

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



# جزوه مبانی برنامه سازی و ویژوال بیسیک

**استاد : احمد صادقی**

**تنظیم : رامین عطاپور**

کاردانی پیوسته کامپیوتر - نرم افزار کامپیوتر

ترم اول

۱۳۹۲

# فهرست مطالب

صفحه	عنوان
۱	الگوریتم
۱۵	فلوچارت
۲۴	زبان های برنامه نویسی
۲۴	زبان برنامه نویسی VB (Visual Basic)
۲۵	ویژگیهای زبان برنامه نویسی VB
۲۶	آشنایی با محیط VB
۲۷	پروژه (Project) در VB
۲۷	پنجره های واقع در یک پروژه استاندارد VB
۳۰	انواع داده ها در VB
۳۲	متغیر ها در VB
۳۲	نحوه تعریف متغیر در VB
۳۲	قواعد نامگذاری متغیرها
۳۲	انواع متغیرهای رشته ای
۳۳	کاراکترهای تعیین نوع متغیر
۳۴	عملگرها در ویژوال بیسیک
۳۷	تقدم عملگرها
۳۸	اولویت کلی عملگرها
۳۹	خصوصیات اشیاء
۳۹	رویدادهای مربوط به اشیاء
۳۹	خصوصیات مربوط به TextBox
۴۲	رویدادهای مربوط به TextBox
۴۲	کنترل Label
۴۲	خصوصیات رایج یک Label
۴۳	تابع MsgBox()
۴۵	تابع InputBox()
۴۶	دستور شرطی IF
۴۷	دستور IF...Then...Else
۴۸	دستور ElseIf
۵۰	دستور شرطی IIF
۵۰	تابع شرطی Switch
۵۱	حلقه تکرار For

## الگوریتم

**تعریف الگوریتم:** مجموعه ای از دستورات متوالی که برای انجام عمل خاصی نوشته می شوند و پس از پایان الگوریتم ، هدف مورد نظر حاصل می شود .

### ویژگیهای الگوریتم:

۱. صریح ، واضح و خوانا باشد .
  ۲. ورودی ها و خروجی های آن مشخص باشد .
  ۳. قابلیت اجرایی داشته باشد .
  ۴. پایان پذیر باشد .
- مثال ۱)** الگوریتمی بنویسید که سه عدد را دریافت کرده و میانگین آنها را حساب کند .

۱. شروع
۲. مقدار سه متغیر  $x, y, z$  را از ورودی دریافت کن
۳.  $M \leftarrow \frac{x+y+z}{3}$
۴. چاپ کن  $M$  را .
۵. پایان

**نکته :** الگوریتم نوشته شده ، توسط کامپیوتر قابل اجرا نیست اما یک انسان می تواند با اجرای ترتیبی دستورات الگوریتم به نتیجه مورد نظر برسد .

**مثال ۲)** الگوریتم مثال ۱ را برای حالتی که ورودی های الگوریتم برابر با ۱۷ ، ۲۸ و ۶۳ باشد ، اجرا نمائید .

$$x = 17 \qquad 36 \leftarrow \frac{17+28+63}{3}$$

$$y = 28$$

$$z = 63$$

در هر الگوریتم می توانیم از اعمال اصلی استفاده کنیم :

$+$  ,  $-$  ,  $*$  ,  $/$  ,  $\% \text{ (MOD)}$

**مثال ۳)** الگوریتمی بنویسید که یک عدد را از ورودی گرفته و اگر عدد مثبت باشد کلمه Positive و اگر عدد منفی باشد کلمه Negative را در خروجی چاپ کند .

۱. شروع

۲. متغیر  $x$  را بگیر

۳. اگر  $x \geq 0$  باشد ، چاپ کن "Positive" را . برو به مرحله ۵

۴. اگر  $x < 0$  باشد ، چاپ کن "Negative" را .

۵. پایان

**مثال ۴)** الگوریتمی بنویسید که یک عدد را دریافت کرده و مشخص کند که آن عدد بر ۵ بخشپذیر است یا خیر ؟

۱. شروع

۲. متغیر  $x$  را بگیر

۳. اگر  $x \% 5 = 0$  باشد ، آنگاه چاپ کن "عدد بر ۵ بخشپذیر است" . برو به مرحله ۵

۴. چاپ کن "عدد بر ۵ بخشپذیر نیست" .

