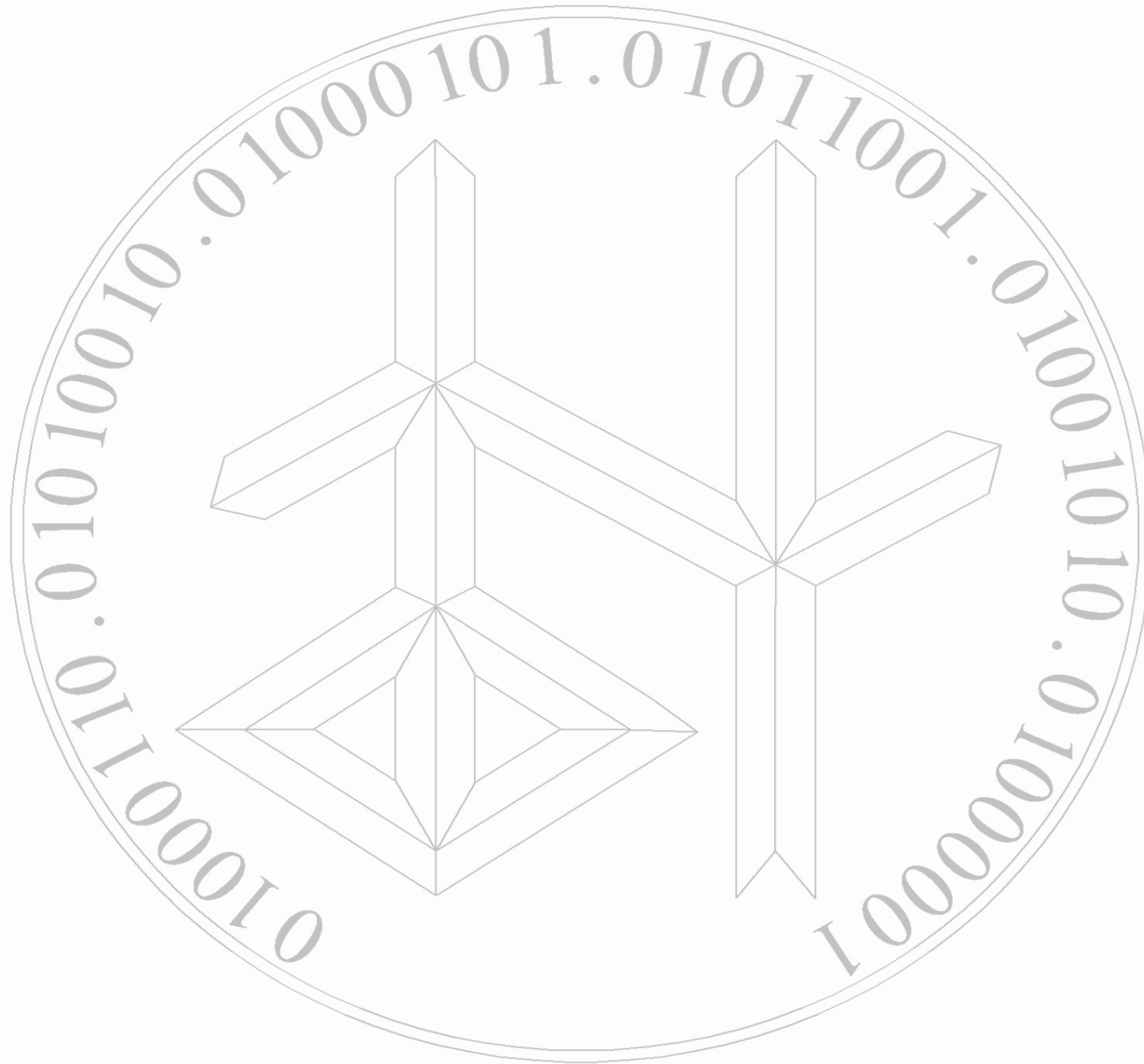


# do while

```
do while (condition)
```

```
    statements
```

```
end do
```

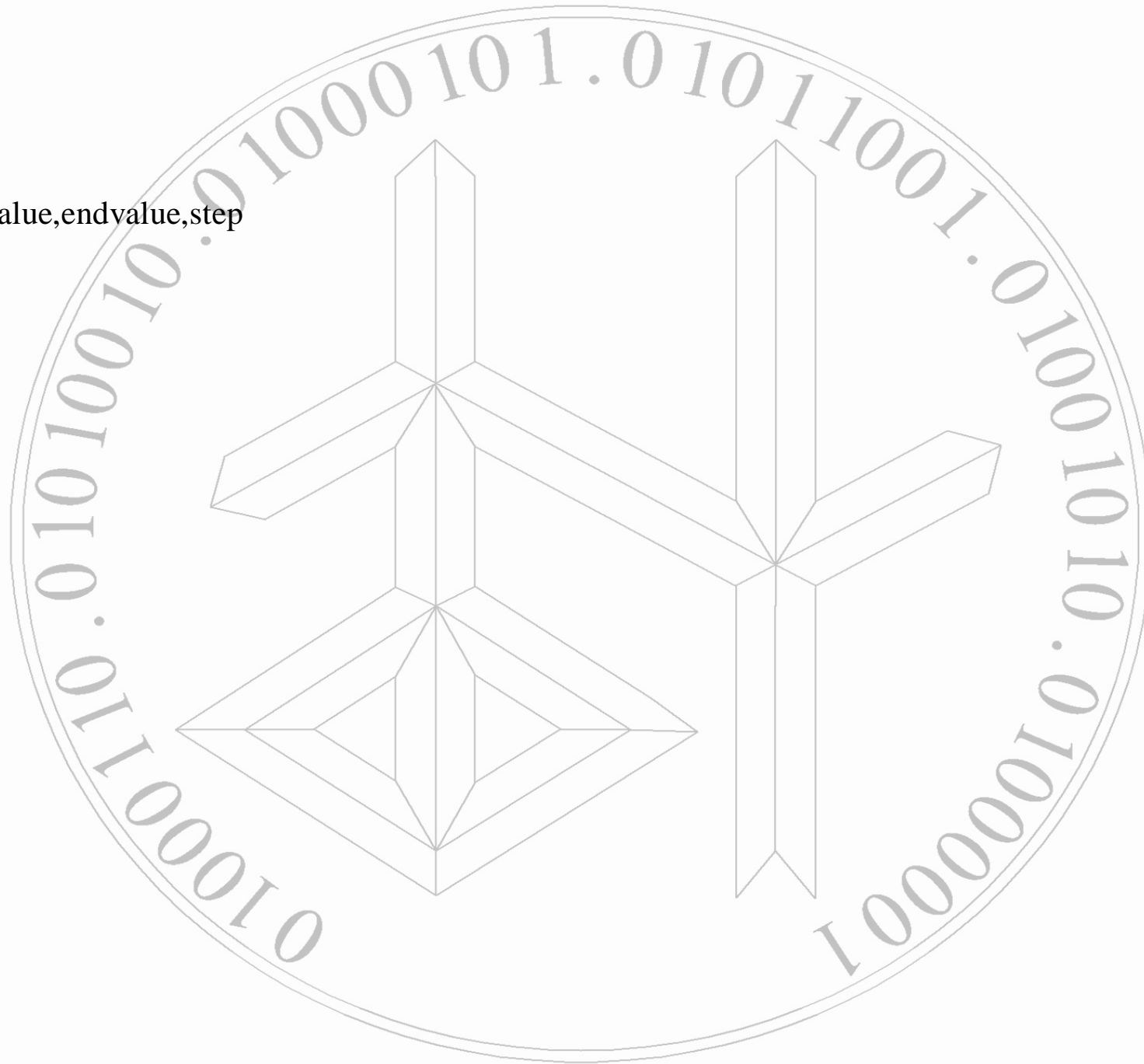


# do

```
do countervariable = initialvalue,endvalue,step
```

```
    statements
```

```
end do
```



# رسم نمودار

سابروتین	عملکرد
gks001	رسم یک نمودار x,y
gks004	رسم چهار نمودار x,y
rtplot	رسم نمودار منحنی پارامتری $r=r(\theta)$
space0	رسم نمودار منحنی ها در فضا
surd2s	رسم نمودار سطوح و کانتورها



