کنترل command button:
شما باید برای انجام هر کاری در ویندوز کلیک کنید این دکمه دارای عنوان است و پس از کلیک کاری را انجام می دهد این دکمه همان Command button است در مثال های قبل از رخداد این کنترل استفاده کرده ایم مثلا Command _ click مشخص می کند که بعد از کلیک شدن Ok چه کاری باید انجام شود.

تابع Input box:
پیامی را به کاربر نمایش می دهد و از می خواهد تا مقداری را وارد کند و شکل کلی آن به صورت زیر است:

A = input box (" پیغام " , [title] , [default] , [xpos] , [ypos])

title وپیغام مانند msg boxعنوان و پیغام را در صفحه به کاربر نمایش می دهد Default مقدار پیش فرض ورودی می باشد و در صورتی استفاده می شود که کاربر مقداری را وارد نکرده Xpos , ypos مختصات محل پنجره را در صفحه مشخص می کند به صورت پیش فرض مقداردر وسط صفحه نمایش داده می شود و کلید پارامترهای Input box به جز پیغام اختیار ی می باشد.

برنامه ای بنویسید که دو عدد را از کاربر بخواهد و حاصل ضرب آن را به کاربر نمایش دهد و در صورتی که کاربر مایل به ادامه عملیات باشد این کار را تکرار کند:

Form _ load ( )

D = ٦

Do until (d= ٧)

A = input box (" enter a " , ٠)

B = input box (" enter b " , ٠)

C = a * b

D = msg box (c, ٤)

Loop

الف – برنامه ای بنویسید که نام و نام خانوادگی که اگر نام و نام خانوادگی اجازه دهد شماره دانشجویی وارد شود؟

ب – در صورتی که هر سه مقدار وارد شد و کلید افزودن فشرده شود از کاربر بخواهد مقدار وارد کند؟
ج – بر نامه ای بنویسید نام و نام خانوادگی و شماره دانشجویی را در صورتی که شماره دانشجویی وارد شده باشد نام و نام خانوادگی را فعال کند؟

توابع:
در ویژوال بیسیک توابع بر دو نوع می باشد Procedure, function ها برای تعریف پروسیجر از شکل کلی زیر استفاده می کنیم:

Public | private sub name لیست آرگومانها

دستورات

[exit sub]

End sub.

در عبارت فوق کلمه های کلیدی Public, privateکه به صورت اختیاری قرار می گیرند مشخص می کند که تابع مورد نظر در چه سطحی فراخوانی می شود پابلیک تعیین می کند که پروسیجر در همه ی زیر برنامه ها می تواند فراخوانی شود و یک پروسیجر عمومی است.
privateمشخص می کند که پروسیجر تنها در جایی که تعریف شده است می تواند فراخوانی شود آرگومانها متغییر هایی هستند که در هنگام فراخوانی باید آنها را به پروسیجر پاس کرد هر پروسیجر دارای یک نام می باشد کلمه کلیدی Exit sub باعث می شود که اجرای پروسیجر قطع شده و برنامه از آن خارج شود در صورتی که پابلیک و پریویت مشخص نشود پیش فرض آن پابلیک است

مثال:

تابعی بنویسید که Max سه عدد را مشخص کند و در یک تکست باکس نتیجه را بنویسد.

```
Private sub max (x,y,z as integer)

  Max = x

  If ( y > max ) then max = y end

  If ( z > max ) then max = z end

End sub
```

برای تعریف پروسیجر خارج از بدنه فرم باید از قابلیت ماژول استفاده کنیم به این منظور یک ماژول ایجاد کرده و کد پروسیجر را داخل آن می نویسیم به این ترتیب پروسیجر برای تمام فرم ها و زیر برنامه ها در دسترس خواهد بود.

