

# جزوه برنامه نویسی ۱

## ویژوال بیسیک ۱

مهندس محمدی

## فصل اول: حل مسال و آشنایی با ویژوال بیسیک

زبان برنامه نویسی مجموعه ای از نشانه ها (Token) و ترکیب نحوی (Syntax) آنها است که برنامه نویس برای حل مسائل از آن کمک می گیرد.

زبان های برنامه نویسی از چهار دیدگاه مختلف با هم مقایسه می شوند:

### الف) روش برنامه نویسی :

- ۱- روش زیرروالی (Procedure) مانند زبان کوبول - فرترن
- ۲- روش ساخت یافته (Structured) مانند زبان پاسکال - C
- ۳- روش مدولار (Modular)
- ۴- روش شی گرا:

در ابتدا برنامه نویسی به روشی نامشخص انجام می شد و بیسیک اولیه هم به همین صورت بوده است. در روش زیرروالی برنامه نویس مساله خود را به چندین عملکرد تقسیم می کند و برای هر عملکرد یک تابع (زیربرنامه) می نویسد در این روش بین کد اصلی و داده فاصله زیادی وجود دارد.

در برنامه نویسی ساخت یافته مساله از قطعات کوچکتر از تابع قابل حل است یعنی ابتدا مساله به تابع هایی تقسیم می شود سپس توابع نیز به قطعات کوچکتری نیز تقسیم می شود و هر تابع فرعی از تعدادی بلوک (مانند بلوک If, For و ...) تشکیل خواهد شد. بین داده ها و توابع فاصله ای وجود ندارد و داده ها بین توابع مبادله می شوند (آرگومان).

برنامه نویسی مدولار نوع توسعه یافته برنامه نویسی ساختار یافته است که در این روش توابع به قطعات کوچکتر و مستقل تقسیم می شود که به آنها مدول (قطعه) گفته می شود. هر مدول فقط به داده های خودش دسترسی دارد. در این روش می توان داده را پنهان یا کپسوله کرد. کپسوله (encapsulation): هنگامی که بخواهیم داده هایی را در توابع مجزا استفاده کنیم و از دید توابع دیگر مخفی بمانند داده ها را کپسوله می کنیم.

در برنامه نویسی پیشرفته برنامه ها به قطعات مستقل کوچکتری تقسیم می شوند که هر یک کار خود را انجام می دهند و به کل برنامه کاری ندارند.

در برنامه نویسی شی گرا مساله به بخش های تشکیل دهنده آن تجزیه می شوند که به آنها شی (Object) گفته می شود. هر شی دستورالعمل ها و داده های خود را دارد.

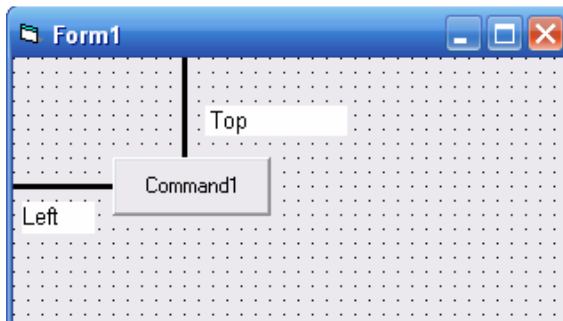
زبانهای برنامه نویسی شی گرا سه خاصیت مشترک وجود دارد: الف) کپسوله سازی (ب) چند ریختی (ج) وراثت کپسوله سازی باعث می شود که کد و داده ها با هم در یک جا قرار گیرند (مانند روش مدولار) و همچنین از تداخل خارجی جلوگیری می کند و باعث ایمنی می شود.

چندریختی یعنی از یک رابط به چندین روش استفاده شود و امکان طراحی رابط عمومی برای گروهی از عملیات مربوط به هم وجود دارد. (مثال فرمان اتومبیل)

انتقال خصوصیات یک شی به شی دیگر وراثت نامیده می شود به همین دلیل مفهوم سلسله مراتبی را پشتیبانی می کند. وراثت باعث می شود تا یک شی، نمونه خاصی از یک حالت عمومی تر باشد.

هر شی از سه جنبه شناخته می شود:

- الف) مشخصه ها (Properties) : حالت فعلی شی را نشان می دهد مانند رنگ
  - ب) متد یا رفتار: نحوه پاسخ یک شی در مقابل رویدادی است که ممکن است برایش رخ دهد.
  - ج) روابط: نشان دهنده ارتباط آن شی با اشیای دیگر است.
- هر برنامه شی گرا شامل تعدادی شی با مشخصه ه و متدهای متفاوت است به نحوی که روابط خاصی بین آنها برقرار باشد.
- متدها مجموعه ای از دستوالعمل ها است که باید در هنگام بروز رویداد اجرا شوند.



**ب) انواع زبان های برنامه نویسی از نظر نزدیکی به زبان ماشین:**

- ۱- زبان سطح پایین: زبانی که به زبان ماشین نزدیک است مانند ماشین و اسمبلی
- ۲- زبان سطح میانی: زبانهایی که هم به زبان ماشین نزدیکند و هم به زبان محاوره ایی انگلیسی نزدیکند. مانند زبان C
- ۳- زبان سطح بالا: زبانهایی هستند که به زبان محاوره ایی انگلیسی نزدیک هستند مانند ویژوال بیسیک

**ج) انواع زبانهای برنامه نویسی از نظر نوع ترجمه:**

۱- مفسری : interpreter

۲- کمپایلری

مفسرها برنامه را خط به خط خطایابی و ترجمه و اجرا می کنند مانند زبان بیسیک  
کمپایلرها برنامه را ابتدا از نظر خطا چک می کنند اگر خطایی وجود نداشته باشد کل برنامه را ترجمه می کنند و سپس آن را اجرا می کنند.  
مانند زبان پاسکال  
برنامه های مفسر نسبت به برنام های کمپایلر آهسته تر عمل می کنند و کمپایلرها در ابتدای کار تاخیر دارند. کمپایلرها ترجمه خود را ذخیره می کنند.  
ویژوال بیسیک نمونه ایی از زبان مفسروکمپایلری است.

فرترن توسط شرکت IBM برای کاربردهای علمی مهندسی (محاسبات پیچیده ریاضی) استفاده می شود. کوبول یک زبان تجاری است. زبان C (دنيس ریچی) برای تولید سیستم عامل Unix استفاده شده است. پاسکال (نیکلاس ورث) برای استفاده های علمی به کار می رود. ++C توسعه یافته C است و یک زبان برنامه نویسی شی گرا است.

**د) انواع زبانهای برنامه نویسی از نظر رابط برنامه نویسی:**

۱- مبتنی بر متن : مانند بیسیک - پاسکال - C

۲- مبتنی بر گرافیک (ویژوال) : مانند Delphi - Power Builder - Visual Basic

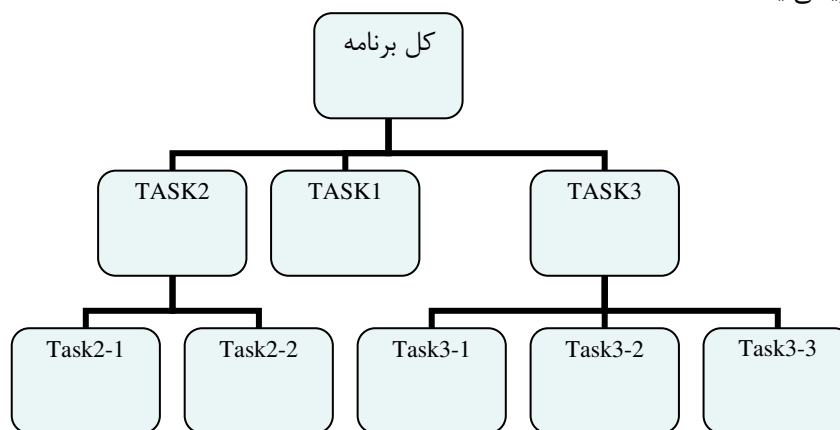
زبان جاوا ترکیبی از C و ++C است و مبتنی بر گرافیک است.

**تاریخچه زبان بیسیک:**

زبان بیسیک (Beginner's All purpose Symbolic Instruction Code) یک زبان برنامه نویسی سطح بالای همه منظوره برای افراد مبتدی است و بیشتر برای آموزش افراد مبتدی استفاده می شود. زبان بیسیک توسط جان کنی و توماس کورتس در کالج دارتموث پدید آمد. این زبان نخستین زبان برنامه نویسی نیست. هنگامی که زبان بیسیک ایجاد شد از سوی شرکت های مختلف مترجم های (مفسرهای) مختلفی برای آن تهیه شد مانند Qbasic و Ansi Basic و gwBasic و basica و Quick Basic و Visual Basic.

**طراحی برنامه:**

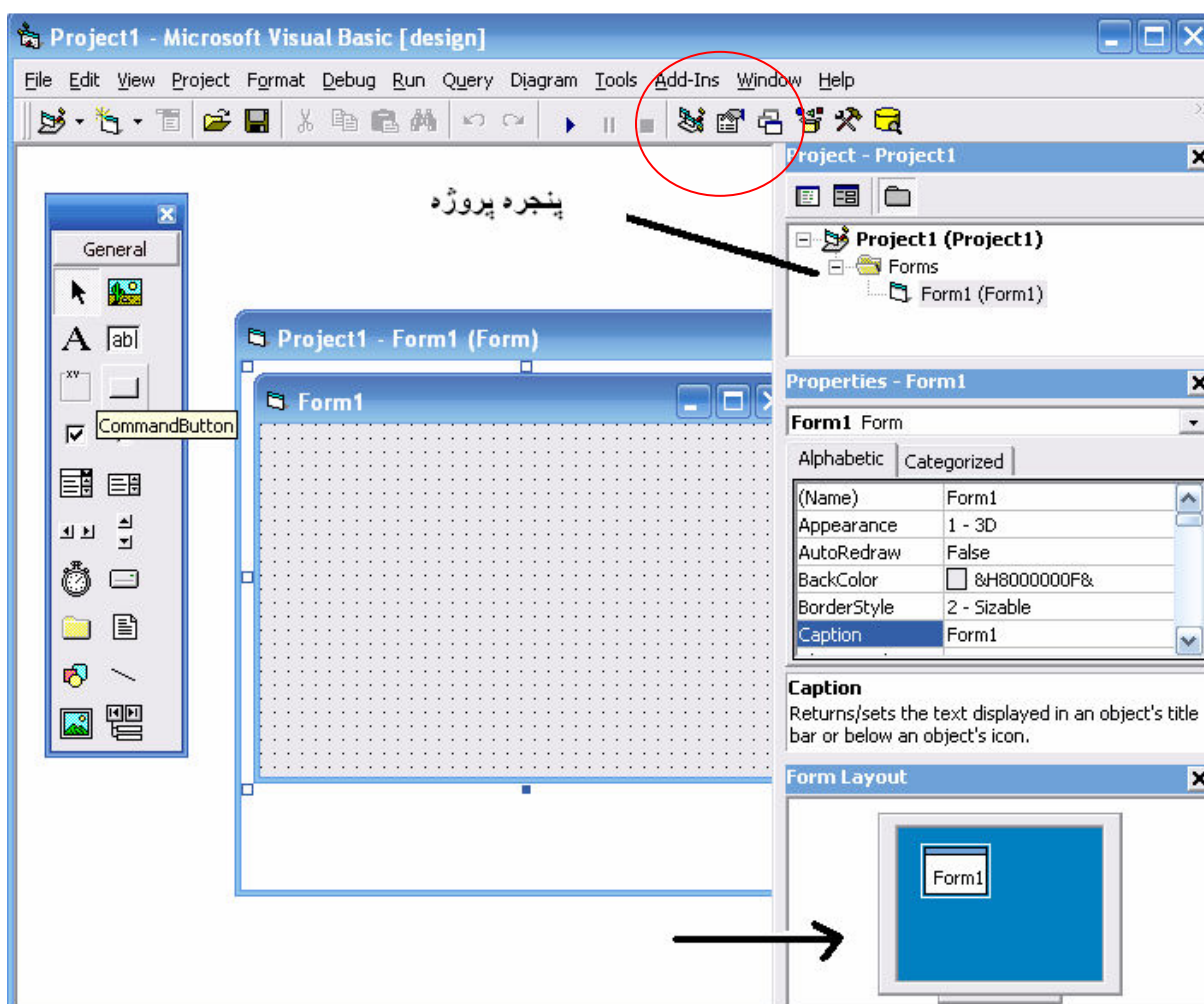
طراحی برنامه سه گام اصلی دارد: الف) تحلیل مساله ب) طراحی برنامه ج) به کارگیری برنامه  
تحلیل مساله (درک مساله) شامل بدست آوردن ورودی های مساله ، بدست آوردن خروجی مساله، بدست آوردن الگوریتمی که ورودی ها را به خروجی مرتبط کند. الگوریتم مجموعه ایی مرتب و سازماندهی شده از دستورالعمل ها است که برای حل یک مساله بکار می روند.  
مثال) برنامه ایی بنویسید که تعدادی عدد دریافت کند و میانگین آنها را نمایش دهد.  
وقتی قسمت اول (تحلیل مساله) انجام شد طراحی برنامه انجام می شود که در این گام با استفاده از جملات شبه انگلیسی که به آنها شبه کد (Pseudo Code) گفته می شود برنامه را طرح ریزی می کنند. برای طراحی برنامه از روش بالا به پایین استفاده می شود. در این روش برنامه به تعدادی وظیفه تقسیم می شود که آنها مدول اصلی (Task) گفته می شود سپس Task ها به مدول های فرعی تقسیم می شوند.



بعد از طراحی گام بعدی که به کارگیری برنامه است انجام می شود در این گام ابتدا شبه کد ها به کد اصلی تبدیل می شوند و برنامه نویس آنها را بوسیله یک زبان برنامه نویسی مثل ویژوال بیسیک تبدیل به کد می کند سپس اشتباهات برنامه بررسی می شود (Debugging) و در نهایت مستندات برنامه نوشته می شود. (Document)

## کاربا ویژوال بیسیک:

محیط ویژوال بیسیک IDE ( محیط های توسعه یافته مجتمع) است.



پنجره پروژه: تمام فایل‌های مربوط به پروژه را نمایش می‌دهد CTRL+R

پنجره Form Layout: محل فرم را هنگام اجرا روی صفحه نمایش مشخص می‌کند.

پنجره مشخصه ها: خصوصیات شی انتخاب شده نمایش می‌دهد. اگر چند تا شی با هم انتخاب شوند خصوصیات مشترک بین آنها را نمایش می‌دهد. F4 یکی از خصوصیات که بین همه اشیا مشترک است خاصیت Name است زیرا هر شی حتما نام دارد.

پنجره مشخصه ها دارای دو زبانه Alphabetic (لیست مشخصات به ترتیب حروف الفبا) و Categorized (لیست مشخصات براساس نوع عملکرد) است.

جعبه ابزار: Tool Box شامل شی هایی است که برنامه نویس می‌تواند از آنها استفاده کند.

پنجره فرم: این پنجره فرم فعال را نمایش می‌دهد.

### مشخصه ها ، متد ها ، رویدادها:

هر شیئی (فرم - کنترل) دارای مجموعه ای از مشخصه هاست که آن را توصیف می‌کند. بعضی از این مشخصات بین کنترل ها مشترکند. (Properties) مانند مشخصات: Top - Enable - Visible - Name - Left و ....

متدها، بلوک های کد هستند که برای بیان اینکه شی در مقابل رویداد چگونه عمل کند ، در داخل شی طراحی شده اند. مانند Move - Drag - setfocus - zOrder

رویدادها، رخدادهایی در داخل و بیرون از برنامه هستند و هنگامی اجرا می‌شوند که رویدادی رخ دهد مثل

Change - Click - Gotfocus - Keypress - Keydown - Keyup - mousedown - dblclick

(سؤال) زمانی که روی یک دکمه باشیم و کلید Enter را بزنیم چه رویدادهایی رخ داده است؟

برنامه نویس می‌تواند برای هر یک از رویدادهای ذکر شده یک روال (SUB) بنویسد تا هنگامی که آن رویداد رخ داد آن کد اجرا شود (برنامه نویسی رویدادگرا)

```
Private Sub Command1_Click()
```

برنامه مربوط به کلیک دکمه

```
End Sub
```

برای مشاهده برنامه های مربوط به یک شی آن را انتخاب کرده و سپس کلید F7 را بزنید.