# رسم نمودار $y=f(x)$

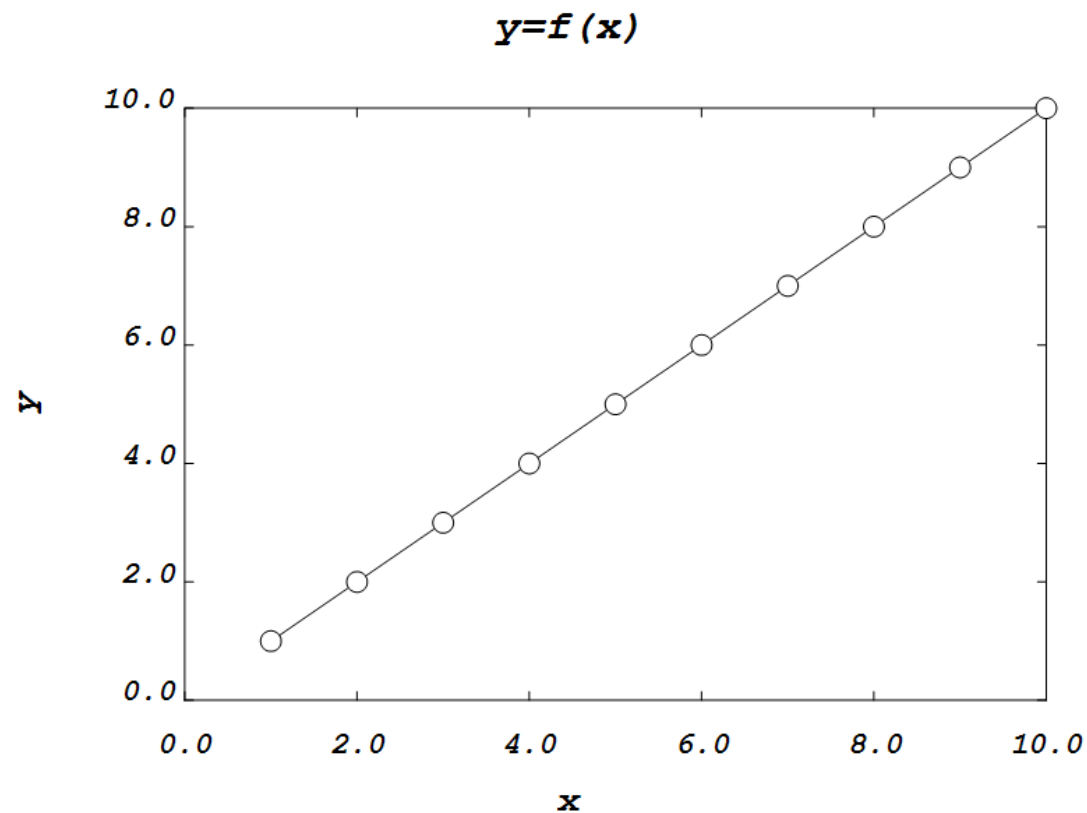
```
program main1
implicit none
integer:: i,l,m,n,p
parameter(p=10)
double precision:: x(p),y(p)
character:: ptitle*8,xtitle*1,ytitle*1
external gks001

l=1
m=5
n=10

do i=1,n
    x(i)=i
    y(i)=i
end do

ptitle='y=f(x) '
xtitle='x'
ytitle='y'

call gks001 (l,m,n,x,y,ptitle,xtitle,ytitle)
end program main1
```



