

زیر برنامه ها : زیر برنامه ها و توابع قطعه برنامه های جامعی هستند که برای انجام کار معینی نوشته شده اند این قطعات ممکن است در برنامه اصلی به دفعات مورد بهره برداری قرار گیرند و البته هر بار داده های معینی را دریافت کنند و جواب معینی را به برنامه اصلی بازگردانند داده های زیر برنامه یا تابع را پارامتر گویند . که ممکن است به صورت ثابت ، متغییر ، یا عبارت باشند .

### مزایای استفاده از زیر برنامه ها

- ۱- عیب یابی ساده تر است
- ۲- اعمال تغییرات و توسعه راحت تر است
- ۳- تقسیم مسئله به مسائل کوچکتر
- ۴- آزمایش مستقل هر قسمت برنامه
- ۵- تکرار دستورات کم می شود.

تابع *Function* : مجموعه ای از یک یا چند دستور است که تقریباً مستقل بوده و یک نتیجه یا مقدار بازگشتی را برمیگرداند.  
پدرازه *Procedure* : مجموعه ای از یک یا چند دستور العمل است که کار یا عملی را برای ما انجام می دهد .

### پدرازه *Procedure*

- محل تعریف در قسمت تعاریف برنامه قبل از بلوک اصلی .
- عنوان هر پدرازه حداقل شامل نام پدرازنده و کلمه *Procedure* می باشد.
- نحوه کاربرد و فراخوانی پدرازه مانند دستورات ( فقط نام و تعیین پارامترها مانند *Inc , Dec , Break , WriteIn* )
- تعداد پارامترهای تعریف شده در پدرازه باید مساوی تعداد آنها در فراخوانی باشد.
- نوع اطلاعات تعریف شده در عنوان و فراخوانی پدرازه باید متناسب (تطابق) باشد.
- می تواند چند ورودی و چند خروجی داشته باشد.

### تابع *Function*

- محل تعریف مانند پدرازه .
- عنوان تابع حداقل شامل کلمه *Function* و نام و نوع خروجی می باشد.
- برای فراخوانی تابع می توان آنرا مانند یک متغییر در یک عبارت و یا دستورات انتساب و یا یک دستور خروجی بکار برد.
- تعداد پارامترها هنگام فراخوانی با تعریف یکسان باشد .
- نوع اطلاعات هنگام فراخوانی و تعریف باید متناسب باشد .
- چند ورودی و حداقل یک خروجی دارند. (فقط یک خروجی دارد نه بیشتر نه کمتر) .
- معمولاً در انتهای تابع داده خروجی به نام تابع واگذار می شود پس تابع حداقل یک دستور دارد.
- به تابعی که از درون بلوک خودش فراخوانی شود تابع بازگشتی (*Recursive*) گفته می شود .
- نوع نتیجه تابع بازگشتی تابع حتماً باید از انواع ساده یا استاندارد یا اسکالر باشد . ( نمی تواند از نوع رکورد یا آرایه باشد).
- تابع می تواند در بعضی از حالت های خاص ورودی نداشته باشد که در این صورت در تعریف تابع جفت پرانتز ها را نمی نویسیم.
- از آنجایی که حداکثر حافظه پشته (*Stack*) ۶۴ کیلوبایت است اگر تعداد فراخوانی های توابع بازگشتی پی در پی افزایش یابد با پیغام خطای (*Stack OverFlow*) برخورد می کنیم و برنامه متوقف می شود .

**تذکر :** تابع را می توان به عنوان ورودی به زیر برنامه ای دیگر ارسال کرد اما پردازه را نمی توان ارسال کرد.  
**تذکر :** استفاده از عبارتهای محاسباتی در قسمت پارامترهای واقعی صحیح نیست ولی درمورد پارامترهای ظاهری (مقداری) محدودیت ندارد.

**نکته :** متغیرهایی که در داخل یک پردازنده (زیرروال) تعریف می شوند ومورد استفاده قرار می گیرند متغیرهای محلی می گویند. متغیرهایی که در بخش تعریف برنامه اصلی معرفی می گردند متغیرهای سراسری می گویند.

**انواع پارامترها :** با استفاده از پارامترها می توانید مقادیری را به پروسیجرها ارسال یا دریافت کنید .  
**پارامترهای حقیقی و پارامترهای ظاهری(مقداری) :** به هنگام تعریف یا شناسایی یک تابع یا پردازه تمامی پارامترهای آن **ظاهری** هستند اما به هنگام فراخوانی همان تابع یا پردازه با مقادیر جایگزین شده در پارامترها از پارامترهای واقعی استفاده می کنیم .