برای اجرای برنامه کلید F5 یا از منوی RUN گزینه Start را بزنید.

### انواع فایل ها در ویژوال بیسیک:

۱- فایل پروژه: با پسوند VBP است و محتوای آن مشخصات پروژه و نام فایل‌های فرم و فرم اصلی و ... است.

۲- فایل محیط کاری: با پسوند VBW است. محتوای آن اطلاعات محیط کاری است.

۳- فایل فرم: با پسوند FRM است و محتوای آن تمام مشخصات یک فرم و اشیا داخل آن است.

۴- فایل تصویر: با پسوند FRX است و محتوای آن تصاویر روی فرم ها یا اشیا است.

۵- فایل‌های DLL و OCX : ....

۶- فایل‌های ماژول: دارای پسوند BAS هستند و محتوای آنها کد داخل ماژول ها است.

### انواع داده ها:

داده ها در زبان بیسیک به دودسته تقسیم می‌شوند ۱- داده های عددی ۲- داده های غیر عددی

۱- نوع عددی: الف) اعداد صحیح ب) اعداد اعشاری

در ویژوال بیسیک عدد 8 و 8.0 دو عدد متفاوت هستند زیرا اولی صحیح و دومی اعشاری است.

نوع داده	میزان حافظه	محدوده	پسوند
Byte	۱ بایت	۰ تا ۲۵۵ صحیح	منفی ندارد
Integer	۲ بایت	-32768 تا 32767 صحیح	%
Long	۴ بایت	صحیح بزرگ	&
Single	۴ بایت	اعشاری با دقت معمولی	!
Double	۸ بایت	اعشاری با دقت مضاعف	#
Currency	۸ بایت	اعشاری تا چهار رقم اعشار	@

سعی کنید برای داده هایتان نوعی را انتخاب کنید که فضای کمتری اشغال کند مثلاً برای نگهداری شماره ماه های سال از کدام نوع استفاده می کنید؟

۲- نوع های غیر عددی: رشته ترکیبی از چند نویسه است و نمی توان روی آن محاسبه کرد مانند "1234" یا "ABCD5+s" رشته ها بین دو " قرار می گیرند

نوع داده	مقدار حافظه	محدوده
String طول ثابت	طول رشته	۱ تا ۶۵۴۰۰ کاراکتر
String طول متغیر	طول رشته + ۱۰ بایت	۰ تا ۲ میلیارد کاراکتر
DATE	۸ بایت	
Boolean	۲ بایت	True تا False
Object	۴ بایت	معادل شی تعریف شده
Variant عددی	16 بایت	هر نوع عددی
Variant متنی	طول رشته + ۲۲ بایت	رشته با طول متغیر

رشته ای که طول آن صفر باشد رشته NULL است. ویژوال بیسیک مقادیر شامل تاریخ و زمان را با علامت # نمایش می دهد.

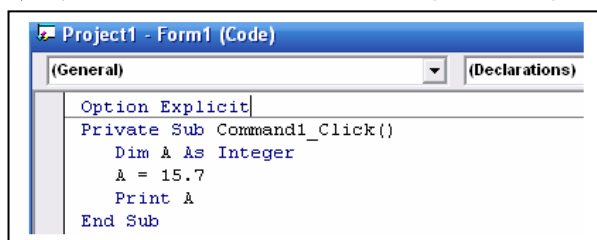
مثال #7:12 AM #

Boolean ها می توانند مقادیر YES , NO و یا صفر و یک را نیز بگیرند. نوع Variant هر مقدری به جز رشته های با طول ثابت را در خود جای می دهد.

## متغیرها: Variable

متغیر مکانی از حافظه برای نگهداری یک مقدار است. هر متغیر یک مقدار را فقط در خود نگهداری می کند. هر متغیر دارای نام و نوع است. برای تعریف یک متغیر از کلمه کلیدی DIM استفاده می شود.

دستور Option Explicit در قسمت General باعث می شود که برنامه نویس متغیرهایش را لزوماً تعریف کند در غیر اینصورت پیغام خطا صادر می شود.



یا اینکه از منوی Tools > Options گزینه

Require Variable Declaration را انتخاب کنیم.

Dim Z As Double

Dim F#

متغیرهای محلی Local:

هنگامی که یک متغیر داخل یک روال (SUB) تعریف می شود در همان روال قابل استفاده است و خارج از آن قابل استفاده نیست.

متغیرهای سراسری: Global

متغیرهایی که در بخش General تعریف می شوند و در همه روال قابل استفاده است.

نام متغیرهایی که باید طبق قوانین زیر انتخاب شود:

۱- نام متغیر باید با یک حرف انگلیسی شروع شود

۲- حرف دوم به بعد می تواند حرف انگلیسی یا عدد باشد.

۳- نام متغیر حداکثر می تواند ۲۵۵ حرفی باشد.

۴- فقط از خط زیر \_ می توانید در نام متغیر ها استفاده کنید. و جا خالی در نام متغیر درست نیست.

۵- از کلمات کلید استفاده نکنید.

(مثال) ABC 5d dim T\_S a\*b S2

```
Option Explicit
Dim s As String
Private Sub Command1_Click()
    Dim A As Integer
    s = "abcd"
    A = 15
    Print A
End Sub

Private Sub Command2_Click()
    s = "xyzw"
End Sub
```

سعی کنید نام متغیرها را با معنی انتخاب کنید. و نوع آنها را بصور مخفف در پیشوند آنها بکار ببرید. IntSum

```
Dim sum as integer , h as single
```

می توان با یک Dim چند متغیر تعریف کرد.

اگر نوع متغیر اعلام نشود پیش فرض آن Variant است.

```
Dim S1 AS string
Dim S2 AS String*5
```

متغیر S1 یک متغیر رشته ایی با طول متغیر است یعنی متغیر طول خود را متناسب با رشته تغییر می دهد.

متغیر S2 یک متغیر با طول ثابت ۵ است که رشته های با حداکثر ۵ حرفی را ذخیره می کند اگر بیشتر از ۵ حرف داخل آن بریزیم فقط ۵ حرف اول آن نگه داشته می شود.

نکته) مشخصات اشیا از نوع Variant هستند مثلا رنگ یک شی از نوع Variant است زیرا Variant همه نوعی را پذیرش می کند.

اگر داخل یک متغیر، مقداری ریخته شود که با نوع متغیر همخوانی نداشته باشد و نتواند تبدیل هم شود پیغام خطا صادر می شود.

```
Dim A as byte
Dim B as integer
Dim S as string
A=300
Let B=15.8
S=1234
S= "abcdef"
```

مقدار پیش فرض متغیرهای بولین False و متغیرهای تاریخ 00:00:00 و متغیرهای عددی 0 و متغیرهای رشته ایی یک رشته خالی NULL است.

اگر متغیری را تعریف نکنیم (و به شرط اینکه تعریف متغیرها اجباری نشده باشد) و از آن استفاده کنیم آن متغیر از نوع Variant در نظر گرفته می شود.

### ثابتهای Constant

ثابتهای مقادیری هستند که در برنامه مقدارشان تغییر نمی کند. ثابتهای عددی یا غیر عددی هستند. اگر مقدار یک ثابت را برنامه تغییر دهید پیغام خطا صادر می شود. برای تعریف ثابت از کلمه کلیدی Const استفاده می شود.

```
Const F=12
Const S=2+5 , H=12E-1
Const D="AMIN"
```

```
A=5
Const A=12
Print A
```

پیغام خطا صادر می شود

**عملگرهای ویرژوال بیسیک:**

عملگرها (Operand) در زبان بیسیک به چهار دسته تقسیم می شوند: ۱- عملگرهای محاسباتی ۲- عملگرهای رابطه ایی ۳- عملگرهای منطقی ۴- عملگرهای رشته ایی  
 ® عملگرهای محاسباتی: عملگرهای محاسباتی به ترتیب تقدم به صورت زیر است.

عملگر	نام	نوع حاصل
^	توان	اعشاری یا صحیح
-	منفی (تفریق یکانی)	صحیح یا اعشاری
* /	ضرب تقسیم	برای / اعشاری
\	تقسیم صحیح	صحیح
Mod	باقیمانده تقسیم	صحیح
- +	جمع و تفریق	صحیح یا اعشاری



نکته) در مورد عملگرهایی که تقدم یکسانی دارند عملگری که سمت چپ قرار دارد اولویت بیشتری دارد.

نکته) پراتنز تقدم بیشتری نسبت به عملگرهای محاسباتی دارد.

حاصل تقسیم / یک عدد اعشاری است اما حاصل تقسیم صحیح یک عدد صحیح است.

اما  $10/3 = 3.33333$  اما  $10\backslash 3 = 3$  است.

اگر عملوندهای تقسیم صحیح \ یا Mod اعشاری باشد ابتدا عملوند گرد می شود سپس محاسبه انجام می شود.

$$11.5 \text{ Mod } 3 = 0 \quad 11.4 \text{ Mod } 3 = 2 \quad 12.8 \text{ Mod } 3.6 = 1$$

$$12.3 \backslash 4.5 = 2$$

نکته) علامت حاصل MOD بستگی به علامت عملوند اولی دارد .

$$-21.6 \text{ Mod } 3 = -1 \quad 21.6 \text{ Mod } 3 = 1 \quad 21.6 \text{ Mod } -3 = 1$$

$$-21.6 \text{ Mod } -3 = -1$$

$$-33.7 \text{ Mod } -6 =$$

نکته) علامت حاصل \ (تقسیم صحیح) بستگی به علامت هر دو عملوند آن دارد  $(+ = - \times -)$   $(- = + \times -)$

$$12.3 \backslash -4.5 = -2 \quad -12.3 \backslash 4.5 = -2 \quad -12.3 \backslash -4.5 = 2$$

عملگر \ (تقسیم صحیح) همان عملگر Div در زبان پاسکال است.

حاصل عبارت  $32 \text{ Mod } -4 * 2 \backslash 3$  چیست؟

عملگر + دو کار مختلف انجام می دهد یکی جمع دو عدد و دیگری الحاق دو رشته.

$$12+5 \rightarrow 17$$

$$"12" + "5" \rightarrow "125"$$

$$"5" + "12" \rightarrow "512"$$

البته عملگر & نیز عمل الحاق رشته ها را انجام می دهد.

$$12+5*(9-2^2)/2 \text{ mod } 15 \backslash 4$$

**دکمه فرمان: Command Button**

متداول ترین رویداد این شی Click است.

دارای دو مشخصه منطقی Default و Cancel است که اگر Default یک دکمه True شود هنگامی که

کاربر کلید Enter را می زند رویداد Click آن اجرا می شود. و هنگامی که Cancel آن True شود با زدن کلید

Esc رویداد Click آن اجرا می شود.



مشخصه	توضیح
Enable	منطقی است و فعال یا غیر فعال بودن دکمه را مشخص می کند
Caption	متن روی دکمه است و اگر متن O&K باشد ALT+K کلید دسترسی آن می شود و زیر K خط کشیده می شود.
DisablePicture	فایلی Bmp که هنگام غیرفعال شدن نشان داده می شود
DownPicture	فایلی Bmp که هنگام فشاد دادن دکمه نمایش داده می شود
Picture	فایلی Bmp که روی دکمه است

**متد Print :**

برای نمایش متن روی فرم یا کادر تصویر Picture Box از متد Print استفاده می شود.

```
Form1.Print "ALI", "REZA"
```

```
Picture1.Print 125
```

اگر نام شی نوشته نشود پیش فرض فرم جاری است.

بجای Print می توان علامت ؟ را بکاربرد.

نکته) دستور Print اگر به تنهایی نوشته شود یک خط خالی در صفحه نمایش ایجاد می شود.

دستور Print می تواند محتوای متغیرها و ثابت ها و حاصل عبارات را روی مانیتور نمایش دهد. شکل کلی آن به صورت زیر است.

عبارت ; یا , عبارت Print

اگر بین دو عبارتی که قرار است چاپ شوند علامت *کاما* باشد دو عبارت در دو ناحیه جداگانه چاپ می شوند مثال

```
Print "AMIN" , "ALI"
```

در این مثال رشته AMIN در ناحیه ۱۴ تایی اول و رشته ALI در ناحیه ۱۴ تایی دوم چاپ می شود. منظور از - جا خالی است.

```
AMIN-----ALI
```

اگر بعد از عباراتی که قرار است چاپ شوند علامت سمی کولن ; قرار گیرد عبارات کنار هم چاپ می شوند و ناحیه ها در نظر گرفته نمی شوند.

```
Print "A";"B"
```

حرف A و B کنار هم چاپ می شوند و مکان نما به ابتدای سطر بعدی منتقل می شود.

اگر در انتهای متد Print علامت *کاما* یا سمی کولن باشد به خط بعد نمی رود.

```
Print "A","C";"D",
```

```
Print "T"
```

برای اعداد مثبت بجای علامت + جا خالی گذاشته می شود.

**توابع SPC و TAB :**

برای ایجاد فاصله بین خروجی ها از تابع SPC استفاده می شود. دقت کنید که این دو تابع فقط در متد Print قابل استفاده هستند. شکل

کلی تابع به صورت SPC(n) است که n تعداد فاصله را مشخص می کند و می تواند عددی بین صفر تا 32767 باشد .

```
Print "ABC";SPC(9);"FM"
```

```
ABC-----FM
```

```
Print "ABC",Spc(3);"SF"
```

```
ABC-----SF
```

```
Print 125;SPC(5);"A";SPC(4);"G"
```

حرف G در کدام ستون چاپ می شود؟

**تابع TAB :** بوسیله این تابع موقعیت نمایش اطلاعات را می توان مشخص کرد. این تابع فقط با متد Print بکار گرفته می شود و شکل کلی

آن بصورت Tab(n) است. N عددی است صحیح بین صفر تا 32767 .

```
Print "AB";Tab(4);"CD"
```

کاراکتر C در ستون چهارم چاپ می شود

```
Print -13;Tab(6) ; 5
```

عدد ۵ در ستون هفتم چاپ می شود دقت کنید یک جا خالی برای علامت مثبت آن در نظر گرفته می شود

نکته) منظور از Tab(n) این است که از اولین ستون به اندازه n تا جلو بروید و مکان نما را در ستون n ام قرار دهید اما منظور از Spc(n) این است که از محل جاری n تا ستون جلو بروید و مکان نما در ستون بعد از آن قرار می گیرد.

```
Print "ABCD" ; TAB(2);"MN"
```

ABCD

-MN

اگر در سطر جاری در موقعیتی که بوسیله Tab مشخص می شود عبارتی چاپ شده باشد اطلاعات در سطر بعدی و در ستون ذکر شده در Tab چاپ می شود.

اگر پس از تابع Tab علامت کاما قرار گیرد تابع Tab نادیده گرفته می شود و داده در سطر بعد نمایش داده می شود.

**متد CLS :**

به کمک این متد می توان متن یا گرافیک موجود روی فرم یا کادر تصویر را در زمان اجرا پاک کرد. این متد تاثیری بر روی اشیاء دیگر ندارد.

Form.cls

Picture1.cls



**تابع MsgBox :**

برای نمایش کادر پیام استفاده می شود.

این تابع یک پیام را نمایش می دهد و کاربر می تواند دکمه ای را انتخاب کند و در ازای دکمه انتخابی عدد برگردانده می شود.

خروجی این تابع Integer است.

```
a = MsgBox("Salam", vbYesNo + vbDefaultButton2 + vbQuestion, "ALI")
```

آرگومان اول این تابع نشان دهنده پیام کادر و آرگومان دوم که از سه قسمت تشکیل می شود دکمه ها و دکمه پیش فرض و آیکن کادر را نشان می دهند و آرگومان سوم نشان دهنده عنوان کادر است. فقط آرگومان اول اجباری است و مابقی اختیاری هستند. اگر آرگومان سوم مشخص نشود نام پروژه به عنوان عنوان استفاده می شود و پیش فرض فقط یک دکمه OK دارد.