فراخوانی زیر برنامه ها:
برای فراخوانی پروسیجر ها از دستور کال استفاده می کنیم شکل کلی آن به این صورت است :

Call نام پروسیجر [لیست آرگومانها]

برای فراخوانی پروسیجر کلمه ی کلیدی کال و سپس نام پروسیجر و سپس لیست آرگومانها که داخل پرانتز قرار دارند را پشت سرهم می نویسیم به این ترتیب پروسیجر با مقادیر مورد نظر فراخوانی می شود "

```
  Call max ( ١, ٢, ٣)

    A=١٢

    B=١٤

    C=١١
Call (a, b, c)
```

فراخوانی پروسیجر ها:
به دو صورت انجام می شود فراخوانی با مقدار و فراخوانی با ارجاع در فراخوانی با مقدار آرگومانها له داخل تابع با روال (پروسیجر) پاس می شود و در جریان روال مقادیر اولیه بدون تغییر باقی مانده اما در فراخوانی با ارجاع مقادیر به داخل تابع فرستاده می شود و با تغییرات آن تغییر می کند
مثال:

```
Sub change a ( by Val a as integer  )
```

```
  A =۲۰

End sub

   Z=۱۰

Call change (z)
Print z
```

۱۰

```
Sub change a ( by ref a as integer )
    A=۲۰
End sub
X = ۱۵
Call change (x)
Print x
```

۲۰

بکارگیری Exit sub:

براساس تعدادی شرط می توان تصمیم به خروج از پروسیجر گرفت به عنوان مثال وقتی که پارامترهای ورودی با همان آرگومانها مقادیر مورد نظر را نداشته باشد مثلا روالی که عمل تقسیم را انجام می دهد با پارامتر صفر فراخوانی شود در صورتی که روال بکار رفته خود ادامه دهد تقسیم بر صفر اتفاق می افتد پس می توان با کنترل آرگومانها ورودی از چنین اتفاقی جلوگیری کرد.

توابع:

توابع همانند روال قطعه ای از کد تکراری هستند که فراخوانی می شوند تفاوت آن ها با روال در پارامتر خروجی می باشد شکل کلی آن ها به این صورت است:

```
Public | private function   نام تابع    لیست آرگومانها

  دستورات

       [ exit function ]

       [  مقدار تابع  =   مقدار ]

End function
```

کلمات کلیدی Public , private مانند روال عمل می کنند کلمه کلیدی فانکشن مشخص می کند که این زیر برنامه یک تابع است لیست آرگومانها و نام تابع از همان قوانین طبعیت می کند کلمه کلیدی اگزیت فانکشن سبب خروج از برنامه می شود همچنین عبارت مقدار = تابع خروجی تابع را مشخص می کند.

مثال :
تابعی بنویسید که سه عدد را گرفته و Max آن ها را برگرداند:

```
Private   function    max    (a , b, c  as integer )

    Maxn = a

    If ( b> maxn ) then maxn = b end

    If ( c > maxn ) then maxn = c end

                Maxn = maxn

End function
```

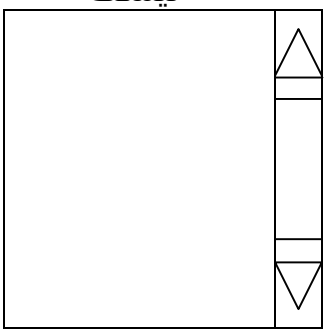
آرگومانهای اختیاری :
هر گاه در اول نام یک آرگومان در تابع یا روال کلمه ی کلیدی Optional قرار دهیم آن آرگومان اختیاری می شود آرگومان اختیاری بدین معناست که شما می توانید به آن مقدار دهی کنید یا اصلا آن را در نظر نگیرید .