# رسم نمودار $x=x(t), y=y(t)$

```

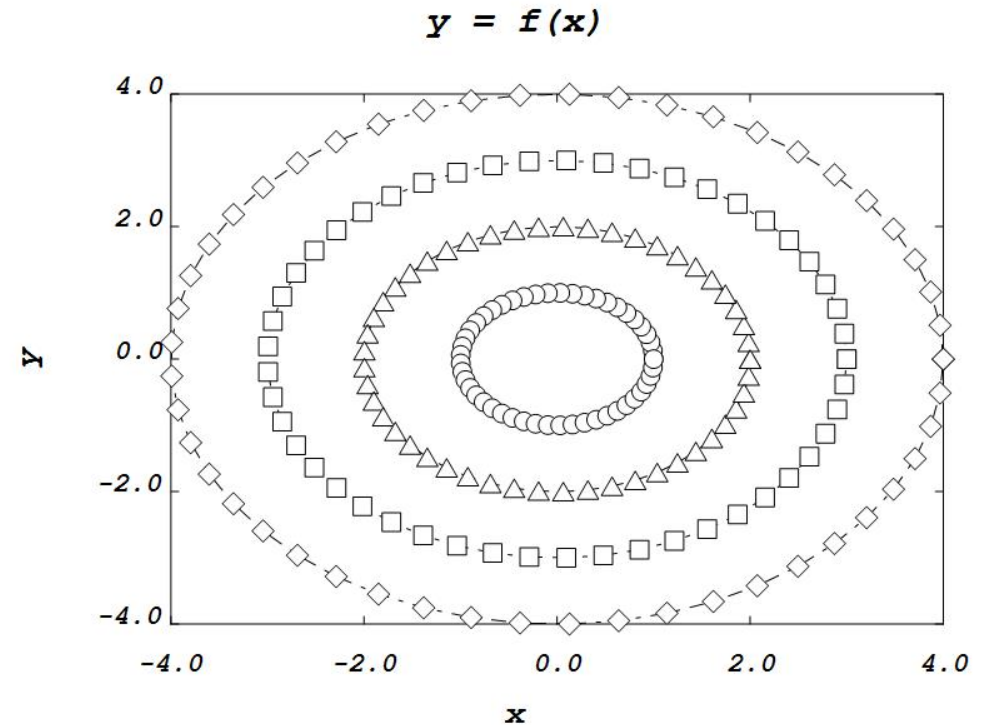
program main2
implicit none
integer:: i, n
integer:: nmax
parameter(nmax = 100)
integer:: l1, l2, l3, l4, m1, m2, m3, m4, n1, n2, n3, n4
double precision:: x1(nmax), x2(nmax), x3(nmax), x4(nmax)
double precision:: y1(nmax), y2(nmax), y3(nmax), y4(nmax)
double precision:: cosi, delta, sini, t(nmax)
double precision:: a, b, c, d, pi2, zero, one
parameter(a = 1.0d+00, b = 2.0d+00, c = 3.0d+00, &
d = 4.0d+00, pi2 = 6.2831853, zero = 0.0d+00, one = 1.0d+00)
character:: ptitle*8, xtitle*1, ytitle*1
logical:: axes, gsave
parameter(axes = .true., gsave = .true.)
external gks004
intrinsic sin, cos, dble
l1 = 1
l2 = 2
l3 = 3
l4 = 4
m1 = 5
m2 = 8
m3 = 11
m4 = 14
n = nmax/2
n1 = n
n2 = n
n3 = n
n4 = n
delta = pi2/(dble(n) - one)
t(1) = zero
do i = 2, n - 1
    t(i) = t(i - 1) + delta
end do
t(n) = pi2
do i = 2, n - 1
    t(i) = t(i - 1) + delta
end do
t(n) = pi2

```

```

do i = 1, n
    cosi = cos(t(i))
    sini = sin(t(i))
    x1(i) = a*cosi
    x2(i) = b*cosi
    x3(i) = c*cosi
    x4(i) = d*cosi
    y1(i) = a*sini
    y2(i) = b*sini
    y3(i) = c*sini
    y4(i) = d*sini
end do
ptitle = 'y = f(x)'
xtitle = 'x'
ytitle = 'y'
call gks004 (l1, l2, l3, l4, m1, m2, m3, m4, n1, &
n2, n3, n4, x1, x2, x3, x4, y1, y2, y3, y4, ptitle, &
xtitle, ytitle, axes, gsave)
end program main2