۵. پایان

**مثال ۵)** الگوریتمی بنویسید که دو عدد را از ورودی دریافت کرده و بزرگترین عدد را چاپ کند .

۱. شروع

۲. مقدار متغیرهای  $x$  ,  $y$  را بگیر

۳. اگر  $x \geq y$  باشد ، چاپ کن  $x$  را . برو به مرحله ۵

۴. چاپ کن  $y$  را .

۵. پایان

**نکته :** اگر بخواهیم مقادیر  $x$  ,  $y$  را چاپ کنیم از علامت "(" استفاده نمی کنیم .

**مثال ۶)** الگوریتمی بنویسید که یک عدد صحیح را گرفته و مقسوم علیه های آن را چاپ کند .

۱. شروع

۲. متغیر  $x$  را بگیر

۳.  $I \leftarrow 1$

۴. اگر  $x \% I = 0$  ، آنگاه  $I$  را چاپ کن

۵.  $I \leftarrow I + 1$

۶. اگر  $I \leq x$  باشد به مرحله ۴ برو

۷. پایان

جدول Trace مربوط به الگوریتم فوق را برای ورودی  $x = 5$  رسم کنید .

x	I	خروجی
۵	۱	۱
	۲	۵
	۳	
	۴	
	۵	
	۶	

**مثال ۷)** الگوریتمی بنویسید که یک عدد را دریافت کرده و مجموع مقسوم علیه های آن را چاپ کند .

۱. شروع

۲. مقدار متغیر  $x$  را بگیر

۳.  $I \leftarrow 1$  و  $S \leftarrow 0$

۴. اگر  $x \% I = 0$  ، آنگاه  $S \leftarrow S + I$

۵.  $I \leftarrow I + 1$

۶. اگر  $I \leq x$  باشد ، به مرحله ۴ برو

۷.  $S$  را چاپ کن

۸. پایان

جدول Trace مربوط به الگوریتم مثال ۷ را برای ورودی  $x = 6$  رسم کنید .

خروجی	S	I	x
۱۲	۰	۱	۶
	۱	۲	
	۳	۳	
	۶	۴	
	۱۲	۵	
		۶	
		۷	

**مثال ۸)** الگوریتمی بنویسید که عددی را از ورودی گرفته و اگر عدد اول باشد Prime را چاپ کند، در غیر اینصورت No-Prime را چاپ کند .

**تعریف عدد اول :** عددی است که غیر از یک و خودش ، هیچ مقسوم علیه دیگری نداشته باشد .

۱. شروع

۲. مقدار متغیر  $x$  را بگیر

۳.  $I \leftarrow 1$  و  $T \leftarrow 0$

۴. اگر  $x \% I = 0$ ، آنگاه  $T \leftarrow T + 1$

۵.  $I \leftarrow I + 1$

۶. اگر  $I \leq x$  باشد ، به مرحله ۴ برو

۷. اگر  $T = 2$  باشد ، چاپ کن "Prime" را . در غیر اینصورت چاپ کن "No-Prime" را .

۸. پایان

**مثال ۹)** الگوریتمی بنویسید که یک عدد را گرفته و مشخص کند که عدد کامل است یا خیر؟

**راهنمایی:** یک عدد زمانی کامل است که مجموع مقسوم علیه هایش به جز خودش، با خود عدد برابر باشد.

۱. شروع

۲. مقدار متغیر  $x$  را بگیر

۳.  $I \leftarrow 1$  و  $S \leftarrow 0$

۴. اگر  $x \% I = 0$ ، آنگاه  $S \leftarrow S + I$

۵.  $I \leftarrow I + 1$

۶. اگر  $I < x$  باشد، به مرحله ۴ برو

۷. اگر  $S = x$ ، چاپ کن "عدد کامل است" در غیر اینصورت چاپ کن "عدد کامل نیست"