مثال :
پروسیجری بنویسید که حداقل یک عدد را و حداکثر سه عدد را به عنوان ورودی بگیرد و آن را به یک لیست اضافه کند:

```
Sub add items ( x s integer , optionally as integer , optional z as integer )

    List ۱ . add item  x

    List ۲ . add item y

    List ۳ . add item  z
 End sub
```

کنترل لیست :
کنترل لیست برای نمایش تعدادی قلم یا آیتم و انتخاب یک یا چند قلم از بین آن ها می شود مثلا استان های ایران را می توان در قالب لیست نمایش داد که هر شخصی به عنوان استان زادگاه خود یکی از آنها را انتخاب کند شکل کلی لیست به این صورت است :

لیست



List ١ as list

List ١ . add item ١

List ١ . add item ٢٥

```
Call add items (١)

Call add items (١١ , ٢٥)

Call add items (٢٢ , ٢٣ , ٥١)
```

لیست دارای تابع به نام Add item است که اقلامی را به لیست اضافه می کند.

کلمات کلیدی Optional باعث شده است که بتوانیم تابع را با یک دو یا سه آرگومان فراخوانی کنیم.

توابع محاسباتی :

١- Abs

این تابع قدر مطلق یک عبارت عددی را می گیرد :

Abs (-١٠) = ١٠ abs (١٠) = ١٠ abs (٠) = ٠

٢- Cos ( )

مقدار Cos زاویه مورد نظر را که بر حسب رادیان است بر کی گرداند :

Cos (٠) = ١ cos (٢ * ٣,١٤) = ١ cos (٣,١٤) = ١

۳ – Exp ( )
مقدار عددی را گرفته و E^x را محاسبه می کند .

٤ - تابع int ( ) (جز صحیح)
عدد اعشاری را گرفته و قسمت صحیح آن را بر می گرداند:

Int (-۲،٦) = -۳ int (۱,٥٤) = ۱

٥ – تابع fix ( ):
این تابع قسمت اعشاری عدد را قطع می کند و عدد صحیح را بر می گرداند :

Fix (٥) = ٥ fix (-۳,٥) = -۳

٦ – تابع Log:
لگاریتم عدد را د مبنای E بر می گرداند:

Log (۱) = ۰

۷ – تابع Round :
این تابع یک عبارت عددی را رند می کند به عنوان ورودی عبارت عددی و تعداد ارقامی را که باید رند شوند را می گیرد .
اگر تعداد ارقام که باید رند شوند منفی باشد مانند تابع Fix عمل می کند .

مثال :

Round (۳۲٥ . ٥٥ . ۱) = ۳۲٦

۸ – تابع علامت Sgn:
این تابع علامت عدد را بر می گرداند به عنوان خروجی

۱	بزرگتر از ۰
۰	مساوی ۰

کوچکتر از ۰	-۱

۹ – تابع ( ) Sin:
این تابع مقدار Sin عدد را که بر حسب ر ادیان است حساب می کند.

sin (۰) = ۰ sin (۳,۱٤) = ۰ sin (۳,۱٤ | ۲) = ۱

۱۰ - تابع ( ) sqr:
این تابع ریشه دوم عدد را بر می گرداند مثلا :

Sqr (۱۰۰) = ۱۰

۱۱ - تابع ( ) Tan:
این تابع تانژانت عبارت عددی را بر می گرداند عبارت عددی بر حسب ر ادیان است .

برنامه زیر چه مقداری را چاپ می کند:

```
Function calc (byval | byref x as integer , y as integer )

   X = sqr ( y )

   Calc = x

End function
```

A = ۱۰۰

B = ٥

Calc (b , a)

Byval print a = ۱۰۰ byref print a = ۱۰۰

Print b = ٥ print b = ۱۰

رشته ها String:
مجموعه از حروف و اعداد که در کنار هم قرار می گیرند را رشته می گویند رشته ها بر دو نوع اند طول ثابت و طول متغییر تعریف رشته با طول متغییر به صورت زیر می باشد:

Dim s as string

و تعریف رشته با طول ثابت به صورت زیر می باشد:

Dim s as string * ۲۰ ;