متناظر با هر کدام از اعداد ثابت هایی تعریف شده که می توان از آنها استفاده کرد و به آنها ثابت نام دار

گفته می شود. مثال



مقدار	نام ثابت	توضیح
0	VbOkonly	فقط دکمه OK
3	VbYesNoCancel	دکمه های Yes, No, Cancel
256	VbDefaultButton2	دکمه دوم دکمه پیش فرض باشد
16	Vbcritical	آیکن پیام بحرانی

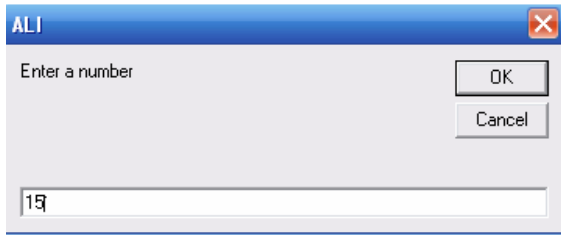
نیازی به حفظ کردن اعداد نیست و مابقی جداول را در کتاب مشاهده کنید.

```
Z = MsgBox(" Do you Want Exit", vbYesNo + vbQuestion + vbDefaultButton2, "Exit")
```

```
If Z = vbYes Then End
```

**تابع Input Box**

تابعی که پیامی را به کاربر نمایش می دهد و کاربر می تواند مقدار را وارد کند. مقدار برگشتی این تابع از نوع Variant است. برای دریافت داده از آن استفاده می شود.



`a = InputBox("Enter a number", "ALI", 15, 500, 600)`

فقط آرگومان اولی اجباری است. کاربر می تواند ۱۰۲۴ نویسه وارد کند. اگر کاربر کلید Cancel را بزند رشته ایی به طول صفر "" برمی گردد.

**عملگرهای رابطه ای (مقایسه ای):** < > = <> >= <=

نتیجه عمل ، عملگرهای رابطه ای همیشه False یا True خواهد بود.

اگر در مقایسه دو عملوند، یکی از دو عملوند NULL باشد آنگاه حاصل این مقایسه نه True و نه False می شود بلکه Null می شود.

Print 12>5

Print 13<Null

می توانیم رشته ها را مانند اعداد با هم مقایسه کرد.

"ALI" > "AX" → نادرست      "REZA" >= "ALIREZA" → درست      "ALI"="ABC" → نادرست

برای مقایسه دو رشته با یکدیگر حرف اول یک رشته با حرف اول رشته دیگر مقایسه می شود و حرف دوم با حرف دوم. برای مقایسه دو حرف با هم کد اسکی آنها مقایسه می شود. ( کد اسکی حروف کوچک از حروف بزرگ بیشتر است )  $a=97$  ,  $A=65$

"a" > "A" → درست      "ABC" >= "ABCD" → نادرست

نکته ها) حروف بزرگ، کوچک تر از حروف کوچک هستند. رقم ها کوچک تر از حروف هستند. معیار مقایسه جدول کد اسکی است.

اگر از دستور Option compare text را در قسمت تعاریف اضافه کنیم آنگاه بین حروف کوچک و بزرگ تفاوتی قائل نمی شود.

برای مقایسه کردن دو عبارت باید نوعشان با هم یکسان باشد یا اینکه قابل تبدیل به هم باشند وگرنه پیغام خطا صادر می شود.

**عملگرهای منطقی:**

```
Option Compare Text
Private Sub Command1_Click()
    a = "12" > 2
    Print a
    b = "A" < 4
    Print b
    c = "abc" = "ABC"
    Print c
End Sub
```

نام	عملگر
نقیض	Not
و	AND
یا	OR
بای انحصاری	Xor
هم ارزی	EQV
مشمول	IMP

حاصل این عملگرها True یا False است.

جدول مربوط به هر کدام از عملگرهای منطقی به صورت زیر است:

X	Y	X OR Y
T	T	T
T	F	T
F	T	T
F	F	F

جدولOR

X	Y	X And Y
T	T	T
T	F	F
F	T	F
F	F	F

جدولAnd

X	Y	X Xor Y
T	T	F
T	F	T
F	T	T
F	F	F

جدولXor

X	Y	X EQV Y
T	T	T
T	F	F
F	T	F
F	F	T

جدولEQV

X	Y	X IMP Y
T	T	T
T	F	F
F	T	T
F	F	T

جدولIMP

عملگر NOT درست را نادرست و

نادرست را درست می کند.

حاصل عملگر OR وقتی T است که حداقل یکی از عملوندهای آن T باشند.  
 حاصل عملگر AND وقتی T است که هر دو عملوندهای آن ارزش T داشته باشند.  
 حاصل عملگر XOR وقتی F است که هر دو عملوند آن ارزش یکسانی داشته باشند. در چه صورت حاصل عملگر XOR درست می شود؟  
 حاصل عملگر EQV وقتی T است که هر دو عملوند آن ارزش یکسانی داشته باشند.  
 حاصل عملگر IMP فقط در صورتی F است که عملوند اولی T و عملوند دومی F باشد.  
 عملگرهای AND و OR و EQV و XOR دارای خاصیت جابجایی است. یعنی  $A \text{ AND } B = B \text{ AND } A$

$$A \text{ XOR } B = B \text{ XOR } A \quad A \text{ EQV } B = B \text{ EQV } A$$

اما عملگر IMP خاصیت جابجایی ندارد. یعنی  $A \text{ IMP } B$  همیشه برابر  $B \text{ IMP } A$  نیست.

عملگرهای XOR و EQV عکس هم هستند یعنی  $X \text{ Xor } Y = \text{NOT}(X \text{ EQV } Y)$

حاصل عبارت روبرو کدام است؟  $\text{Print } 4 > 7 \text{ EQV } 5 > 3 \text{ Or } 8 > 10 \text{ Xor } 7 > 2$

نکته) اگر عملگرهای محاسباتی و رابطه ایی و منطقی را در یک عبارت داشته باشیم اولویت اجرا به این ترتیب است: جدول صفحه ۸۰ کتاب

۱- عملگر محاسباتی ۲- عملگر رابطه ایی ۳- عملگر منطقی

حاصل عبارت روبرو کدام است؟  $\text{Print } 4 + 2 > (3 - 1) \text{ IMP } 15 \setminus 2 * 2 > 7$

تست :

۱- پسوند نوع داده عددی Single کدام است؟

الف) !      ب) #      ج) &      د) %

۲- کدام گزینه در مورد دستور Dim صحیح است؟

الف) از دستور Dim فقط برای تعریف متغیر از نوع آرایه استفاده می شود      ب) در هر دستور Dim می توان چندین نوع متغیر تعریف کرد

ج) در هر دستور Dim فقط یک نوع متغیر می توان تعریف کرد      د) در هر دستور Dim می توان چندین متغیر فقط از یک نوع تعریف کرد

۳- ترتیب تقدم اجرای عملگرهای محاسباتی بیسیک در کدام گزینه صحیح است؟

الف) تقسیم صحیح - ضرب - باقیمانده تقسیم - تفریق یکانی      ب) تفریق یکانی - تقسیم صحیح - ضرب - باقیمانده تقسیم

ج) تفریق یکانی - ضرب - تقسیم صحیح - باقیمانده تقسیم      د) ضرب - تقسیم صحیح - باقیمانده تقسیم - تفریق یکانی

۴- داده های عددی نوع اعشاری دقت معمولی - اعشاری دقت مضاعف و صحیح بزرگ به ترتیب چند بایت از حافظه را اشغال می کنند؟

الف) 2, 4, 8      ب) 4, 8, 4      ج) 4, 6, 4      د) 4, 6, 8

۵- کدام عملگر یک عملگر رشته ای نیز هست؟

الف) +      ب) -      ج) Imp      د) \

۶- کدام عملگر روی یک عبارت عمل می کند؟

الف) Eqv      ب) IMP      ج) Not      د) OR

۷- حاصل عبارت شرطی زیر برای  $N=12$  کدام است؟

$(N \text{ MOD } 3 < 2 \text{ IMP } N < 10 \text{ XOR } A < > 0)$

الف) 1      ب) -1      ج) 0      د) پیغام خطا می دهد.

۸- در چه صورت حاصل عملگر EQV ، True خواهد شد؟

الف) هر دو عملوند True باشند.      ب) هر دو عملوند ارزش یکسان داشته باشند.

ج) حداقل یکی از عملوند True باشند.      د) یکی از عملوندها True و دیگری False باشد.

۹- حاصل عبارت  $17.51 \text{ Mod } 2 * 4 \wedge 2 \setminus 3$  کدام است؟

الف) 7      ب) 8      ج) 18      د) 5

۱۰- معادل عبارت روبرو در زبان بیسیک کدام است؟

$$Z = \frac{A+B}{B+A^2} + B$$

الف)  $Z=(A+B+B) B + A^2$  (ب)

ب)  $Z=(A+B) B + A^2$  (ب)

ج)  $Z=(A+B)/(B+A^2)+B$  (ج)

د)  $Z=(A+B)+B/B +A^2$  (د)

۱۱- معادل عبارت مقابل در بیسیک کدام است؟

$$K = A^B - \frac{4}{(B+x)^2} - 10B^A$$

الف)  $K=A^B-(4/B+x)^2 - 10 * B^A$  (الف)

ب)  $K=A^B-(4/B+x)^2-(10*B)^A$  (ب)

ج)  $K=A^B-4/(B+x)^2-(10*B)^A$  (ج)

د)  $K=A^B-4/(B+x)^2-10*B^A$  (د)

۱۲- کدام نام می تواند به عنوان نام یک متغیر محسوب شود؟

الف) A%

ب) For

ج) 2FH

د) A4

۱۳- در عبارت مقابل بالاترین و پایین ترین اولویت با کدام عملگرها می باشد؟

$$X=4-(5-2*7 \setminus 2) + 2^4$$

الف) \* و +

ب) \ و -

ج) ^ و -

د) \ و ^

۱۴- کدام نام متغیر مجاز است؟

الف) 2X

ب) X3

ج) LOOP

د) X%Y

۱۵- کدام عملگر منطقی چنانچه فقط یکی از دو عبارت منطقی درست باشد، نتیجه درست (T) را ارائه می دهد؟

الف) EQV

ب) AND

ج) XOR

د) IMP

۱۶- اولین و آخرین عملگری که در عبارت مقابل اجرا می شود به ترتیب کدام است؟

$$(5+4) / 3 \text{ Mod } (5 * (6 - 2))$$

الف) - و Mod

ب) + و Mod

ج) - و /

د) + و -

۱۷- حاصل عبارت  $156*2 \setminus 51*2 / 3 * 2$  کدام است؟

الف) 0

ب) 4

ج) 20.5

د) 40

۱۸- کدام گزینه در خصوص ترتیب تقدم عملگرها صحیح است؟

الف) ^ و / و + و <> و -

ب) ^ و \* و Mod و + و <> و -

ج) - و Mod و \ و + و \*

د) + و Mod و Eqv و + و \* و Mod و not

۱۹- عبارت  $A \text{ Imp } B$  چه وقت دارای ارزش نادرست است؟

الف)  $A=True$  و  $B=True$  (الف)  $A=True$  و  $B=False$  (ب)  $A=False$  و  $B=True$  (ج)  $A=False$  و  $B=False$  (د)

۲۰- کدام عبارت همیشه صحیح نیست؟ منظور از F نادرست (False) است و منظور از T درست است.

الف)  $A \text{ Xor } B = \text{Not } (B \text{ EQV } A)$

ب)  $A \text{ AND } F = F$

ج)  $A \text{ IMP } B = B \text{ IMP } A$

د)  $F \text{ IMP } A = T$

۲۱- اولویت انجام عملیات در عبارت روبرو را تعیین کنید؟

$$C^B + 5 / (1 + Y)^3 / 6$$

1 2 3 4 5 6

الف) 1-5-4-3-6-2 (الف)

ب) 4-1-5-3-6-2 (ب)

ج) 4-1-5-6-3-2 (ج)

د) 1-5-4-3-2-6 (د)

۲۲- حاصل عبارت روبرو کدام است؟

الف) 11

ب) 10

ج) 7

د) 3

۲۳- حاصل عبارت  $A=22 + -3^3 \setminus 2 \text{ Mod } 5$  کدام است؟

الف) 0

ب) 3

ج) 19

د) 26

۲۴- فایل محیط کاری با پسوند ..... در ویژوال بیسیک ذخیره می شود.

الف) VBP

ب) FRM

ج) FRX

د) VBW

۲۵- مهمترین خاصیت شی که تنها را دسترسی به آن هم می باشد کدام است؟

الف) Visible      ب) Caption      ج) Name      د) Value

۲۶- از کدام زبان برنامه نویسی در کاربردهای علمی و مهندسی که نیاز به محاسبات پیچیده ریاضی دارد استفاده می شود.

الف) Cobol      ب) Fortran      ج) Pascal      د) C++

۲۷- کدام گزینه نادرست است؟

الف) ویژگی هایی که حالت فعلی شی را مشخص می کند وراثت می نامند.

ب) رفتار یک شی نحوه پاسخ آن شی در مقابل رویدادهایی است که ممکن است برایش رخ دهد.

ج) روابط هر شی نشان دهنده ی ارتباط آن شی با شی های دیگر است.

د) هیچ کدام

۲۸- از کدام زبانه های موجود در کادر محاوره ای New Project جهت باز کردن پروژه ای که از قبل وجود دارد استفاده می شود؟

الف) New      ب) Existing      ج) Recent      د) Project

۲۹- کدام گزینه جز ویژگی های مشترک زبان های برنامه نویسی شی گرا نیست؟

الف) کپسوله سازی      ب) چندریختی      ج) وراثت      د) پروسه

۳۰- کمیتی که به یک رابط امکان می دهد تا برای یک کلاس عمومی از عملیات یکسانی استفاده کند را ..... نامند.

الف) چندریختی      ب) وراثت      ج) کپسوله سازی      د) مدول

۳۱- مقدار برگشتی تابع MsgBox یک مقدار از نوع ..... است.

الف) Byte      ب) integer      ج) Single      د) Boolean

۳۲- اگر در هنگام طراحی Command Button در بخش Caption عنوان R&UN قرار گیرد چه عنوانی روی کلید ظاهر می شود؟

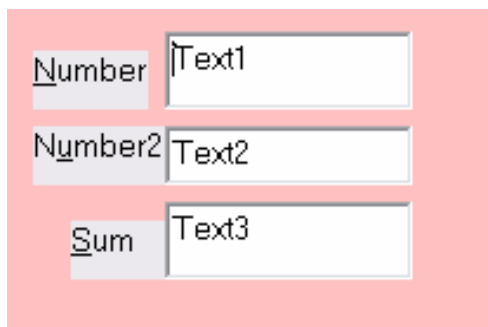
الف) RUN      ب) RUN      ج) RUN      د) R&UN

## کنترل Label:

از این کنترل برای نمایش متن استفاده می شود و این متن توسط کاربر قابل ویرایش نیست. این کنترل فوکوس دریافت نمی کند از آن می توان برای ایجاد کلید های دسترسی به سایر کنترل ها استفاده کرد.

مشخصه Caption آن متن را مشخص می کند و حداکثر ۱۰۲۴ حرف می تواند باشد. مشخصه Alignment آن ترازبندی آن را تعیین می کند.

اگر متن نوشته شده از پهنای برچسب بیشتر باشد به خط بعدی منتقل می شود مگر اینکه مشخصه AutoSize آن True باشد که در این صورت پهنای آن با اندازه متن داخل آن تنظیم می شود. برای شکستن محتوا به خط بعدی و بزرگ کردن عمودی شی مشخصه WordWrap آن را True کنید.



اگر در شکل روبرو Taborder اشیا به این ترتیب باشد Number و Text1 و

Number2 و Text2 و Text3

آنگاه کلید ALT+N باعث می شود که Text1 فوکوس را دریافت کند و ALT+U

باعث می شود فوکوس در اختیار Text2 قرار گیرد و.....

البته باید مشخصه UseMnemonic برچسب ها True باشد.

بطور کلی هر شی که دارای مشخصه Caption باشد می توان با گذاشتن علامت &

برای آن کلید دسترسی تعریف کرد.

سؤال) اگر بخواهیم در محتوای یک برچسب علامت & وجود داشته باشد چکار میکنیم؟ مثلا Ali&Reza

**کنترل Text Box :**

از این کنترل برای نمایش داده ها و دریافت داده ها از کاربر در هنگام اجرا استفاده می شود. مشخصه Locked آن باعث می شود که فوکوس را در دریافت کند اما نتوان داخل آن چیزی نوشت و فقط خواندنی می شود. حداکثر می توان 2048 نویسه وارد کرد.