**پارامترهای ظاهری(مقداری) خود دارای ۴ نوع می باشند :**

(۱) **پارامترهای ارزشی :** فقط می توان از آنها برای ارسال مقادیر استفاده نمود .

(۲) **پارامترهای مرجع :** می توان از آنها برای ارسال و دریافت مقادیر استفاده نمود.

- برای تعیین پارامترهای مرجع باید قبل از ذکر نام آن از کلمه رزرو شده *Var* استفاده کنید .

- در فراخوانی پروسیجرهایی که دارای پارامتر مرجع هستند باید از یک متغیر هم نوع استفاده کنید .

(۳) **پارامترهای متغیر بدون نوع :** برای ارسال و دریافت پارامترهای بدون نوع استفاده می شود .

- این نوع پارامتر شبیه پارامتر مرجع می باشد با این تفاوت که بعد از ذکر نام پارامتر ، نوع آن قید نخواهد شد.

- معمولاً در برنامه ها از این نوع پارامترها استفاده نمی شود .

(۴) **پارامترهای ثابت :** به پارامترهای گفته می شود که در پروسیجر مربوطه نمی توان مقدار آنها را تغییر داد .

- برای تعریف این نوع پارامترها باید قبل از ذکر نام آن از کلمه رزرو شده *Const* استفاده کرد .

**Procedure Add(a,b : integer);**

*Begin*

*Writeln(a,b);*

*End;*

*Begin*

*Add(۸,۱۰);*

*End.*

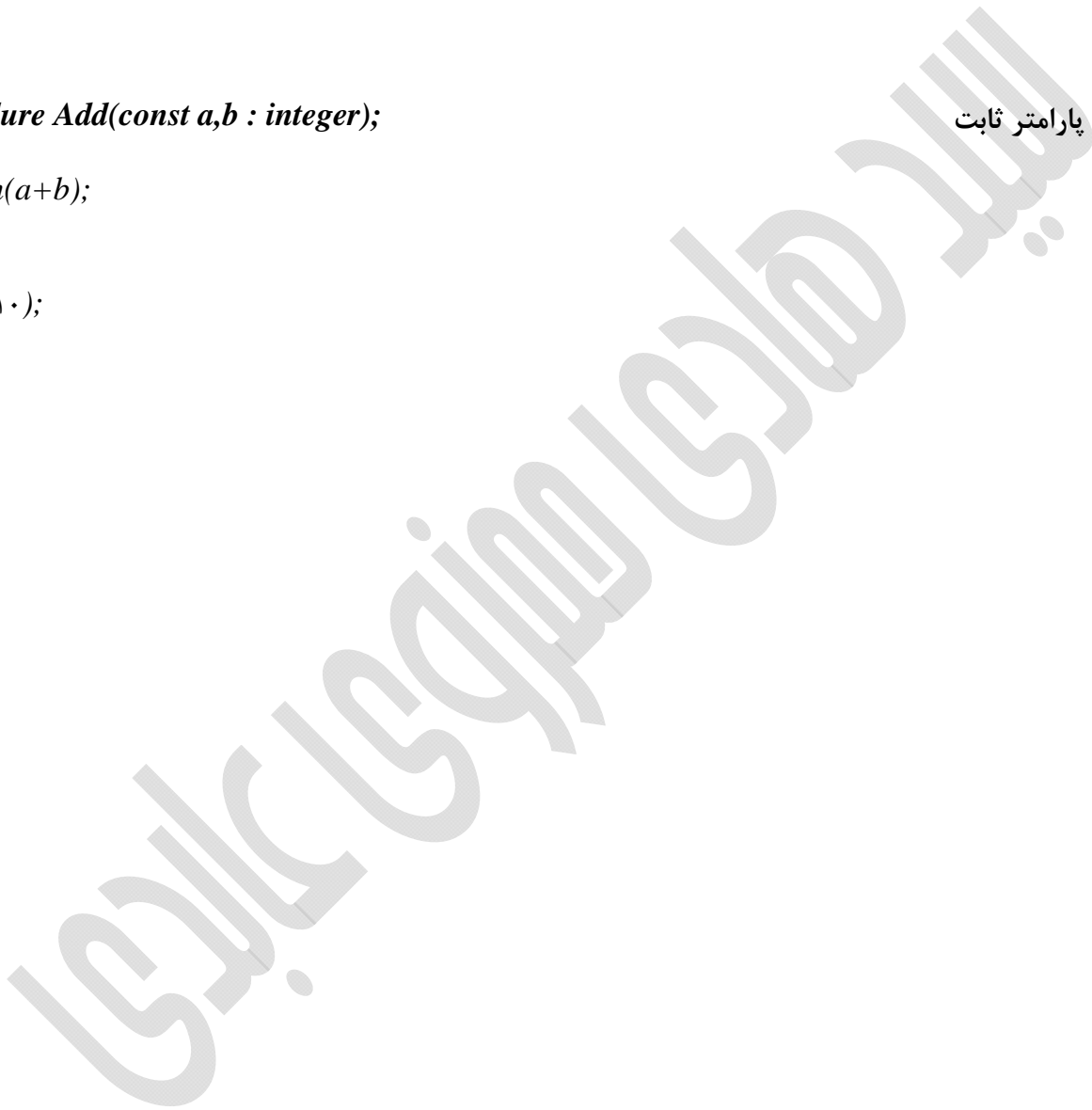
مثال (۱) پارامتر ارزشی

```
Var a:integer;  
Procedure Add(a,b : integer;Var c: integer);  
Begin  
C:=a+b;  
End;  
Begin  
Add(۸,۱۰,a);  
Writeln(a);  
End.
```