۸. پایان

**مثال ۱۰)** الگوریتمی بنویسید که طول اضلاع یک مثلث را گرفته و مشخص کند مثلث قائم الزاویه هست یا خیر؟

\* یک مثلث زمانی قائم الزاویه است که مجذور یکی از اضلاع با مجموع مجذور دو ضلع دیگر برابر باشد. (رابطه فیثاغورث:  $a^2 = b^2 + c^2$ )

۱. شروع

۲.  $a, b, c$  را بگیر

۳. اگر  $a^2 = b^2 + c^2$  یا  $b^2 = a^2 + c^2$  یا  $c^2 = a^2 + b^2$  آنگاه چاپ کن "مثلث قائم الزاویه است"، در

غیر اینصورت چاپ کن "مثلث قائم الزاویه نیست".

۴. پایان

**مثال ۱۱)** الگوریتمی بنویسید که مجموع اعداد فرد دو رقمی را حساب کرده و حاصل را چاپ کند .

روش اول :

روش دوم :

- |  |   |
|--|---|
| ۱. شروع  | ۱. شروع                                   |
| ۲. $I \leftarrow 11$ و $S \leftarrow 0$            | ۲. $I \leftarrow 11$ و $S \leftarrow 0$   |
| ۳. اگر $I \bmod 2 \neq 0$ باشد، $S \leftarrow S+I$ | ۳. $S \leftarrow S+I$                     |
| ۴. $I \leftarrow I+1$                              | ۴. $I \leftarrow I+2$                     |
| ۵. اگر $I \leq 99$ ، آنگاه به مرحله ۳ برو          | ۵. اگر $I \leq 99$ ، آنگاه به مرحله ۳ برو |
| ۶. چاپ کن S را                                     | ۶. چاپ کن S را                            |
| ۷. پایان   | ۷. پایان                                  |

**مثال ۱۲)** الگوریتمی بنویسید که مجموع اعداد سه رقمی زوج را حساب کرده و حاصل را چاپ کند .

۱. شروع
۲.  $I \leftarrow 100$  و  $S \leftarrow 0$
۳.  $S \leftarrow S+I$
۴.  $I \leftarrow I+2$
۵. اگر  $I \leq 998$  باشد، آنگاه به مرحله ۳ برو
۶. چاپ کن S را
۷. پایان

**مثال ۱۳)** الگوریتمی بنویسید که شعاع یک دایره را گرفته و محیط و مساحت آن را چاپ کند .

\* محیط دایره :  $2\pi r$       مساحت دایره :  $\pi r^2$

۱. شروع
۲. شعاع دایره را در متغیر  $r$  قرار بده
۳.  $\pi \leftarrow 3/14$
۴.  $P \leftarrow 2\pi r$
۵.  $S \leftarrow \pi r^2$
۶. چاپ کن " P = محیط " و " S = مساحت "
۷. پایان



**مثال ۱۴)** الگوریتمی بنویسید که ضرایب یک معادله درجه اول ( $ax+b=0$ ) را گرفته و ریشه آن را حساب کند .

۱. شروع

۲. ضرایب  $a, b$  را بگیر

۳.  $x \leftarrow -b/a$

۴. چاپ کن  $x$  را

۵. پایان

**مثال ۱۵)** الگوریتمی بنویسید که ضرایب معادله درجه دوم ( $ax^2+bx+c=0$ ) را گرفته و ریشه های آن را حساب کند .

۱. شروع

۲. ضرایب  $a, b, c$  را بگیر

۳.  $\Delta \leftarrow b^2-4ac$

۴. اگر  $\Delta = 0$  ، آنگاه چاپ کن "معادله ریشه مضاعف دارد که برابر  $-b/2a$  است" و به

مرحله ۷ برو

۵. اگر  $\Delta > 0$  ، آنگاه  $x_1 = \frac{-b-\sqrt{\Delta}}{2a}$  و  $x_2 = \frac{-b+\sqrt{\Delta}}{2a}$  . چاپ کن  $x_1$  و  $x_2$  را و به مرحله ۷ برو

۶. اگر  $\Delta < 0$  ، آنگاه چاپ کن "معادله ریشه حقیقی ندارد"