(سؤال) فرق بین Enable و Locked کادر متن چیست؟

خاصیت MultiLine کادر متن باعث می شود که بتوانیم متن وارد شده را در چند سطر نمایش داد و در این حالت می توان 32K نویسه وارد کرد.

داری مشخصه ScrollBars برای نوار لغزان عمودی و افقی است.

ویژگی PasswordChar آن باعث می شود که کاربر هر حرفی را در کادرم متن تایپ می کند کاراکتر خاصی را نمایش دهد. (البته کادر متن نباید در حالت MultiLine قرار گیرد) و با این ویژگی می توان کادر متن گذرواژه ایجاد کرد.

مشخصه Maxlength حداکثر تعداد کاراکترهایی که می توان در کادر متن تایپ کرد مشخص می کند.

Password \*\*\*\*\*

**دستور IF :**

دستور If به دو دسته تقسیم می شوند ۱- شکل اول IF ..... THEN ..... ELSE ... شکل دوم IF ..... THEN ..... ELSE ... شکل اول IF به دو صورت نوشته می شود به مثالهای زیر دقت کنید:

دستورات THEN شرط(ها) IF

(نکته) در صورتی که دستورات مربوط به IF در سطرهای بعد از THEN

نوشته شوند ENDIF نیاز است. اما اگر جلوی THEN و در همان سطر باشند به

ENDIF نیاز نیست.

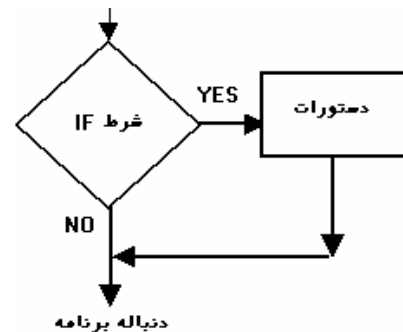
(نکته) اگر دستورات را بخواهیم جلوی Then بنویسیم آنها را با علامت : از یکدیگر جدا می کنیم.

(مثال)

THEN شرط(ها) IF

دستورات  
ENDIF

```
A=4 : B=3
IF A+B > 10 IMP B<2 THEN M=M+2 : B=8
IF B>5 AND A>6 THEN
  M=M+3
ENDIF
IF M=2 EQV Z>-2 THEN M=M+4
? M,B
```



در مثال بالا شرط اول درست است بنابراین  $M=2$  و  $B=8$  می شود اما شرط دوم نادرست است بنابراین  $M=M+3$  انجام نمی شود. شرط

```
Z=8 : Y=3
IF Z>Y+5 OR Y<4 then F=F+3 : ? F : END
IF Y*2 > 3 Then ? F+4
? Z,Y
```

سوم درست است بنابراین  $M=6$  می شود و بنابراین

اعداد 6, 8 چاپ می شود.

(نکته) دستور END در بیسیک باعث می شود که برنامه به پایان برسد.

شکل دوم دستور IF :

دستورات ۲ ELSE دستورات ۱ Then شرط(ها) IF

اگر شرط(ها) درست باشد مجموعه دستورات ۱ و اگر شرط(ها) نادرست باشد مجموعه دستورات ۲ اجرا می شود.

(نکته) اگر دستورات بیش از یکی باشد بین آنها کولن : می گذاریم.

نکته) اگر دستور IF در یک خط باشد نیازی به EndIF ندارد.

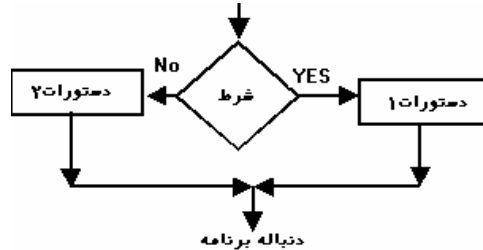
در شکل روبرو چون دستورات مربوط به IF و Else در چند خط نوشته شده اند به EndIF نیاز داریم.

نکته) کلمات IF , THEN و ELSE و EndIF کلمات ذخیره شده هستند.

مثال) خروجی چیست؟

```
IF شرط(ها) Then
    دستورات ۱
Else
    دستورات ۲
EndIF
```

```
A=14 : H=10 : Z=2
IF A>14 Xor H=Z+8 then
    M=M+3
    ? "*"
Else
    M=M+5
    ? "AMIN"
ENDIF
? M
```



```
IF شرط ۱ THEN
    دستورات ۱
ELSEIF شرط ۲ THEN
    دستورات ۲
ELSEIF شرط ۳ THEN
    دستورات ۳
ELSE
    دستورات ۴
ENDIF
```

```
A=4 : B=1
IF A<>B Then
    Print " *"
ElseIF A>B+4 THEN
    Print "$"
ELSEIF A<B+8 THEN
    Print "#"
Else
    Print "%"
```

مثال)

### IF های تودرتو: متداخل

هنگامی که بخواهیم با توجه به شرط ها از بین چند حالت فقط یک حالت را انتخاب کنیم از IF های تودرتو می توان استفاده کرد.

در شکل روبرو اگر شرط ۱ درست باشد دستورات ۱ انجام می شود و دنباله برنامه اجرا می شود و اگر شرط ۱ نادرست باشد و شرط ۲ درست باشد دستورات ۲ اجرا می شود و دنباله برنامه اجرا می شود. حال در چه صورت دستورات ۳ و دستورات ۴ اجرا می شوند؟

نکته) با توجه به اینکه TEXT کادر متن از نوع رشته است می توانیم رشته عددی را به عدد تبدیل کرد این کار بوسیله تابع Val انجام می شود.  
 $Z = \text{val}(\text{Text1.text})$

اما بطور کلی اگر تعداد انتخابها زیاد باشد(بیش از سه حالت) کنترل IF های متداخل ساده نیست و خوانایی برنامه کاهش می یابد در این حالت از دستور Select استفاده می شود. بهترین روش برای بررسی شرط های چندگانه است.

### دستور Select Case: شکل کلی

ابتدا مقدار عبارت یا متغیر جلوی Select محاسبه می شود سپس با هر کدام که برابر بود دستورات مربوط به آن را اجرا خواهد کرد.

نکته) فقط یکی از حالت ها (دستورات) برای اجرا انتخاب می شود. اگر مقدار عبارت یا متغیر جلوی Select با هیچکدام از مقادیر برابر نبود دستورات مربوط به

Case Else انجام می شود.(در صورت وجود)

نکته)قسمت Case Else اختیاری است یعنی می تواند وجود نداشته باشد که

```
Select Case متغیر یا عبارت
Case ۱ مقادیر
    دستورات ۱
Case ۲ مقادیر
    دستورات ۲
....
Case Else
    دستورات n
End Select
```

در این صورت احتمال اینکه هیچکدام از حالت ها انتخاب نشود وجود دارد.

نکته) اگر حاصل عبارت با چند مقدار (Case) مساوی شد (چند تا از Case ها برقرار شد) فقط اولین حالت (Case) اجرا می شود و بقیه اجرا نمی شوند.

```
A=2
Select Case A+3
Case 1,4,7
    Print "*"
Case 3,5
    Print "?"
Case 5,6
    Print "$"
End Select
```

(مثال) در مثال روبرو A+3 برابر 5 می شود به همین دلیل علامت ؟ چاپ می شود.

در مثال روبرو A+3 با حالت سوم نیز تطابق دارد اما \$ چاپ نمی شود.

نکته) عبارت جلوی Select می تواند صحیح یا اعشاری باشد.

نکته) مقادیر جلوی Case ها می توانند صحیح یا اعشاری یا رشته ایی یا عبارت یا متغیر باشند.

(مثال) در مثال روبرو OK چاپ میشود .

```
B=6 : A=70
Select Case A/10 - 1
Case 3
    Print "AMIN"
    Print "*"
Case B, 8.5
    Print "OK"
Case 6, B+1
    Print "Cancel"
End Select
```

(مثال)

در مثال روبرو Z+1 برابر 5.5 می شود و منظور از 1 To 8 همه اعداد صحیح و

اعشاری بین 1 تا 8 است و چون 5.5 در این محدوده قرار می گیرد کلمه AMIN

چاپ می شود و منظور از IS >=4 این است که 5.5 >= 4 است که درست است

اما # چاپ نمی شود زیر حالت قبلی انتخاب شده است.

```
Z=4.5
Select Case Z+1
Case 1 To 8
    Print "AMIN"
Case IS >= 4
    Print "#"
End Select
```

(مثال)

عدد 5 چاپ می شود.

```
M%=7.9
Select Case M%
Case is <= 7
    A=A+2
Case 1 to 8.3
    A=A+5
Case Else
    A=A+3
End Select
Print A
```

**تابع IIF() :** براساس شرط خروجی دارد

در بعضی موارد خیلی ساده می توانم بجای استفاده از دستور IF ... Else از تابع IIF استفاده کرد به صورت زیر

(حالت اشتباه , حالت درست , شرط) IIF

(مثال)

A=IIF( B>5 and C<12 , 25 , B+C)

M = IIF( X>Y , X , Y)

B= IIF ( A=0 , NULL , C/A)

می توان برای تابع IIF معادل IF ... Else نوشت اما همیشه نمی توان IF ... Else را به IIF تبدیل کرد.

IF X>Y then M=X else M=Y

**تابع Choose():** این تابع از بین چند گزینه یکی را انتخاب می کند.

تابع Choose نوع ساده شده دستور Select Case است. این تابع بر اساس مقدار آرگومان اول یکی از آرگومانهای بعدی را تحویل می دهد. (مثال)

```
A= choose( 3 , "ali" , 15 , "reza" , "mohammad",19)
```

در مثال بالا اولین آرگومان ۳ است پس سومین مقدار را که "reza" است برمی گرداند و در متغیر a می ریزد.

```
B=choose(3 , 18 , 575)
```

در این مثال مقدار سوم وجود ندارد بنابراین Null برمی گردد.

```
A=val(inputBox("Enter a number"))
```

```
S=choose(A , "Sat" , "Sun" , "Mon" , "Thu" , "Wed" , "Thr" , "Fri")
```

بنابراین آرگومان اول و دوم الزامی است و آرگومان اول باید بین ۱ تا آخرین آرگومان باشد.

نکته) در ویژوال بیسیک اگر یک خط برنامه بسیار بزرگ شود و بخواهیم دنباله خط را در خط بعدی بنویسیم در انتهای خط علامت \_ می گذاریم.

### **تابع Switch():**

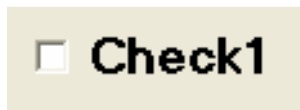
خروجی این تابع از نوع Variant است. لیستی از عبارتها را ارزیابی کرده و هر کدام که برقرار بود اولین آنها را بر می گرداند.

```
A=Switch(۱, مقدار ۱, عبارت ۱, مقدار ۲, عبارت ۲, مقدار ۳, عبارت ۳, مقدار ۴, عبارت ۴)
```

```
A=Switch( B<=5 , "Reza" , B>7 , "Ahmad" , B>10 , "ALI" , B<20 , "Mohammad")
```

هر کدام یک از عبارت ها ارزیابی می شود اولین عبارتی که درست باشد مقدار روبرو آن برگردانده می شود.

### **کنترل Check Box : کادر علامت**



از این شی برای انتخاب های Yes/No یا True/False استفاده می شود.

مشخصه Value آن تعیین می کند که آیا این گزینه انتخاب شده است یا نه .

```
If check1.value = vbchecked then
```

Value	ثابت نام دار
۰	vbUnchecked
۱	VbChecked
۲	VbGrayed

بطور پیش فرض مقدار Value صفر است.

سؤال) چگونه می توان مقدار Value کادر علامت را هنگامی که فرم اول باز می شود یک کرد؟

این شی رویداد Double\_click ندارد و اگر روی آن دابل کلیک کنیم مثل این است که دو بار، کلیک انجام شود.

چون این شی دارای خاصیت Caption است پس می توان با علامت & کلید دسترسی برای آن تعریف کنیم.

بوسیله خاصیت Style آن می توان ظاهر آن را عوض کرد و بوسیله یک Command دارای عکس می شود.

### **کنترل Option Button دکمه انتخاب:**

برای انتخاب انحصاری یک گزینه از میان چند گزینه به کار می رود.

یعنی بیش از یک دکمه انتخاب نخواهد شد و وقتی یک دکمه، انتخاب می شود بقیه دکمه ها از انتخاب خارج می شوند. به این دکمه ها، دکمه

های رادیویی Radio Button گفته می شود. بوسیله مشخصه Value آنها می توان تشخیص داد که کدام دکمه انتخاب شده است.

```
If Option1.value = True Then
```

```
.....
```

```
Elseif Option2.value=True then
```

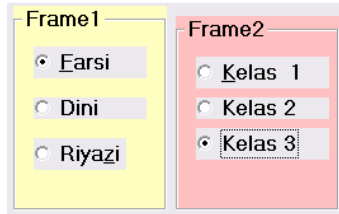
```
.....
```

```
End if
```

نکته) با خاصیت ForeColor می توان رنگ قلم شی را عوض کرد. از ثابت های نام دار VbRed , VbGreen , .... برای رنگ ها استفاده می شود. هر کدام از این ثابت ها معادل یک عدد در مبنای ۱۶ هستند. بوسیله خاصیت BackColor می توان رنگ زمینه یک شی را عوض کرد.

### کنترل Frame :

وظیفه این کنترل دسته بندی ظاهری و منطقی کنترل های دیگر است. اگر می خواهیم چند دسته دکمه انتخاب داشته باشیم آنها را با Frame دسته بندی می کنیم.



هر کدام از این Frame ها یک دسته از دکمه های انتخاب را در خود جای می دهد که انتخاب آنها بر دسته دیگر تأثیری ندارد.

### تست:

IF NOT X=3 EQV Y=5 THEN ? "ALI" ELSE ? "REZA"

۱- در چه صورت کلمه ALI چاپ می شود؟

الف) X=3 و Y=5      ب) X=4 و Y=5      ج) X=5 و Y=3      د) الف و ج

۲- خروجی برنامه روبرو کدام است؟

الف) 3      ب) 8      ج) 10      د) 5

X=3 : Y=5 : Z=7

IF X=1 AND Y>2 THEN S=S+Y  
IF Y<=6 XOR Z>10 THEN S=S+X  
IF Y>=3 OR X<=3 THEN S=S+Y  
Print S

۳- خروجی دستورات مقابل کدام است؟

الف) 2      ب) 42      ج) 42      د) 4  
12

A=4 : B=2 : C=12  
IF A>B XOR C>A+B THEN Print A;  
IF B<C THEN Print B  
IF C<A+B THEN Print A+B;

۴- خروجی برنامه مقابل کدام است؟

الف) 14      ب) 9      ج) 5      د) 4

A=4 : B=5  
IF NOT(A<B IMP A=B) THEN D=D+B  
IF B>=A+D THEN A=D+A  
IF A>D EQV B<D THEN D=A+D  
Print D

۵- حاصل اجرای برنامه روبرو کدام است؟

الف) 0      ب) 3      ج) 7      د) 17

X=4 : Y=3 : Z=10  
IF X+Y\*2<10 AND Y^2=Z-1 THEN M=M+X  
IF Y<Z-7 XOR X<10 THEN M=M+Y  
IF Y+Z=13 IMP Z=Y/4 THEN M=M+Z  
Print M

۶- خروجی برنامه روبرو کدام است؟

الف) 5      ب) 8      ج) 6      د) پیغام خطا

X=5 : Y=6 : Z=8  
IF X<Y AND X<Z THEN Print X : END  
ELSEIF Y<X AND Y<Z THEN Print Y : END  
ELSE PRINT Z  
END IF

۷- از دستور Select Case معمولا چه موقع استفاده می شود؟

الف) از IF نتوانیم استفاده کنیم (ب) تعداد انتخابها بیش از سه باشد (ج) شرط خاصی لازم نباشد (د) نیاز به سرعت بالای اجرا باشد  
 ۸- اگر  $B=7$  و  $A=6$  باشد خروجی برنامه مقابل کدامست؟

```
IF A=B THEN IF B=C THEN PRINT "A" ELSE PRINT "R" ELSE PRINT "B"
```

الف) A (ب) R (ج) B (د) هیچکدام

۹- در چه صورت شرط جلوی IF درست میشود؟  
 IF (((A AND B) OR (A AND C)) AND (C OR B)) THEN

الف) اگر A و B درست باشند (ب) اگر A و C درست باشند

ج) اگر A و B و C درست باشند (د) همه موارد

۱۰- اگر A, B و C سه عبارت شرطی باشند در چه شرایطی کلمه OK چاپ می شود؟

IF (A OR NOT B) AND (A OR B) AND C Then Print "OK"

الف) اگر A یا C درست باشند (ب) اگر هر سه درست باشند (ج) اگر A و C درست باشند (د) اگر A درست باشد

```
A=13.2
SELECT CASE A
Case 13,15,17
Print " ODD"
Case IS >13
Print " Great 13"
Case IS <12
Print " Less 12"
Case IS <10
Print " Less 10"
END SELECT
```

۱۱- خروجی برنامه روبرو کدام است؟

الف) ODD (ب) Great 13 (ج) Less 12 (د) Less 10

۱۲- حرف a در کدام ستون نمایش داده می شود؟  
 Print "b"; spc(6) ; "da"

الف) 7 (ب) 8 (ج) 9 (د) 10

۱۳- در مشخصه های Text و Label به ترتیب از راست به چپ حداکثر چند نویسه رامی توان وارد کرد؟

الف) ۱۰۲۴ و ۱۰۲۴ (ب) ۲۰۴۸ و ۱۰۲۴ (ج) ۱۰۲۴ و ۲۰۴۸ (د) ۲۰۴۸ و ۲۰۴۸

۱۴- در CheckBox اشیا ..... به صورت مستقل از هم عمل کنند ولی در Option Button ..... به صورت مستقل از هم عمل کنند.