اتصال رشته ها :
از دو عملگر + و & استفاده می شود + برای الحاق رشته هایی که شامل کاراکتر می باشند استفاده می شود

& برای رشته هایی که شامل سمبل ها و کاراکتر های Operator و متغییر استفاده می شود

مقایسه رشته ها:
تابع Strcomp برای مقایسه رشته ها استفاده می شود مقایسه دو رشته به فرم های مختلفی انجام می شود شکل کلی این تابع به این صورت است:

Strcomp (string ۱ , string ۲ , comp parison)

String ۲ , string ۱ رشته هستند که می خواهیم مقایسه کنیم یکی از انواع زیر می باشد:

۰	مقایسه باینری
۱	مقایسه متنی
۲	مقایسه با روش Ms access

متداول ترین نوع مقایسه ، مقایسه ی متنی است در این نوع مقایسه تمام سمبل ها و اعداد کوچکتر از حروف هستند:

.’ ali ‘ < ‘ ۱ ali ‘ < ‘ Aali ‘

_ a _ a < a l I < ali

تابع Strcomp طبق جدول زیر خروجی می دهد:

وقتی رشته اول کوچکتر است	-۱
وقتی دو رشته مساوی باشد	۰
وقتی رشته اول بزرگتر باشد	۱
وقتی یکی از رشته ها Nullاست	null

Null:
رشته ای است که مقداری در آن ذخیره نشده است.

توابع کار با رشته ها:

تابع Left $:
این تابع تعداد مشخصی کاراکتر را از سمت چپ رشته جدا کرده و بر می گرداند و شکل کلی آن به صورت زیر است:

Left (string length)

تعداد کاراکتر ها را پارامتر Length مشخص می کند.

مثال :

A $ = " computer "

Left $ (a $, ۳) = "com "

Left $ (a $, ۱) = " c "

تابع Right $:
این تابع تعداد کاراکتر مشخص را از سمت راست رشته مشخص می کند و شکل کلی آن به صورت زیر است:

Right $ (a $, ۲) = " er "

تابع Mid $:
قسمتی از رشته را یا طول مشخصی از مکان مشخصی را جدا می کند.

Mid $ (a $, ۲ , ۲) = “com “

توابع Trim , l trim , r trim:
trim کردن به معنای حذف کاراکتر های خالی از رشته می باشد تابع تریم کاراکتر های خالی را از دو طرف رشته پاک می کند.

تابع l trim تنها از سمت چپ و R trim از سمت راست کاراکتر های خالی را پاک می کند .

مثال :

A = “ _ _ ali _ _ _ “

Trim (a) = “ali”

L trim (a) =” ali _ _”

R trim (a) =”_ _ ali”

تابع Str:
این تابع برای تبدیل عدد به رشته استفاده می شود:

Z = ۱۲۳ x = str (z) = “۱۲۳”

Z+۵ = ۱۲۵ x+۵ = “۱۲۳۵”

تابع Instr:
این تابع رشته ای را در رشته ی دیگر جست و جو می کند شکل کلی آن به این صورت است:

Instr ([start] string ۱, string ۲)

String۲, string ۱ رشته های مقایسه و استارت یک پارامتر اعتباری است محل شروع جست و جو را نشان می دهد.

جدول زیر خروجی این تابع است:

طول رشته اول صفر باشد	۰

Null	رشته اول Null باشد
Start	طول رشته دوم صفر باشد
Null	رشته دوم Null باشد
۰	رشته دوم پیدا نشود
موقعیت شروع آنرا بر میگرداند	رشته دوم در رشته اول باشد
۰	String ۱ > string ۲

A = "Mohammad ali "

B = "ali"

Instr (a , b)
۹
۰

در صورتی صفر می شود که رشته دوم پیدا نشود:

B = " arman"

تابع Space:

این تابع به تعداد مشخص شده فضای خالی ایجاد می کند:

" ali " + space (٦) +" reza "

Ali _ _ _ _ _ _ reza