```



# رسم نمودار $r=r(\theta)$

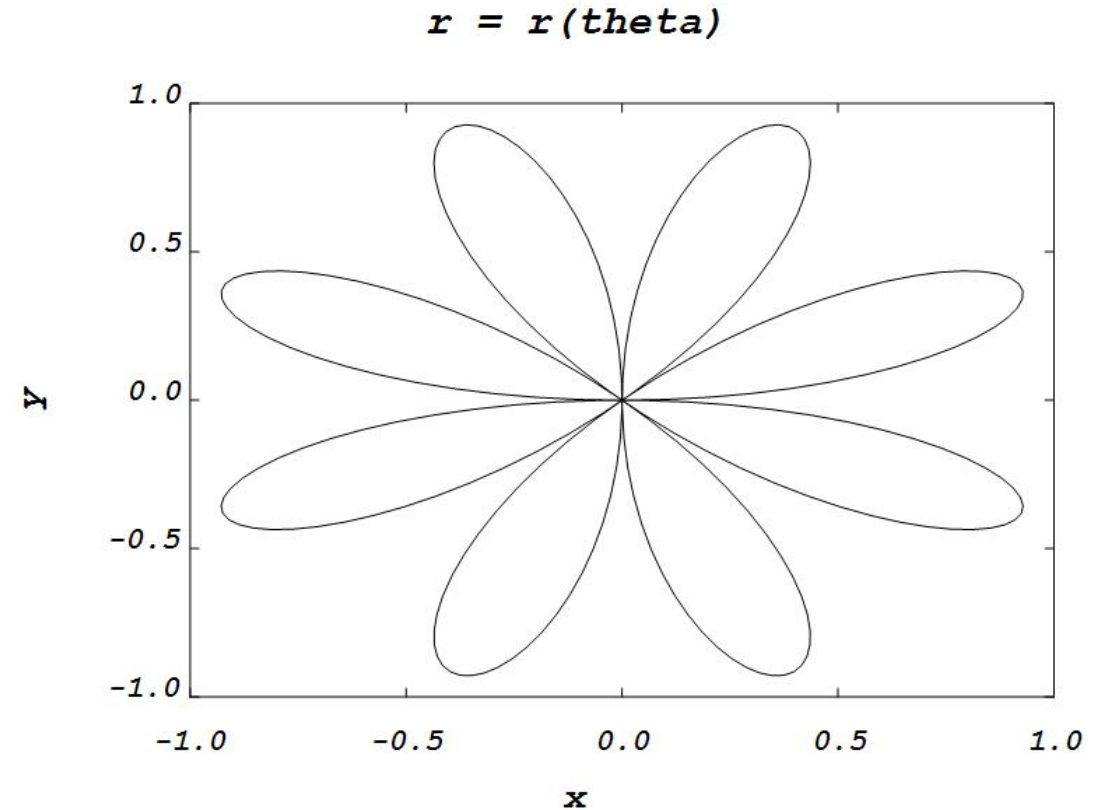
```
program main3
implicit none
integer::i,n
parameter(n = 250)
double precision:: r(n), theta(n)
double precision:: delta, pi, x01aafg
double precision:: zero, two, four
parameter(zero = 0.0d+00, two = 2.0d+00, four = 4.0d+00)
external x01aafg, rtplot
intrinsic dble,sin

pi = x01aafg(delta)
delta = two*pi/dble(n - 1)
theta(1) = zero

do i = 2, n - 1
    theta(i) = theta(i - 1) + delta
end do

theta(n) = two*pi
do i = 1, n
    r(i) = sin(four*theta(i))
end do

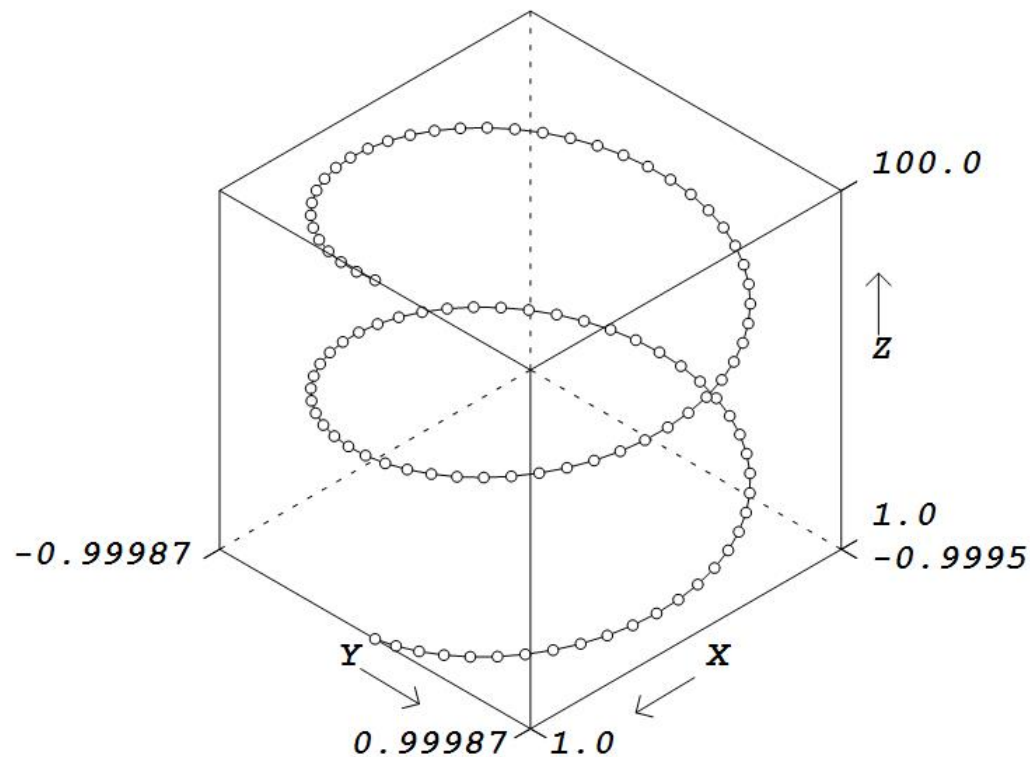
call rtplot(n,r,theta)
end program main3
```