الف) نمی توانند - می توانند (ب) نمی توانند - نمی توانند (ج) می توانند - نمی توانند (د) می توانند - می توانند

۱۵- خاصیت value شی checkbox کدام مقدار را نمی تواند دارا باشد؟

الف) Checked (ب) True (ج) Unchecked (د) Grayed

## ساختارهای تکرار:

برای اجرای دستورات به تعداد دفعات از ساختارهای تکرار استفاده می کنیم. ساختارهای تکرار (حلقه های تکرار) به دو دسته کلی تقسیم می شوند: ۱- ساختارهای تکرار معین (FOR) ۲- ساختارهای نامعین (WHILE و LOOP)

در ساختارهای معین تعداد تکرار مشخص است اما در ساختارهای نامعین تعداد تکرار مشخص نیست.

ساختارهای تکرار در زبان بیسیک سه دسته هستند: ۱- FOR ...NEXT ۲- WHILE ...WEND ۳- DO ...LOOP

### حلقه For ...NEXT:

هنگامی از حلقه For استفاده می شود که تعداد تکرار مشخص باشد شکلی این دستور به صورت زیر است:

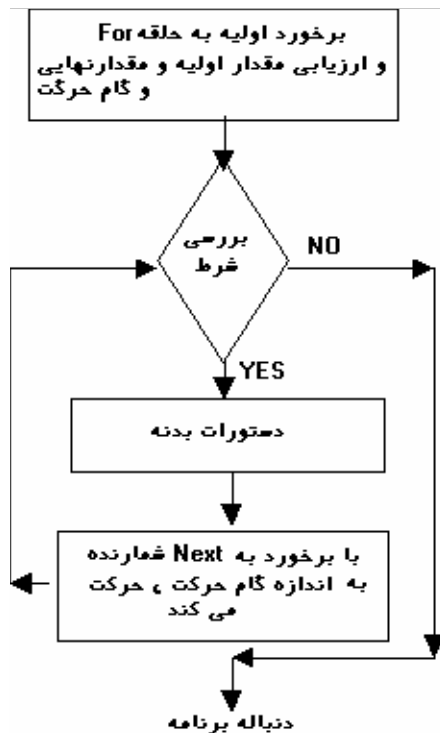
گام حرکت STEP مقدار نهایی TO مقدار اولیه = شمارنده FOR

دستورات بدنه

شمارنده NEXT

```
For A=1 TO 5 STEP 1
  PRINT "*";
  PRINT A;
NEXT A
```

(مثال)



در مثال بالا ابتدا  $A=1$  می شود و مقدار اولیه حلقه یک است و مقدار نهایی 5 است و گام حرکت 1 است و ابتدا \* چاپ می شود و بعد عدد یک چاپ می شود سپس با برخورد NEXT شمارنده که A است یک واحد (گام حرکت) افزایش می یابد و 2 می شود و چون  $2 \leq 5$  است بنابراین حلقه دوباره تکرار می شود. این عمل ادامه پیدا می کند تا اینکه  $A=5$  می شود که در این صورت نیز حلقه تکرار می شود (برای مقدار نهایی نیز حلقه یک بار تکرار می شود) و \* چاپ می شود و عدد 5 چاپ می شود با برخورد به NEXT شمارنده A یکی اضافه می شود و 6 می شود و چون  $6 \leq 5$  نیست از حلقه for خارج می شویم. بنابراین حلقه 5 بار تکرار شده است و وقتی از حلقه خارج شدیم  $A=6$  است. خروجی به صورت زیر است

\*1\*2\*3\*4\*5

نکته) اگر گام حرکت نوشته نشود به طور پیش فرض یک در نظر گرفته می شود.

نکته) اگر حلقه های متداخل نداشته باشیم می توانیم نام شمارنده را در جلوی NEXT ننویسیم.

نکته) گام حرکت و مقدار اولیه و مقدار نهایی می توانند صحیح یا اعشاری باشند

نکته) در صورتی که گام حرکت عددی مثبت باشد یک حلقه for صعودی داریم که در این صورت باید مقدار اولیه کوچکتر یا مساوی مقدار نهایی باشد در غیر این صورت حلقه For تکرار نمی شود. اما مقاردهی اولیه برای شمارنده انجام می شود.

نکته) در صورتی که گام حرکت عددی منفی باشد حلقه For نزولی داریم که در این صورت باید مقدار اولیه بزرگتر یا مساوی مقدار نهایی باشد. در غیر این صورت حلقه for تکرار نمی شود. اما مقدار دهی اولیه برای شمارنده انجام می شود.

```
For B=5.5 to -2 STEP -1
  Print B;
NEXT
```

(مثال) در مثال روبرو اعداد 5.5 و 4.5 و 3.5 و 2.5 و 1.5 و 0.5 و -0.5 و -1.5 چاپ می شود

و هنگامی که از حلقه for خارج می شویم مقدار B برابر -2.5 است.

(مثال) در مثال روبرو یک حلقه نزولی داریم اما مقدار اولیه از مقدار نهایی کوچکتر

است که در این صورت حلقه اجرا نمی شود و دستور  $Print H*2$  عدد -10

را چاپ می کند.

```
For H=-5 to -1 Step -1
  Print H
NEXT
Print H*2
```

```
For S=2.3 To 4.8 Step 0.75
  Print S
NEXT
Print S*5
```

(مثال)

در مثال روبرو 2.3, 3.05, 3.80, 4.55 چاپ می شود و وقتی از حلقه خارج می شویم 5\*5.30 چاپ می شود.

(مثال) بی نهایت \* چاپ می شود چرا؟

```
For A=1 to 10 Step 0
  Print "*"
Next
```

(مثال) فقط یک علامت @ چاپ می شود یعنی حلقه یک بار اجرا می شود.

```
For A=1 to 1 Step -1
  Print "@"
Next
```

```
B=1
For A=1 To 10 Step B
  Print A;
  B=B+1
Next
```

(مثال) در مثال روبرو گام حرکت در ابتدای ارزیابی یک است به همین دلیل تا انتهای حلقه گام حرکت یک می ماند هر چند که متغیر B در بدنه For در حال تغییر است. در این حلقه اعداد 1 تا 10 چاپ می شوند. حلقه 10 بار تکرار می شود.

نکته) گام حرکت و مقدار اولیه و مقدار نهایی حلقه فقط یک بار در ابتدای حلقه ارزیابی می شوند و دیگر تغییر نخواهند کرد. به عبارت دیگر اگر مقدار نهایی یا مقدار اولیه یا گام حرکت به صورت متغیر یا عبارت باشند و آن متغیر یا عبارت در بدنه For تغییر کند تاثیری بر مقدار ارزیابی شده اولیه ندارد

نکته) وقتی از حلقه For خارج می شویم مقدار شمارنده حلقه برابر مقدار نهایی نیست بلکه با توجه به گام حرکت، یک گام بیشتر (کمتر) است. (مثال)

```
Z=2
For S=Z+1 To 14 Step Z
  Print S
  Z=Z+6
Next S
Print Z*2
```

نکته) مقدار شمارنده حلقه می تواند در بدنه حلقه تغییر کند ( این کار توسط برنامه نویسان توصیه نمی شود) و تغییرات آن بر تعداد تکرار حلقه تاثیر می گذارد. ( در کتاب نوشته است که شمارنده حلقه For در بدنه حلقه نباید از طرف برنامه نویس تغییر کند)

```
For H=1 To 12 Step 2
  Print H;
  H=H+1
Next H
```

(مثال) در مثال روبرو در هر بار برخورد به Next دو واحد به شمارنده (H) اضافه می شود اما در بدنه حلقه نیز یک واحد به آن اضافه می شود که مجموعاً سه واحد به شمارنده اضافه شده است مانند این است که گام حرکت سه باشد.

اعداد 1,4,7,10 چاپ می شوند و وقتی از حلقه خارج می شویم مقدار شمارنده 13 است.

تمرین)

```
For N=20 To 1 Step -2
  Print N
  N=N+1
Next N
```

```
For M=1 to 10 Step 2
  M=M+2
Next
Print M
```

(مثال) حاصل اجرای برنامه روبرو کدام است؟  
الف) 12    ب) 11    ج) 13    د) 14

```
For A=1 to 10 Step 13
Print A
Next
```

مثال) حلقه یک بار تکرار می شود و عدد ۱ چاپ می شود و وقتی از حلقه خارج می شویم مقدار شمارنده 14 است.

**دستور Exit For :**

دستور Exit For باعث می شود از حلقه For خارج شویم.

```
For A=1 To 10
Print A
IF A=7 Then Exit For
M=M+3
Next A
Print A,M
```

مثال) در مثال ذکر شده حلقه For فقط هفت بار تکرار می شود زیرا وقتی A=7 می شود از حلقه For خارج می شویم. و اعداد 18, 7 چاپ می شود. چرا M=18 شده است؟

اگر چندین حلقه for متداخل داشته باشیم دستور Exit For از همان حلقه ای که داخل آن است خارج می شود.

مثال) در مثال روبرو اعداد 6 و 20 چاپ می شود چرا؟

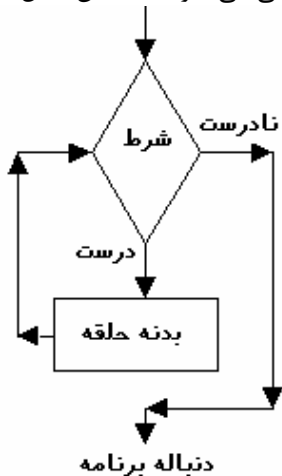
```
X=5
For K=1 To X
X=X+K
Next K
Print K\K\X" To "Z"
Print K
Next
```

نکته) شمارنده حلقه For باید عددی باشد و نمی تواند رشته ای باشد.

**حلقه While ... Wend**

هنگامی تعداد تکرار حلقه مشخص نباشد از حلقه While...Wend استفاده می شود. در این حلقه تا زمانی که شرط حلقه درست باشد حلقه تکرار می شود و زمانی که شرط حلقه نادرست شد از حلقه خارج می شود. شرط حلقه در ابتدای حلقه بررسی می شود به همین دلیل ممکن است حلقه اصلا تکرار نشود.

شرط حلقه می تواند مرکب باشد یعنی از چند شرط که باهم ترکیب شده اند تشکیل شده باشد. نکته) از حلقه While می توان به جای حلقه for استفاده کرد اما عکس آن همیشه صادق نیست. مثال)



```
A=6
S=1
While A>0
S=S*A
A=A-1
Wend
Print S
```

در این مثال 6! محاسبه و چاپ می شود. شرط ورود به حلقه درست بودن شرط حلقه است.

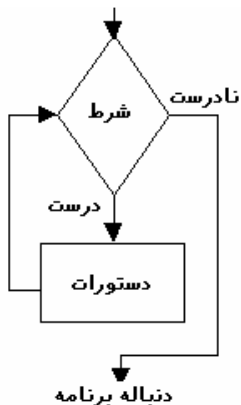
**حلقه Do ... Loop**

این حلقه نیز زمانی استفاده می شود که تعداد تکرار حلقه مشخص نباشد. حلقه Do ... Loop چهار شکل مختلف دارد که در زیر هر کدام از اشکال بررسی می شوند:

**۱- حلقه Do While ... Loop**

این حلقه شباهت زیادی به حلقه While...Wend دارد. شرط در ابتدا بررسی می شود به همین دلیل ممکن است اصلا اجرا نشود و شرط ورود به حلقه درست بودن شرط است. در ادامه فلوجارت نحوه کار این حلقه رسم می شود.

```
Do While (ها) شرط
--
دستورات
Loop
```



```
A=1
Do While A<100
Print A
A=A+2
Loop
```

مثال)

در مثال روبرو اعداد فرد بین ۱ تا ۱۰۰ چاپ می شوند. سؤال) اگر شرط حلقه به صورت A>100 بود چه چیزی چاپ می شد؟

**۲- حلقه Do Until ...Loop**

در این حلقه شرط حلقه در ابتدای حلقه بررسی می شود به همین دلیل ممکن است اصلا اجرا نشود و شرط ورود به حلقه و تکرار دستورات نادرست بودن شرط حلقه است. (برعکس حلقه قبلی) تفاوت این حلقه با حلقه Do While ...Loop در این است که حلقه ایی که Until دارد شرط تکرار آن نادرست بودن شرط است و حلقه ایی که While دارد شرط تکرار آن درست بودن شرط است.

به عبارت دیگر شرط خروج حلقه Until درست بودن شرط است و شرط خروج حلقه While نادرست بودن شرط است.

```

Do Until (شرط(ها))
--
-- دستورات
Loop
  
```

نکته) حلقه های شماره ۱ و ۲ را می توان به یکدیگر تبدیل کرد برای اینکار فقط باید شرط ها را نقیض (معکوس) کرد. برای نقیض کردن می توانید از Not استفاده کنید یا اینکه نقیض

شرط را خود بنویسید. مثلا نقیض  $A > 5$  برابر است با  $A <= 5$

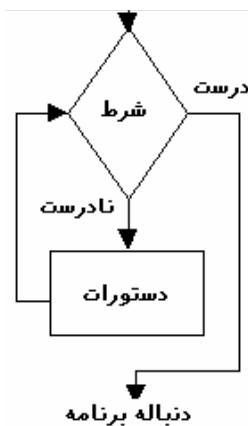
نقیض  $A >= 9$  برابر است با  $A < 9$ .

سئوال) نقیض  $A > 2$  OR  $A < 8$  چه چیزی می شود؟! (مثال)

در مثال روبرو اعداد فرد بین ۱ تا ۱۰۰ چاپ می شوند.

مثال روبرو را با مثال قبل مقایسه کنید.

چهار حلقه زیر با هم معادلند:



```

A=1
Do Until A>100
  Print A
  A=A+2
Loop
  
```

```

Do While A>10
.
.
.
Loop
  
```

```

Do Until A<=10
.
.
.
Loop
  
```

```

Do Until Not A>10
.
.
.
Loop
  
```

```

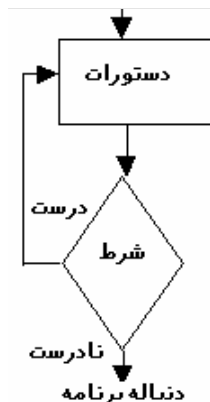
While A>10
.
.
.
Wend
  
```

**۳- حلقه Do ...Loop While**

در این حلقه شرط در انتهای حلقه بررسی می شود به همین دلیل حلقه حداقل یکبار اجرا می شود. شرط تکرار حلقه درست بودن شرط است. دقت کنید که این حلقه با حلقه شماره ۱ معادل نیست زیرا در حلقه شماره ۱ شرط در ابتدای حلقه بررسی می شود. شرط خروج از این حلقه نادرست بودن شرط است.

```

Do
--
-- دستورات
Loop While (شرط(ها))
  
```



```

A=InputBox("Enter a number")
Do
  R=A Mod 10
  A = A \ 10
  Print R;
Loop While A>0
  
```

(مثال)

در مثال روبرو رقمهای عدد ورودی چاپ می شوند.