مثال (۲) پارامتر مرجع

```
Procedure Add(const a,b : integer);  
Begin  
Writeln(a+b);  
End;  
Begin  
Add(۸,۱۰);  
End.
```

مثال (۳) پارامتر ثابت



۱- کدام گزینه صحیح نمی باشد؟

- (۱) پردازنده می تواند بصورت یک دستور فراخوانی شود.
- (۲) تابع می تواند در عبارتها مورد استفاده قرار گیرد.
- (۳) پارامتر ظاهری (مقداری) هنگام ارسال به زیر برنامه آدرس آن به زیر برنامه ارسال میگردد
- (۴) پارامتر مرجعی یا خروجی هنگام ارسال به زیر برنامه آدرس آن به زیر برنامه ارسال می گردد

۲- کدام گزینه صحیح نمی باشد ؟

- (۱) پردازنده و تابع هر دو می توانند دارای پارامترهای ورودی و خروجی باشند
- (۲) تفاوت تابع و پردازنده در داشتن مقدار خروجی است
- (۳) هنگام فراخوانی پارامترهای مرجعی باید به جای آنها تنها از متغیر یا ثابت نوع دار استفاده کنیم.
- (۴) هنگام فراخوانی پارامترهای ظاهری (مقداری) باید بجای آنها تنها از متغیر استفاده کرد.

۳- کدام گزینه صحیح نمی باشد ؟

- (۱) پردازنده ها و توابع قبل از بلاک اصل معرفی می شوند.
- (۲) می تواند یک پردازنده یا تابع درون یک تابع یا پردازنده دیگر معرفی شود
- (۳) یک پردازنده یا تابع می تواند خودش را فراخوانی کند
- (۴) یک پردازنده یا تابع نمی تواند دیگری را فراخوانی کند

۴- کدام گزینه صحیح است ؟

- (۱) متغیرهای سراسری در تمامی بلاکهای برنامه قابل دسترسی می باشند
- (۲) متغیرهای محلی با پایان یافتن اجرای پردازنده یا تابع از بین می روند
- (۳) متغیرهای سراسری در بخش تعاریف بلاک اصلی معرفی می شوند
- (۴) هر سه مورد

۵- کدام گزینه صحیح است ؟

- (۱) متغیرهای سراسری و محلی هر دو دارای مقدار پیش فرض می باشند
- (۲) متغیرهای محلی دارای مقدار پیش فرض بود ه اما متغیرهای سراسری مقدار پیش فرض ندارند .
- (۳) پارامترهای ظاهری (مقداری) و مرجعی هر آدرس آنها به زیر برنامه ارسال می شود .
- (۴) اگر جلوی پارامتر کلمه *CONST* قرار گیرد در طول زیر برنامه قابل تغییر نیست .