# رسم نمودار $x=x(t), y=y(t), z=z(t)$

```
program main4
implicit none
integer:: i, n
integer:: nmax
parameter(nmax = 200)
double precision:: x(nmax), y(nmax), z(nmax)
double precision:: xtemp(nmax), ytemp(nmax)
double precision:: t(nmax)
double precision:: delta
double precision:: pi
parameter(pi = 3.1415927d+00)
double precision:: one, two, zero
parameter(one = 1.0d+00, two = 2.0d+00, zero = 0.0d+00)
external space0
intrinsic cos, dble, sin
n = nmax/2
t(1) = zero
t(n) = two*pi
delta = (t(n) - t(1))/(dble(n) - one)
do i = 2, n - 1
    t(i) = t(i - 1) + delta
end do
x(1) = one
y(1) = zero
z(1) = one
do i = 2, n - 1
    delta = two*t(i)
    x(i) = cos(delta)
    y(i) = sin(delta)
    z(i) = dble(i)
end do
x(n) = x(1)
y(n) = y(1)
z(n) = dble(n)
call space0 (n, nmax,x, xtemp, y, ytemp,z)
end program main4
```

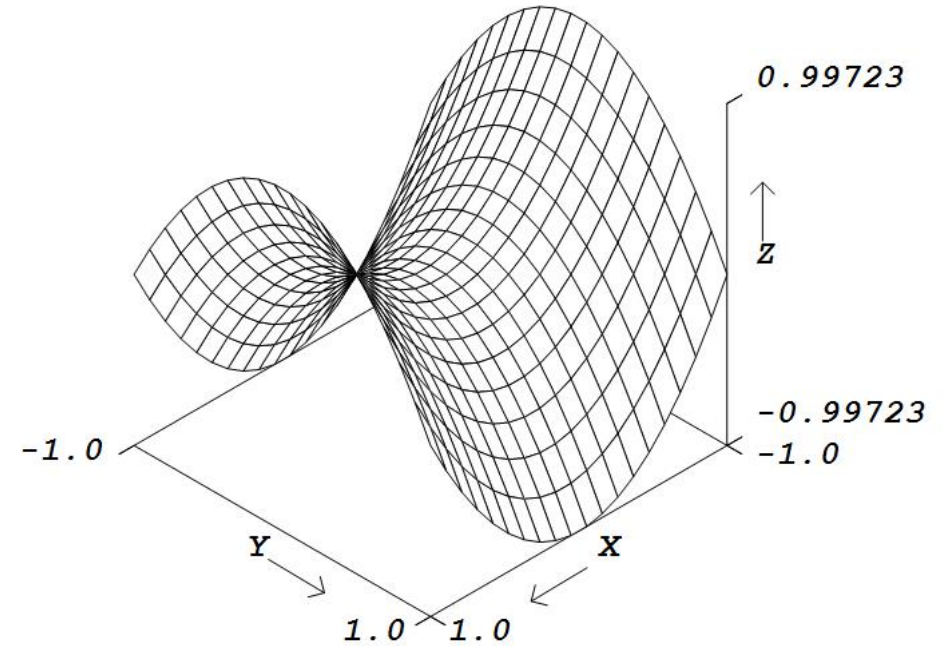
**SIMFIT 3D space plot (x,y,z)**



# رسم نمودار $z=f(x,y)$

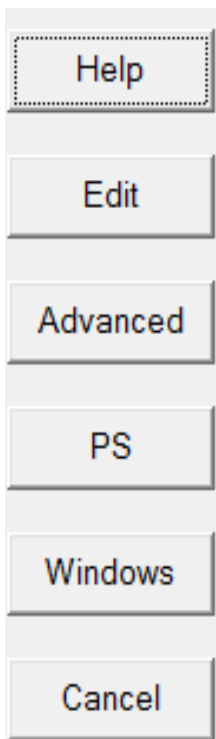
```
program main5
implicit none
integer:: i, j, nx, ny
integer:: isend, nmax
parameter(isend = 4, nmax = 100)
double precision:: x(nmax), y(nmax)
double precision:: xmax, xmin, ymax, ymin
double precision:: vector(nmax**2 + 6), z(nmax,nmax)
double precision:: delta
double precision:: one
parameter(one = 1.0d+00)
logical:: unused(nmax,nmax)
external surd2s
intrinsic dble
nx = 20
ny = 20
x(1) = - one
x(nx) = one
delta = (x(nx) - x(1))/(dble(nx) - one)
do i = 2, nx - 1
    x(i) = x(i - 1) + delta
end do
y(1) = - one
y(ny) = one
delta = (y(ny) - y(1))/(dble(ny) - one)