مثلا اگر عدد ورودی ۵۴۲۸ باشد اعداد زیر به

ترتیب چاپ می شوند(چپ به راست)

8 2 4 5

اگر شرط حلقه  $A=0$  بود آنگاه فقط رقم یکان عدد ورودی چاپ می شد.

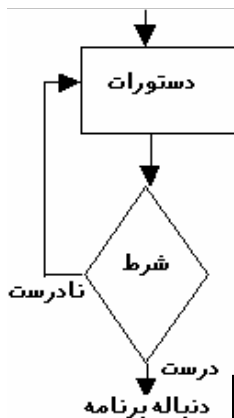
**۴- حلقه Do...Loop Until**

در این حلقه شرط در انتهای حلقه بررسی می شود پس حداقل یک بار اجرا می شود و شرط تکرار حلقه نادرست بودن شرط است. (به دلیل Until) این حلقه با حلقه شماره ۲ معادل نیست. چرا؟ شرط خروج از این حلقه درست بودن شرط است.

```
Do
--
دستورات--
شرط(ها) Loop Until
```

نکته) حلقه های شماره ۳ و ۴ را می توان به هم تبدیل کرد فقط باید شرط را نقیض کرد.

حلقه های زیر با هم معادلند.



Do . .. Loop While B <> 4	Do . .. Loop Until B = 4	Do . .. Loop Until Not B <> 4
------------------------------------	-----------------------------------	--

نکته) حلقه های شماره ۱ و ۲ را همیشه نمی توان به حلقه های شماره ۳ و ۴ تبدیل کرد. (مثال)

در مثال روبرو فقط رقم یکان عدد ورودی چاپ می شود. چرا؟ اگر شرط حلقه را  $A=0$  قرار دهیم چه چیزی چاپ می شود؟

```
A=InputBox("enter a number")
Do
R=A Mod 10
A= A \ 10
Print R;
Loop Until A>0
```

با توجه به شرایط هر کدام از حلقه ها برنامه نویسی یکی از حلقه ها را انتخاب می کند مثلاً اگر برنامه نویسی حلقه ایی بخواهد که حداقل یک بار اجرا شود و شرط خروج از حلقه نادرست بودن حلقه باشد از کدام یک استفاده خواهد کرد؟

**دستور Exit Do**

این دستور نیز شبیه Exit For است با این تفاوت برای حلقه Do...Loop بکار می رود. دستور Exit Do باعث خارج شدن حلقه Do...Loop می شود.

```
Do
A=A+1
Print A;
If A=20 then Exit Do
Print "@"
Loop Until A>100
```

دقت کنید این دستور برای حلقه While...Wend بکار نمی رود.

مثال) در مثال روبرو اعداد ۱ تا ۱۹ و به همراه @ چاپ می شوند و سپس عدد ۲۰ چاپ می شود و از حلقه خارج می شویم.

**حلقه های تکرار متداخل:**

حلقه ها می توانند متداخل باشند یعنی داخل یک حلقه، حلقه دیگری قرار گیرد. حلقه ها باید متداخل باشند نه متقاطع.

```
For A=1 to 10
For F=4 to 8
Print "*"
Next F
Next A
```

حلقه های متقاطع را مفسر بیسیک خطا خواهد گرفت.

مثال) در مثال روبرو دو حلقه For متداخل داریم البته همه حلقه از هر نوع که باشند می توانند داخل هم قرار گیرند. در این مثال 50 بار \* چاپ می شود.

اگر جای دستور Next A و Next F را عوض کنیم چه اتفاقی می افتد؟

```
Do
A=A+1
For F=A to 10
Print "*"
Loop While A<5
Next F
```

```
While B>100
Do
B=B+1
Wend
Loop Until B>50
```

تعداد تکرار داخلی ترین دستور در چند حلقه متداخل برابر است با حاصل ضرب تعداد تکرار حلقه ها.

در مثال بالا حلقه For اولی ۱۰ بار تکرار می شود و حلقه For دومی ۵ بار تکرار می شود که دستور Print مجموعاً ۵۰ بار تکرار می شود.

```
For X=1 to 5
For A=1 to 10 Step X
Print A
Next A,X
```

تست) در برنامه روبرو دستور Print چند بار اجرا می شود؟

الف) 24      ب) 22      ج) 50      د) 35

نکته) اگر بجای دستور Next A,X دستور Next X,A رابنویسیم چه اتفاقی می افتد؟

تست) خروجی برنامه روبرو کدام است؟

الف) 1 2 3 4 5 6

ب) 1 1 1 1 2 2 2 3 3 4

ج) 1 1 1 1 2 2 2 3 3 4 5 6

د) خطا

```
Do
A=A+1
For M=A To 4
Print A;
Next M
Loop Until A>6
```

تست) مقدار خروجی برنامه روبرو کدام است؟

الف) -4

ب) 6

ج) -6

د) خطا

```
X=24
For K=1 to 3 step 2
For H=8 to 2 Step -3
X=X-H
Next H,K
Print X
```

```
For A=1 to 4
For M=1 to A
For K=M to 2
Print "#"
Next K
Next M
Next A
```

مثال) دستور Print چند بار اجرا می شود؟

```
For S=4 to 1
For B=1 to 8
Print "@"
Next B,S
```

مثال) دستور Print چند بار اجرا می شود؟

الف) 32      ب) 8      ج) 4      د) صفر

تست

۱- خروجی برنامه روبرو کدام است؟

الف) -200      ب) -60      ج) -500      د) -100

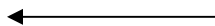
```
J=1
For K=1 to 5
For M=-10 to -5*J Step -1
For J=5 to 10
K=K*(K+1)^2+M*J
Next J,M,K
Print M*J*K
```

۲- خروجی برنامه مقابل چیست؟

الف) 1      ب) 1 2 3 4 5 6 7 8

ج) 1 3 5 7      د) 1 3 5 7 9

```
X=1
Do
Print X;
X=X+2
Loop Until X<9
```



```
For A=1 to 3
For B=1 to A
Print A;
Next B
Print B
Next A
```

جزوه برنامه نویسی یک

۳- خروجی برنامه روبرو کدام است؟

1 (د) 1 2 (ج)  
2 2 1 2 3  
3 3 3 1 2 3 4

1 2 (ب) 1 1 (الف)  
2 2 3 2 2 2  
3 3 3 4 3 3 3 3

۴- در برنامه روبرو کلمه Test چند بار چاپ می شود؟

```
For i=1 To 3
For J=6 To 1 Step -1
Print "Test"
Next J
Next i
```

18(د) 9(ج) 6(ب) 3(الف)

```
For K=1 To 9
For N=0 To 9
Print K*100+N*10+K
Next N,K
```

۵- خروجی قطعه برنامه روبرو چیست؟

(الف) چاپ اعداد سه رقمی که با مقلوب خود برابرند  
(ب) چاپ اعداد سه رقمی که رقم یکان و صدگان آنها فرد باشد  
(ج) چاپ اعداد سه رقمی که از دو طرف یکسان خوانده می شوند  
(د) الف و ج

```
For A= 1 to 5
K=K+1
For B=10 To 14
Print K;
Next A
Print
Next B
```

۶- خروجی برنامه روبرو کدام است؟  
(الف) پنج بار اعداد ۱ تا ۵ را چاپ می کند  
(ب) بیست بار اعداد ۱ تا ۵ را چاپ می کند  
(ج) اعداد ۱ تا ۲۰ را چاپ می کند  
(د) برنامه خطا دارد

```
For S=1 to 100
For W=1 To S
Print "OK"
Next W,S
```

۷- OK چند بار نمایش داده می شود؟

100(الف) 5551(ب)  
5050(د) 5550(ج)

```
A=4
For I=A To A+A
Print I
A=A+1
Next
```

۸- خروجی برنامه روبرو کدام است؟

(الف) چاپ اعداد زوج (ب) یک حلقه بی نهایت  
(ج) چاپ اعداد ۴ تا ۸ (د) خطا رخ می دهد

```
For K=1 to 100
Print "AMIN"
IF K<20 THEN Print "ALI"
Next K
```

۹- خروجی برنامه روبرو چند خط دارد؟

100(الف) 119(ج)  
120(ب) 80(د)

```
For A=10 to 20 Step -1
Print A;
Next
Print A*3
```

۱۰- خروجی برنامه روبرو کدام است؟

10 11 12 13 14 15 16 17 18 19 20 (الف)  
30(ب) 0(ج) خطا (د)

```
For H=1 To 4
For K=2 To H
Print "*";
Next K,H
```

۱۱- خروجی برنامه روبرو کدام است؟

(الف) 10 تا ستاره (ب) 6 تا ستاره (ج) 11 تا ستاره (د) خطا

```
For A=1 To 5
For B=4 To 1
X=X+B
Next B
Next A
Print X
```

۱۲- خروجی برنامه روبرو کدام است؟

5(ب) 0(الف)  
50(د) 20(ج)

۱۳- در کدام حالات استفاده از حلقه For / Next دستورات داخل آن اجرا نخواهد شد.  
 الف) اگر مقدار اولیه از مقدار نهایی بزرگتر بوده و مقدار افزایشی منفی باشد.  
 ب) اگر مقدار اولیه از مقدار نهایی بزرگتر بوده و مقدار افزایشی مثبت باشد.  
 ج) اگر مقدار اولیه از مقدار نهایی کوچکتر بوده و مقدار افزایشی مثبت باشد.

```
A=InputBox("Enter a number")
F=1
Do
    R=A Mod 10
    F=F*R
    A=A \ 10
Loop Until A=0
Print F
```

د) هر سه مورد

۱۴- خروجی برنامه روبرو کدام است؟

- الف) رقم یکان عدد ورودی      ب) حاصلضرب رقم های عدد ورودی  
 ج) مقلوب عدد ورودی      د) صفر

### توابع:

توابع داخلی، در واقع تعدادی زیربرنامه ی آماده هستند که به همراه هر زبان برنامه نویسی ارایه می شوند تا وظایف و عملیات عمومی و رایج را انجام دهند.

توابع به چهار دسته تقسیم می شوند: ۱- توابع ریاضی      ۲- توابع رشته ای      ۳- توابع تبدیلی      ۴- توابع تاریخ و زمان

### توابع ریاضی:

Ⓜ تابع ABS: قدر مطلق یک عدد را برمی گرداند. و برای محاسبه فاصله ی دو مقدار استفاده می شود.

ABS(0) → 0      ABS(12) → 12      ABS(-8.9) → 8.9      ABS(-5) → 5

نکته) اگر نوع پارامترهای ورودی برای یک تابع رعایت نشود پیغام خطا صادر می شود. ABS("ALI")

Ⓜ تابع INT: این تابع جز صحیح عدد ورودی را برمی گرداند. جزء صحیح یک عدد، بزرگترین عدد صحیح کوچکتر یا مساوی آن عدد است. در ریاضیات این تابع را با [ ] مشخص می کنند      -9 → [ -8.3 ]

عدد ورودی تابع INT سه حالت دارد: ۱- عدد ورودی مثبت باشد که در این صورت خروجی قسمت صحیح عدد ورودی است. ۲- عدد ورودی منفی و صحیح باشد که در این صورت خروجی همان عدد صحیح منفی است. ۳- عدد ورودی منفی و اعشاری باشد که در این صورت قسمت اعشار آن دور ریخته می شود و یک واحد از عدد کم می شود و برگردانده می شود. مثال:

INT(3.7) → 3      INT(5) → 5      INT(0.99) → 0      INT(-5) → -5      INT(-4.01) → -5  
 INT(-.001) → -1      INT(0) → 0      INT(-6.4) → -7      ABS(INT(-2.3)) → 3      INT(ABS(-2.3)) → 2

Ⓜ تابع FIX: این تابع قسمت صحیح یک عدد را برمی گرداند. این تابع با تابع INT متفاوت است. مثال:

FIX(3.7) → 3      FIX(4.99) → 4      FIX(0.58) → 0      FIX(-4.01) → -4      FIX(-6) → -6  
 FIX(-9.9) → -9      FIX(-5.2) → -5      FIX(ABS(-2.3)) → 2      ABS(FIX(-2.3)) → 2

سؤال) در چه صورت حاصل تابع Fix و INT یکسان خواهد بود یا به عبارتی دیگر اگر Fix(x)=INT(X) باشد X چگونه است؟

تست) دستور روبرو چه چیزی را چاپ می کند؟	Print X-Fix(X)
الف) عدد X	ب) قسمت صحیح عدد X
	ج) قسمت اعشار عدد X
	د) عدد صفر

Ⓜ تابع SQR: این تابع جذر یک عدد ورودی را برمی گرداند.

SQR(0) → 0      SQR(-9) → خطا      SQR(64) → 8      SQR(9) → 3

نکته) عدد ورودی تابع SQR نباید منفی باشد.

Ⓜ تابع SGN: علامت عدد ورودی را مشخص می کند. اگر عدد منفی باشد عدد -1 و اگر عدد مثبت باشد عدد +1 و اگر عدد ورودی

صفر باشد عدد 0 را برمی گرداند. مثال)

SGN(-4) → -1      SGN(-7.2) → -1      SGN(0) → 0      SGN(5.4) → 1      SGN(FIX(-0.78)) → 0  
 SGN(INT(-0.24)) → -1      SGN(-4+4.2) → 1

توابع  $\text{Sin}$  و  $\text{Cos}$  و  $\text{Tan}$ : این توابع یک عدد بر حسب رادیان به عنوان زاویه دریافت می کنند و نسبت مثلثاتی آن را بر می گردانند. اگر زاویه بر حسب درجه بود باید آن را به رادیان تبدیل کنید که از فرمول زیر استفاده می شود:

$$180 / (\text{زاویه درجه} \times 3.1415) = \text{زاویه بر حسب رادیان}$$

تست) برای محاسبه سینوس زاویه 30 درجه کدام گزینه صحیح است؟

الف)  $\text{Sin}(30)$       ب)  $\text{Sin}(30 * 180 / 3.1415)$       ج)  $\text{Sin}(30 * 3.1415 / 180)$       د)  $\text{Sin}(3.1415 * 180 / 30)$

نکته) خروجی این توابع یک عدد اعشاری است.

نکته) خروجی دو تابع  $\text{Sin}$  و  $\text{Cos}$  بین  $-1$  و  $+1$  است یعنی

$$-1 \leq \text{Sin}(X) \leq +1 \quad -1 \leq \text{Cos}(X) \leq +1$$

اما در مورد تابع  $\text{Tan}$  این صادق نیست.

تابع  $\text{RND}$ :

این تابع یک عدد تصادفی تولید می کند.

تابع  $\text{Log}$ :

لگاریتم طبیعی یک عدد مثبت را محاسبه می کند.

### تست:

۱- خروجی دستور روبرو کدام است؟

الف)  $-4$       ب)  $4$       ج)  $5$       د)  $-5$

Print ABS(INT(-4.2))

۲- در صورتی که عدد  $X$  یک عدد منفی صحیح باشد کدام گزینه صحیح است؟

الف)  $\text{FIX}(X) = \text{INT}(X) - 1$       ب)  $\text{INT}(X) = \text{FIX}(X) - 1$

ج)  $\text{FIX}(X) > \text{INT}(X)$       د)  $\text{FIX}(X) = \text{INT}(X)$

۳- خروجی برنامه روبرو کدام است؟

الف)  $-5$       ب)  $4$       ج)  $5$       د)  $9$

PRINT ABS(INT(4.8\*SGN(-2.5)))

۴- معادل تابع  $\text{Fix}(X)$  کدام گزینه است؟

الف)  $\text{SGN}(X) * \text{INT}(\text{ABS}(X))$       ب)  $\text{ABS}(\text{INT}(X))$       ج)  $-\text{INT}(X)$       د)  $\text{ABS}(X) * \text{INT}(X)$

### فصل پنجم:

#### رشته ها و توابع رشته ای:

نکته) پسوند نوع String علامت \$ است.