۶- معادل تابع  $M$  کدام است ؟ $Function\ m(x, y : integer) : real ;$ 

Begin

 $M := \ln(\exp(x) * \exp(y));$ 

End;

 $X - Y$  (۲) $X + Y$  (۱) $X / Y$  (۴) $X * Y$  (۳)۷- اگر  $b : real, a : integer$  باشد و پردازنده یا رویه *main* بفرم زیر تعریف شده باشد کدام گزینه صحیح است ؟ $procedure\ main(var\ x : integer ; y : real) ;$  $main(int(a), b);$  (۲) $main(trunc(a), b);$  (۱) $main(a * 2.0, b - 2);$  (۴) $main(a, trunc(b));$  (۳)۸- حاصل عبارت زیر کدام است ؟  $Write(13\ SHR\ 2\ SHL\ 2);$ 

۱۲ (۴)

۶ (۳)

۱۲ (۲)

۱۳ (۱)

```
Function A ( x : integer ) : boolean ;
If x mod ۲ <> ۰ then
A := true
Else
A := false ;
End;
Const
X : integer = ۳ ;
```

۱۰- خروجی تابع زیر معادل کدام است ؟

- (۱) ABS  
(۲) ODD  
(۳) Round  
(۴) Int

```
Procedure p( var a : integer ; b : integer ) ;
Begin
A:=a*b div a + a ;
B := a * b div a ;
End;
Begin
P ( x , x ) ;
Writeln ( x , x ) ;
End.
```

۱۱- خروجی برنامه روبرو کدام است ؟

- (۱) ۳ ، ۶  
(۲) ۳ ، ۳  
(۳) ۶ ، ۶  
(۴) ۳ ، ۳

```
FUNCTION main ( x: integer ) : integer ;
Begin
If x < ۳ then
Main := ۱۰
Else
Main := main ( x-۱ ) * x;
End;
Begin
Write ( main ( ۳ ) ) ;
End.
```

۱۲- خروجی برنامه روبرو کدام است ؟

- (۱) ۶  
(۲) ۳۰  
(۳) ۲۰  
(۴) ۴۰

۱۳- ..... پر دازه و ..... تابع می باشد ؟

- (۱) REAL , WRITE (۱)      (۲) SIN , SQR (۲)      (۳) EXP , WRITE (۳)      (۴) REAL , LN (۴)

۱۴- در چه قسمت از برنامه عملکرد دو دستور EXIT و HALT یکسان می باشد ؟  
(۱) در زیر برنامه      (۲) بلاک اصلی      (۳) درون تابع      (۴) هر سه مورد

۱۵- هنگام ارسال پارامترهای مرجعی ..... پارامتر به زیر برنامه ارسال می شود ؟  
(۱) آدرس      (۲) مقدار      (۳) عبارت      (۴) متغیر

```
VarA,b,c:integer;
Procedure test ( var x,y :
integer );
Begin
x := x+ y ;
Y := y* ۲ ;
B := c - a ;
End ;
Begin
A:=۱۰ ; b:=۴ ; c:=۸ ;
Test ( a,b ) ;
Write ( a,b,c);
End.
```

۱۶- خروجی برنامه روبرو چیست ؟

- (۱) ۱۰۴۸  
(۲) ۱۴۸۸  
(۳) ۱۴۸۱۲  
(۴) ۱۴-۶۸

پاسخنامه :

۱- گزینه ۳

۲- گزینه ۴

۳- گزینه ۴

۴- گزینه ۴

۵- گزینه ۴

۶- گزینه ۱ :  $\ln(\exp(x) * \exp(y)) = \ln(\exp(x+y)) = x+y$ ۷- گزینه ۳ : چون پارامتر  $x$  مرجعی یا خروجی است تنها می توانیم بجای آن از متغیر صحیح یعنی  $x$  استفاده کنیم

۸- گزینه ۲

۹- گزینه ۲ :

$$f(3) = f(2) + f(1) = 3 + 2 = 5$$

$$f(2) = f(1) + f(0) = 2 + 1 = 3$$

$$f(1) = 2 \quad f(0) = 1$$

۱۰- گزینه ۲ : تابع فرد بودن عدد را بصورت منطقی بر می گرداند

۱۱- گزینه ۳

۱۲- گزینه ۲ :  $main(3) = main(2) * 3 = 10 * 3 = 30$ ۱۳- گزینه ۳ :  $write$  پردازنده بوده و مانند دستور بکار می رود.

۱۴- گزینه ۲

۱۵- گزینه ۱

۱۶- گزینه ۴