do j = 2, ny - 1
    y(j) = y(j - 1) + delta
end do
do j = 1, ny
    do i = 1, nx
        z(i,j) = x(i)**2 - y(j)**2
    end do
end do
xmax = x(nx)
xmin = x(1)
ymax = y(ny)
ymin = y(1)
call surd2s (isend, nmax, nx, ny,vector, xmax, xmin, ymax, ymin, z,unused)
end program main5
```

*SIMFIT 3D plot for  $z = f(x,y)$*





# ویرایش نمودار



عملکرد	گزینه
محدوده و تقسیم بندی محورهای X و Y	range
نحوه نمایش نمودار	style
حذف تمام تغییرات اعمال شده	undo
عنوان نمودار	title
شکل خطوط و علائم	line/symbol
دقت محورهای X و Y	precision
بیرون رفتن از پنجره edit برای دیدن تغییرات اعمال شده	quit/view

عملکرد	گزینه
ویرایش و یا حرکت عانوین نمودار	titles
ویرایش و یا حرکت محورهای افقی و عمودی نمودار	legends
ویرایش محورها، محدوده ها، تقسیم بندی های محور افقی و عمودی، یادداشت و برچسب ها	labels
ویرایش نسبت ابعاد، انحرافات و نحوه نمایش نمودار	style
علائم، نوع خطوط، ویرایش/بازگرداندن/اضافه کردن و متوقف کردن داده	data
ویرایش رنگ های پس زمینه، محورها، علائم، خطوط و برچسب ها	colours
تبدیل کردن و تغییرات مختصات	transform
فراخوانی و نوشتن فایل های پیکربندی	configure
ذخیره و پرینت در فرمت postscript	ps
ذخیره و پرینت در فرمت windows	win

عملکرد	گزینه
انتخاب رشته ای از متن برای برچسب گذاری	text
ویرایش و یا حرکت عناوین نمودار	=>text
انتخاب یا ویرایش فلش، خط یا جعبه برای برچسب گذاری	A/L/B
درگ کردن فلش، خط یا جعبه (head^ انتخاب شده	=>A/L/B^
درگ کردن فلش، خط یا جعبه (tail_ انتخاب شده	<=A/L/B_
انتخاب یا ویرایش شی گرافیکی برای برچسب گذاری	object
درگ کردن شی گرافیکی انتخاب شده	=>object
انتخاب یا ویرایش پنل اطلاعات	panel
درگ کردن یک طرف پنل اطلاعات	=>panel