برای الحاق دو رشته از عملگرهای & یا + استفاده می شود.

سعی کنید برای جلوگیری از خطا از عملگر & استفاده کنید.

برای مقایسه دو رشته با یکدیگر می توان از عملگرهای مقایسه ایی استفاده کرد همچنین می توان از تابع Strcomp استفاده کرد.

$A = \text{strcomp}(\text{string1}, \text{string2}, \text{compare})$

اگر رشته اول از رشته دوم کوچکتر باشد خروجی  $-1$  است و اگر بزرگتر باشد  $1$  و در صورت مساوی بودن  $0$  برمی گرداند و اگر هر کدام از رشته ها Null باشد خروجی Null می شود. معیار مقایسه کد اسکی حروف است.

آرگومان compare نیز نوع مقایسه را مشخص می کند که طبق ثابت های نام دار زیر است:

توضیح	ثابت نام دار
عدم حساسیت به حروف کوچک و بزرگ A=a	vbtextcompare
حساس به حروف کوچک و بزرگ A<>a	Vbbinarycompare
عمل مقایسه را با توجه به اطلاعات پایگاه داده Access انجام می دهد	vbDatabaseCompare
نوع مقایسه را با توجه به عبارت Option Compare انجام می دهد	Vbusecompareoption

Option Compare Text مقایسه به روش متنی

Option Compare Binary مقایسه به روش دودویی

Option Compare Database

### عملگر Like :

برای مقایسه یک رشته با رشته دیگر می توان از Like استفاده کرد و امکانی است برای مقایسه یک رشته با یک الگو. در ایجاد یک الگو از کاراکترهای خاصی استفاده می شود مانند [ ! ] [ ] [ # ] [ ? ] \*

"ABCD4az" LIKE "A\*D#[a-f]z" → True  
 "123ABdsty" LIKE "#?3A[!C-H]\*"  
 "ABCDEF" LIKE "#??"

### تابع Len :

این تابع یک رشته دریافت می کند و طول رشته را به صورت یک عدد صحیح برمی گرداند.

LEN("ALI") → 3      LEN("REZA") → 4      LEN("") → 0      LEN("□□") → 2

### تابع MID :

برای انتخاب یک زیر رشته از یک رشته استفاده می شود.

MID(عدد طول , عددشروع , عبارت رشته ای)

MID("ABCDE",3,2) → "CD"      MID("ALIREZA",4,1) → "R"      MID("AMIN",1,2) → "AM"

اگر طول مشخص نشود تا انتهای رشته برگردانده می شود.

MID("COMPUTER",4) → "PUTER"      MID("ALI",2,0) → ""      MID("AMIN",3) → "IN"

برنامه روبرو یک رشته را دریافت می کند و معکوس می کند.

از MID می توان برای جایگزین کردن قسمتی از یک رشته در رشته دیگر استفاده کرد.

```
Z="ABCDEFGH"
MID(Z , 3 , 2) = "XY"
PRINT Z
```

```
s = Text1.Text
M = ""
For i = Len(s) To 1 Step -1
    C = Mid(s, i, 1)
    M = M & C
Next
Print M
```

### تابع Right و Left :

**تابع Left:** این تابع از یک رشته تعدادی از نویسه های آن را از سمت چپ برمی گرداند.

نکته) عدد n میتواند بین 0 تا 32767 باشد و باید عدد صحیح باشد (مثال)

Left("ALI", 2) → "AL"      Left("ALIREZA", 3) → "ALI"      Left("ALI",5) → "ALI□□"  
 Left("AMIN",0) → ""

**تابع Right:** این تابع از یک رشته تعدادی از نویسه های آن را از سمت راست برمی گرداند.

نکته) عدد n میتواند بین 0 تا 32767 و صحیح باشد. (مثال)

Right("ALI", 2) → "LI"      Right("ALIREZA",4) → "REZA"      Right("ALI",5) → "□□ALI"  
 Right("AMIN",0) → ""  
 Left(Right("BASIC",4),3) → "ASI"      MID("BASIC",2,3) → "ASI"

**تابع INSTR:**

برای جستجوی یک رشته در رشته دیگر از این تابع استفاده می کنیم. این تابع محل اولین وقوع را بر می گرداند. خروجی آن یک عدد صحیح است.

INSTR("ABCDEAMHAM","AM") → 6      INSTR(2,"ALIBZAALIC","ALI") → 7  
 INSTR("ABCDEF","BM") → 0      INSTR(5,"ALIREZA","ALI") → 0      INSTR("ABCDE","ABCDEF") →

نکته) پارامتر اول که یک عدد صحیح است نقطه شروع جستجو را نشان می دهد و اختیاری است اگر نوشته نشود از ابتدای رشته جستجو انجام می شود.

**تابع InStrRev:**

جستجوی یک رشته را در رشته دیگر از انتها یا هر مکانی دیگر شروع می کند. و خروجی آن مانند تابع InStr است.

A=InstrRev("ABCDEFGHXYQ","GH",10)

از کاراکتر ۱۰ ام رشته اول به سمت چپ شروع به جستجو می کند و رشته GH را جستجو می کند و حاصل ۷ خواهد شد.  
 اگر آرگومان سوم مشخص نشود به طور پیش فرض از آخر به سمت چپ جستجو را انجام می دهد.

A=InStrRev("ALIDEALIXZHM","ALI") →

B= InStr("ALIDEALIXZHM","ALI") →

**توابع Trim و RTrim و LTrim:**

تابع LTRIM و RTRIM: تابع LTRIM یک رشته دریافت می کند و فضاهای خالی سمت چپ رشته را حذف می کند و تابع RTRIM یک رشته را دریافت می کند و فضاهای خالی سمت راست رشته را حذف می کند.

LTRIM(" □□ALI□ REZA□□□□") → "ALI□REZA□□□□"      LTRIM("□□□□") → ""

RTRIM(" □□ALI□ REZA□□□□") → "□□ALI□REZA"

LTRIM(RTRIM\$("□□ALI□ REZA□□□□")) → "ALI□REZA"

تابع TRIM نیز همه جاخالی های سمت چپ و راست را حذف می کند و در واقع ترکیبی از Ltrim و RTrim است ولی جا خالی های داخل متن را بر نمی دارد.

**تابع STRING:** از این تابع برای تکرار یک رشته به تعداد مشخصی استفاده می شود.

String(5,97) → "aaaaa"

STRING(3,"ALI") → "AAA"

STRING(2,"M") → "MM"

Print Len(String(8,S))

تست) اگر رشته S دارای 5 نویسه باشد خروجی دستور روبرو کدام است؟

الف) 8      ب) 5      ج) 40      د) 13

**تابع Space:** این تابع یک رشته خالی به طول مشخص ایجاد می کند.

Space(5) → "□□□□□"      Ltrim(Space(4)) → ""      Space(1) → "□"

**تابع Replace:**

یک رشته را جستجو کرده و به جای آن، یک یا چند رشته جایگزین می کند. (بین حرف کوچک و بزرگ تفاوت قایل می شود مگر در صورتی که Option Compare Text وجود داشته باشد)

A=Replace(string1 , string2 , string3)

String2 در string1 جستجو می شود و در صورت پیدا شدن بجای آن string3 قرار می دهد.

S = "ABCDEFGHDEMNdE"

a = Replace(S, "DE", "xyz")

Print a

در این مثال DE برداشته شده و بجای آن xyz درج می شود.      ABCxyzFGHxyzMNde

A=Replace(s1, s2 , s3 , 5 , 1)

آرگومان چهارم نقطه شروع جستجو و آرگومان پنجم تعداد دفعات جایگزینی را مشخص می کند. (قبل از نقطه شروع جستجو را حذف می کند)

S = "ABCDEFGHDEMNdE"

a = Replace(S, "DE", "xyz", 2, 1)

Print a → BCxyzFGHDEMNdE

**معکوس کردن رشته: StrReverse**

a = StrReverse("abcd") → dcba

یک رشته را دریافت می کند و معکوس آن را برمی گرداند.

**تابع OCT :**

این تابع یک عدد در مبنای ۱۰ دریافت می کند و مبنای ۸ آن را برمی گرداند. خروجی یک رشته است نه یک عدد.

a = Oct(25) → "31"

a = Oct(25) + Oct(45) → "3155" الحاق انجام شده است.

**تابع HEX :**

این تابع یک عدد در مبنای ۱۰ دریافت می کند و مبنای ۱۶ آن را برمی گرداند. خروجی یک رشته است نه یک عدد.

a = Hex(158) → "9E"

**توابع تبدیلی:****تابع ASC :**

این تابع کد اسکی اولین نویسه یک رشته را برمی گرداند.

ASC("ABC") → 65

ASC("B" + "A") → 66

ASC("B")+ASC("A") → 131

ASC("aMBC") → 97

**تابع CHR :**

این تابع عکس تابع ASC است. کد اسکی یک نویسه را دریافت می کند و نویسه آن را برمی گرداند.

CHR (65) → "A"

CHR (66) → "B"

CHR (65)+CHR (66) → "AB"

CHR (65+66)=CHR (131) →

است ۱۳۱

کاراکتری که کد آن

CHR (-7) → خطا

Print CHR (7) → بیپ

Print CHR (ASC("MAHDI"))

تست(خروجی دستور روبرو کدام است؟)

77(د)

78(ج)

M(ب)

MAHDI (الف)

**تابع STR :**

این تابع یک عدد را به عنوان پارامتر ورودی دریافت می کند و آن عدد را به رشته تبدیل می کند و برمی گرداند. (مثال)

STR (124) → " 124"

STR (-2.5) → "-2.5"

STR (5+9) → " 14"

STR (5)+STR(9) → " 5 9"

STR (0.75) → " 0.75"

STR (LEN("ALI")+LEN("REZA")) → " 7"

نکته) تابع Str برای اعداد مثبت یک جا خالی در ابتدای رشته اضافه می کند.

**تابع CStr :**

مانند تابع Str عمل می کند ولی برای اعداد مثبت جاخالی نمی گذارد.

CSTR (124) → "124"

**تابع VAL :**

این تابع عکس تابع STR است و یک رشته را به عدد تبدیل می کند. خروجی عدد است

VAL("23") → 23

VAL("-4.8") → -4.8

VAL("-23P6") → -23

VAL("2.8KM3") → 2.8

VAL("21"+"9") → 219

VAL("21")+VAL("9") → 30

Val("15.9.7") → 15.9

Val("A23") → 0

نکته) اگر درون رشته جاخالی باشد ابتدا جاخالی ها را حذف می کند سپس تبدیل را انجام می دهد.

Val(" 12 5 ") → 125

**تابع UCASE و LCASE :**

این توابع یک رشته را دریافت می کنند و حروف کوچک آن را بزرگ می کنند و یا حروف بزرگ آن را کوچک می کنند.

UCASE("AbdM3#h") → "ABDM3#H"

LCASE("DBe!9+A") → "dbE!9+a"

استفاده از رویداد **KeyPress** در محدود کردن یا کنترل متن تایپ شده در **TEXT BOX** :

این رویداد دارای یک آرگومان به نام KeyAscii است که یک عدد صحیح است و کد اسکی کلید زده شده را نمایش می دهد. اگر این آرگومان با صفر مقدار دهی شود از کاراکتر زده شده صرف نظر می کند.  
برنامه روبرو فقط باعث تایپ اعداد 0 تا 9 می شود.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
KeyAscii = 0
Beep
End If
End Sub
```

نکته) دستور Beep باعث یک بوق کوچک می شود.

مشخصه Locked یک کادر متن باعث فقط خواندنی شدن کادر متن می شود و از ویرایش متن داخل آن جلوگیری می کند. True False  
نکته) برای استفاده از علامت کوتیشن " در متن باید از دوتا کوتیشن استفاده کنید.

Text1.text= "ALI""Reza" → ALI"Reza

تست:

1- خروجی دستور زیر چیست؟ Print VAL(STR(15)+CHR(ASC("42")))

الف) 154      ب) 19      ج) 1542      د) 57

۲- خروجی برنامه روبرو چیست؟

A="1234567"

B=LEFT(A,3)+MID(A,3,2)+RIGHT(A,5)

N=INSTR(5,B,"3")

PRINT LEN(B)\*N

الف) 30      ب) 40

ج) 60      د) 20

۳- خروجی برنامه مقابل کدامست؟

A\$="ALIREZA"

B\$=LEFT\$(A\$,2)

C\$=RIGHT\$(A\$,2)

D\$=MID\$(A\$,4,3)

PRINT B\$+C\$+D\$

الف) ALAZREZ      ب) ALZAREZ      ج) ALZAIREZ      د) ALAZIREZ

۴- کدام گزینه صحیح نیست؟ ( X یک عدد صحیح است و S\$ یک رشته است)

الف) Rtrim\$(Space\$(X))=String(X, " ")      ب) LEN(Right\$(S\$,X))=X

ج) INSTR(4, "UNIVERSITY", "I")=8      د) ASC("BCD")=66

۵- خروجی دستور روبرو کدام است؟ Print Str(Val("-34A2"))

الف) -34A2      ب) -34      ج) 34      د) 0

۶- کدام گزینه در مورد INT صحیح است؟

الف) بزرگترین عدد صحیح بزرگتر یا مساوی عامل تابع را بر می گرداند

ب) بزرگترین عدد صحیح کوچکتر یا مساوی عامل تابع را بر می گرداند

ج) کوچکترین عدد صحیح بزرگتر یا مساوی عامل تابع را بر می گرداند

د) کوچکترین عدد صحیح کوچکتر یا مساوی عامل تابع را بر می گرداند

۷- کدام گزینه تانژانت زاویه 45 درجه را نمایش می دهد؟

الف) TAN(45)      ب) TAN(45\*180/3.1415)      ج) TAN(45\*3.1415/180)      د) TAN(3.1415\*180/45)

۸- خروجی دستور روبرو کدام است؟ Print Instr(3,"AM","MOHAMMADI")

الف) 4      ب) 5      ج) 0      د) 1

۹- خروجی دستور روبرو کدام است؟ Print ASC(UCASE("aBcD"))+VAL("-6A5")

الف) 0      ب) 59      ج) 91      د) 65

## فصل ششم: تصویرها و ماوس و صفحه کلید

یکی از مشخصه های هر تصویر عمق رنگ Color Depth آن است (تعداد رنگ هایی که عکس پشتیبانی می کند) هر چه تعداد بیت های عمق رنگ بیشتر باشد کیفیت تصویر بیشتر است. تصویری که عمق رنگ آن ۸ بیت است ۲۵۶ رنگ (تصویر تخت) و تصویرهایی با عمق ۱۶ را تصویرهای با کیفیت بالا (۶۵۵۳۶ رنگ) و تصاویر با عمق ۲۴ بیت را تصویرهایی با رنگ واقعی True color می گویند. ویژگیهای بیسیک همه این انواع تصاویر را پشتیبانی می کند.

پسوند	توضیح
gif	تصویرهایی با ۲۵۶ رنگ یا کمتر
bmp	با عمق بین ۱ تا ۲۴ بیت
dib	با عمق بین ۱ تا ۲۴ بیت
ico	۱۶ رنگ و معمولاً با سایز ۳۲×۳۲
cur	نوعی bmp برای اشاره گر ماوس
Emf wmf	تصاویر مربوط به اشکال هندسی مثل خط و دایره
rle	تصویر فشرده
jpg	تصاویر رنگ تخت و رنگ واقعی

برای نمایش تصاویر از دو کنترل Image یا PictureBox استفاده می شود. PictureBox دارای مشخصه ها و رویدادهای بیشتری نسبت به Image است. مثل BackColor یا change

دستور زیر عکس داخل Picture1 را معکوس می کند (معکوس رنگ) پارامتر آخر عمل معکوس کردن را مشخص می کند.

**Picture1.PaintPicture** Picture1.Picture, 0, 0, , , , , vbDstInvert

دستور زیر یک عکس را که مسیر آن را مشخص کرده ایم در کنترل Picture قرار می دهد.

**Picture1.Picture = LoadPicture("D:\A.bmp")**

دستور زیر عکس داخل Picture1 را در مسیر مشخص شده ذخیره می کند.

**SavePicture** Picture1.Picture, "E:\Z.bmp"

### تفاوت کادر تصویر (Picture Box) و تصویر (Image):

Picture Box انعطاف بیشتری دارد و متدهای بیشتری را پشتیبانی می کند. اما Image کارایی بهتری دارد و در کامپیوترهای با سرعت پایین بهتر عمل می کند. هر دو دارای مشخصه Picture هستند.

سؤال) آیا متد CLS کادر تصویر عکس داخل آن را پاک می کند؟

غلط است → **Image1.Picture = "D:\B.Bmp"**

زیرا باید از تابع LoadPicture برای قرار دادن عکس در این دو کنترل استفاده کرد.

**Picture1.Picture = LoadPicture( "" )** دستور روبرو چه عملی را انجام می دهد؟

تابع LoadPicture دارای چند آرگومان دیگر نیز هست که مربوط به فایل های آیکن و اشاره گر ماوس است. صفحه ۱۹۴ کتاب در شی Image (تصویر) اگر عکسی قرار گیرد که از خود شی بزرگتر باشد به طور خودکار شی نیز به اندازه عکس بزرگ می شود مگر اینکه خاصیت Stretch آن True باشد. اما در مورد کادر تصویر PictureBox اگر عکس بزرگتر باشد قسمتی از عکس را نمایش می دهد و کادر تصویر بزرگ نمی شود. این شی خاصیت Stretch ندارد. اما خاصیت AutoSize دارد که باعث می شود شی به اندازه تصویر بزرگ شود. فرم نیز دارای خاصیتی به نام Picture است که با استفاده از LoadPicture می توان در آن عکسی را درج کرد.

### ماوس و صفحه کلید:

برنامه نویس می تواند برای هر کدام از رویدادهای ماوس برنامه بنویسد. برنامه می تواند تشخیص دهد که دکمه سمت چپ یا وسط یا راست فشار داده شده است. رویدادهای ماوس عبارتند از:

MouseMove , MouseUp, MouseMove, DblClick , DragDrop , Click, DragOver

شکل ماوس را می توان برای هر کنترلی تغییر داد. مثلاً وقتی اشاره گر ماوس بر روی یک دکمه فرمان قرار می گیرد شکل ماوس عوض شود این کار با مشخصه MousePointer انجام می شود.

Command1.MousePointer=vbHourglass

بعضی از ثابتهای نام دار برای شکل ماوس:

مقدار	ثابت	شکل
0	vbdefault	شکل معمولی ماوس
1	vbArrow	علامت +
2	VbCrosshair	علامت I
12	vbnodrop	رهاکردن ممنوع
99	Vbcustom	شکلی که با خاصیت MouseIcon اشاره می کند

اشاره گر ماوس یک آیکن ۱۶×۱۶ با پسوند Ico است.

وظیفه ایجاد رویدادهای ماوس و ارسال آنها به برنامه، بر عهده ی سیستم عامل است.

کنترل هایی که شکل ظاهری ندارند مانند Timer رویدادهای ماوس را ندارند.

نکته) سه رویداد MouseUp و MouseDown و MouseMove دارای پارامتری هستند که بوسیله آن می توان نوع دکمه زده شده را تعیین کرد.

Private Sub Form\_MouseDown(**Button** As Integer, Shift As Integer, X As Single, Y As Single)

End Sub

Private Sub Form\_MouseUp(**Button** As Integer, Shift As Integer, X As Single, Y As Single)

End Sub

Private Sub Form\_MouseMove(**Button** As Integer, Shift As Integer, X As Single, Y As Single)

End Sub

این رویدادها دارای چهار آرگومان هستند:

آرگومان **Button** اگر ۱ باشد دکمه چپ و اگر ۲ باشد دکمه راست و اگر ۴ باشد دکمه وسط زده شده است.  $1 = vbRightButton$   
آرگومان **Shift** اگر ۱ باشد کلید **Shift** و اگر ۲ باشد کلید **Ctrl** و اگر ۴ باشد کلید **Alt** به همراه فشاردادن ماوس نگه داشته ایم.

$1 = VbshiftMask$  ,  $2 = VbCtrlMask$  ,  $3 = VbAltMask$

از ثابت های نام دار هم می توانید استفاده کنید صفحه ۲۰۶ کتاب

پارامترهای **X** و **Y** مختصات محل فشاردادن ماوس را نشان می دهد.

در ازای حرکت ۱۰ تا ۱۵ Twip<sup>۱</sup> ماوس رویداد **Mouse Move** اجرا می شود. پس در ازای هر Twip اجرا نمی شود.

هنگامی که یک کلیک انجام می شود به ترتیب رویداد های زیر اجرا می شوند:

MouseDown ❖

MouseUP ❖

Click ❖

هنگامی که دابل کلیک انجام می شود به ترتیب رویدادهای زیر اجرا می شوند:

MouseDown ❖

MouseUp ❖

Click ❖

DblClick ❖

MouseUp ❖

<sup>1</sup> 1 Inch = 2.54 cm = 72 Points = 1440 twips

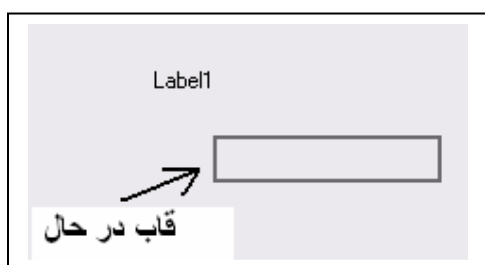
ویژوال بیسیک دارای دو مشخصه به نام CurrentX , CurrentY است که محل چاپ بوسیله Print را مشخص می کنند.  
یک برنامه نمونه:

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    CurrentX = X
    CurrentY = Y
    If Button = 2 Then Print "A"
    If Button = 4 Then Print "Z"
    If Shift = 2 And Button = vbLeftButton Then Print "H"
End Sub
```

### کشیدن و رها کردن: DragDrop

دو نوع کشیدن و رها کردن داریم: خودکار و دستی

**روش اول:** در این روش از خواص کنترل استفاده می شود. همه اشیا دارای خاصیت Dragmode هستند که امکان می دهد تا کاربر شی را



با ماوس جابجا کند و قاب شی حرکت می کند. ولی انتقال شی به عهده برنامه نویس است ابتدا خاصیت DragMode یک شی را با مقدار **Automatic** مقدار دهی کنید. سپس در رویداد Form\_DragDrop این برنامه را بنویسید:  
از متد Move استفاده می کنیم.

```
Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
    Source.Move X, Y
End Sub
```

آرگومان source نام همان کنترلی را که درگ روی آن انجام شده برمیگرداند.

نکته) در حالت درگ خودکار رویدادهای MouseDown و MouseUp غیرفعال می شوند. همچنین در این حالت رویداد DragDrop آن شی فقط هنگامی که MouseUp می شود اجرا می شود.  
رویداد DragOVER زمانی رخ می دهد یک شی را درگ کرده و بر روی شی دیگری رها کنیم.

### روش دوم:

خاصیت Dragmode را با Manual مقداردهی کنید. سپس با MouseDown شدن روی شی مربوطه عملیات کشیدن و رها کردن شروع می شود و باید از متدی به نام Drag استفاده کنید. این متد باعث شروع عملیات کشیدن و رها شدن می شود. برنامه زیر Label1 را به طور دستی حرکت می دهد:

```
Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
    Source.Move X, Y
End Sub
```

```
Private Sub Label1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Label1.Drag
End Sub
```

ثابت های مورد استفاده در متد Drag

توضیحات	مقدار	ثابت نام دار
لغو عمل کشیدن و رها کردن	۰	VbCancelDrag
شروع عمل کشیدن و رها کردن	۱	VbBeginDrag
پایان عمل کشیدن و رها کردن	۲	VbEndDrag

**رویداد های کلید:**

سه رویداد KeyPress و KeyDown و KeyUP برای صفحه کلید وجود دارد. زمانی که کلیدی فشار داده می شود رویدادهای KeyPress و KeyDown اجرا می شوند و هنگامی که کلید فشار داده شده رها می شود رویداد KeyUp اجرا می شود.

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
```

حرف کوچک و بزرگ را یکسانند و کد حرف بزرگ را نشان می دهد

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

آرگومان KeyAscii معادل کد اسکی کلید زده شده است. اگر برنامه نویس KeyAscii را تغییر دهد کلید زده شده را تغییر داده است.

```
End Sub
```

```
Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
```

حرف کوچک و بزرگ یکسانند و کد حرف بزرگ را نشان میدهد

```
End Sub
```

با توجه به رویدادهای بالا مشخص می شود که رویداد KeyPress نمی تواند تشخیص دهد که کدام یک از کلیدهای ALT و Ctrl و Shift فشار داده شده اند.

برای هر کدام از کلیدهای صفحه کلید یک ثابت نام دار در ویژوال بیسیک تعریف شده است که در صفحه ۲۱۷ کتاب مشاهده می شود. پارامتر KeyCode یک مقدار اسکی است که نشان دهنده کلید فشار داده شده است. KeyCode بین حروف کوچک و بزرگ تفاوت قابل نمی شود. از تابع CHR برای تبدیل به کاراکتر مربوطه استفاده کنید. اما در رویداد KeyPress پارامتر KeyAscii وجود دارد که بین حرف کوچک و بزرگ تفاوت قائل می شود.

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If Shift = vbAltMask Then Print "M"
    If Shift = vbCtrlMask + vbShiftMask Then Print "Z"
    If Shift = vbCtrlMask Then Print "T"
End Sub
```

می توانیم هم زمان زدن کلیدهای Ctrl+shif و حالت های دیگر را هم چک کرد.(کلیدهای ترکیبی)

نکته) رویداد KeyPress فقط در ازای فشار دادن حروف بزرگ و کوچک ، اعداد ، علائم نقطه گذاری و Enter و Tab و BackSpace اجرا می شود. مثلا در ازای زدن کلیدهای جهت نما یا Esc یا F1 تا F12 یا Home و ... اجرا نمی شود.

رویداد KeyPress نویسه هایی که دارای کد بین 0 تا 31 هستند را تشخیص نمی دهد.

اگر کاربر یک کلید را ( مثلا حرف A) فشار دهد و دستش را روی کلید نگه دارد آنگاه به دفعات رویداد KeyPress و Keydown اجرا می شود.

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii = 65 Then KeyAscii = 66
```

```
End Sub
```

نیازی نیست که کد اسکی همه کلیدها را حفظ کنید زیرا برای آنها ثابت نام دار تعریف شده است: VbKeyReturn → Enter

VbkeyBack → BackSpace

رویداد KeyDown (و KeyUp) می تواند همه کلیدهای صفحه کلید را تشخیص دهد. و آرگومان Shift آن نشان دهنده وضعیت کلیدهای Alt, Shift, Ctrl در هنگام فشار داده کلید است بنابراین کلیدهای ترکیبی را تشخیص می دهد

**مشخصه ی KeyPreview:**

اگر داخل یک فرم که حداقل یک شی (که مشاهده می شود) وجود داشته باشد در این صورت رویدادهای کلید مربوط به فرم هیچگاه اجرا نمی شوند زیرا همیشه فوکوس در اختیار این اشیا خواهد بود و رویداد های صفحه کلید آنها اجرا می شوند نه فرم. برای اینکه رویدادهای فرم نیز

اجرا شوند خاصیت KeyPreview فرم را True کنید در این صورت با زدن هر کلید ابتدا رویدادهای صفحه کلید فرم اجرا می شوند و سپس رویدادهای صفحه کلید آن شی.

معمولا از KeyPreview زمانی استفاده می شود که چندین کنترل نیاز به یک واکنش در برابر رویدادهای کلید داشته باشند.

### دستور SendKeys: ارسال ضربات کلید به برنامه

بوسیله این دستور می توانید بدون اینکه کلیدی را فشار دهید ضربات کلید را به برنامه ارسال کنید.

```
Private Sub Text1_Click()
    SendKeys "M"
End Sub
```

این برنامه باعث می شود وقتی روی Text1 کلیک می کنیم حرف M زده شود و Text1 ظاهر شود.

وقتی کلید توسط SendKeys ارسال می شود دقیقا مانند این است که کاربر کلید مربوطه را زده باشد و رویدادهای KeyDown و KeyPress و KeyDown اجرا می شوند.

برای ارسال کلیدهای خاص مانند % ( ) - + ^ باید آنها را در زوج آکولاد {} قرار دهید.

```
SendKeys "A{+}{F1}{ESC}BC"
```

کلید	علامت
Shift	+
Ctrl	^
Alt	%

```
SendKeys "+A^Z+(EC)"
```

```
SendKeys "F1"
```

یعنی حرف مربوطه ۵ بار زده شود "{A 5}"

{Right 10} یعنی کلید جهت دار راست به تعداد ۱۰ مرتبه زده شود. جدول صفحه ۲۲۵

```
SendKeys "{Enter}{+}A+R"
```

```
Sendkeys "Enter{Enter}"
```

نکته)تنها کلیدی که نمی توان با SendKeys آن را به برنامه فرستاد کلید Print Screen است.

### تست:

۱-بوسیله کدام رویداد می توان کلیدی که کاربر زده است را تغییر داد؟

الف) Key Down      ب) Key Up      ج) Key Press      د) هر سه مورد

۲-هنگامی که کلیدی فشار داده شود رویداد..... فراخوانی می شود.(ولی رها نشود)

الف) Key Down      ب) Key Press      ج) Key Up      د)الف و ب

۳-برنامه روبرو چه عملی را انجام می دهد؟

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    CurrentX = X
    CurrentY = Y
    If Shift = vbCtrlMask Then Print "A"
    If Shift = vbAltMask And Button = vbRightButton Then Print "B"
End Sub
```

الف) هنگامی که ماوس کلیک راست می شود در آن محل حرف B را چاپ می کند و زمانی که Ctrl زده می شود حرف A را چاپ می کند

ب) هنگامی که ماوس حرکت می کند و Ctrl نگه داشته شده است حرف A و در هنگام حرکت ماوس با نگه داشتن کلیک راست و Alt حرف B را چاپ می کند

ج) هنگامی که ماوس حرکت می کند و Ctrl زده می شود حرف A و در صورتی که کلید Alt زده می شود حرف B چاپ می کند.

د) هنگامی که ماوس کلیک راست می شود و کلید ALT زده می شود حرف B و در صورتی که کلید Ctrl زده می شود حرف A را چاپ می کند.

۴-برای بدست آوردن معکوس یک رشته از ..... و برای حذف جا خالی های یک رشته از ..... می توان استفاده کرد؟(راست به چپ)

الف) Strreverse و Space      ب) Instrev و Trim      ج) Trim و Strreverse      د) Instrev و Space

**Instr(1 , strReverse("TABLWZBAKD") , "BA")**

۵- حاصل عبارت روبرو کدام است؟

- الف) 7 (ب) 8 (ج) 3 (د) 4

۶- اصطلاح " خودمستندی " Self Documenting در برنامه سازی به کدام گزینه اشاره می کند؟

- الف) استفاده از نامهای توصیفی طولانی برای متغیرها (ب) قراردادن توضیحات لازم در میان برنامه  
 ج) تهیه مستندات راهنما پس از اتمام برنامه (د) همه موارد

۷- به منظور ایجاد کلید دسترسی در کنترل Label کدام خصوصیت باید تنظیم گردد؟

- الف) AccessMode (ب) KeyLink (ج) Usemnemonic (د) Appearance

۸- در رویداد MouseMove در صورتیکه همزمان با حرکت اشاره گر، کلیدهای Shift , Alt نیز فشرده شده باشد مقدار آرگومان Shift چه عددی است؟

- الف) ۵ (ب) ۳ (ج) ۶ (د) ۴

۹- کدام یک از رویدادهای زیر محل مناسبی برای قراردادن دستورات آغازین برنامه نیست؟

- الف) Form\_Load (ب) Form\_Activate (ج) Form\_Startup (د) Form\_initialize

۱۰- نتیجه اجرای دستورات روبرو چیست؟

```
Str1="year2007"
A=str1 like "?[!f-i]ar##*"
B= str1 Like "y[a-f]ar[!10-19]*"
```

الف) B=True A=False

ب) B=True A=True

ج) B=False A=True

د) B=False A=False