۷. پایان

جدول Trace مربوط به الگوریتم فوق را برای معادله  $3x^2-5x+2$  رسم کنید .

$$3x^2-5x+2$$

$$\Delta = b^2 - 4ac$$

$$\Delta = 25 - 24 = 1$$

$$x_1 = \frac{-b-\sqrt{\Delta}}{2a} = \frac{5-1}{6} = \frac{4}{6} = \frac{2}{3}$$

$$x_2 = \frac{-b+\sqrt{\Delta}}{2a} = \frac{5+1}{6} = \frac{6}{6} = 1$$

خروجی	$x_2$	$x_1$	$\Delta$	$c$	$b$	$a$
$\frac{2}{3}$	1	$\frac{2}{3}$	1	2	-5	3
1						

**مثال ۱۶)** الگوریتمی بنویسید که زمان را برحسب ساعت ، دقیقه و ثانیه گرفته و مشخص کند زمان داده شده کلاً چند ثانیه است ؟

۱. شروع

۲. مقادیر H (ساعت) ، M (دقیقه) و S (ثانیه) را دریافت کن

۳.  $T \leftarrow (H*3600)+(M*60)+S$

۴. چاپ کن مقدار T را

۵. پایان

جدول Trace مربوط به الگوریتم فوق را برای زمان ۷ : ۳۸' : ۲۶" رسم کنید .

H = 7					
M = 38					
S = 26					
$T=(7*3600)+(38*60)+26 = 27506$					
	H	M	S	T	خروجی
	7	38	26	27506	27506

**مثال ۱۷)** الگوریتمی بنویسید که زمان را برحسب ثانیه گرفته و مشخص نماید چند ساعت و چند دقیقه و چند ثانیه است؟

۱. شروع

۲. T را برحسب ثانیه بگیر

۳.  $H \leftarrow T/3600$

۴.  $T \leftarrow T\%3600$

۵.  $M \leftarrow T/60$

۶.  $S \leftarrow T\%60$

۷. چاپ کن H,M,S را

۸. پایان

**مثال ۱۸)** فرض کنید مبنای محاسبه عوارض (مالیات) حقوق یک کارمند بصورت زیر باشد :

الف) اگر حقوق کمتر از ۵۰۰۰۰۰ تومان باشد ، مالیاتی تعلق نمی گیرد .

ب) اگر حقوق بین ۵۰۰۰۰۰ و ۱۰۰۰۰۰۰ تومان باشد ، بر مازاد حقوق ، ۱۰ درصد مالیت تعلق می گیرد .

ج) اگر حقوق بیشتر از ۱۰۰۰۰۰۰ تومان باشد ، بر مازاد حقوق ۱۵ درصد مالیات تعلق می گیرد .

الگوریتمی بنویسید که حقوق یک کارمند را برحسب تومان گرفته و میزان مالیات حقوق را حساب نماید .

۱. شروع

۲.  $S$  (حقوق یک کارمند) را برحسب تومان بگیر

۳. اگر  $S < 500,000$  ، آنگاه  $T \leftarrow 0$  و برو به مرحله ۶

۴. اگر  $S < 1,000,000$  ، آنگاه  $T \leftarrow (S-500,000) * \frac{1}{10}$  و برو به مرحله ۶

۵. در غیر اینصورت  $T \leftarrow (S-500,000) * \frac{15}{100}$

۶. چاپ کن  $T$  را

۷. پایان

جدول Trace مربوط به الگوریتم فوق را به ازای حقوق ۷۸۵۰۰۰ تومان رسم کنید .

$$S = 785,000$$

$$S < 1,000,000$$

$$T = (785,000 - 500,000) * \frac{1}{10} = 28,500$$

S	T	خروجی
785,000	28,500	28,500

**مثال ۱۹)** الگوریتمی بنویسید که ابتدا تعداد درسهای انتخابی یک دانشجو در یک ترم را گرفته و پس از آن تعداد واحد هر درس و نمره آن درس را از ورودی دریافت کند و معدل دانشجو را در خروجی چاپ نماید .

۱. شروع

۲. تعداد درسهای انتخابی دانشجو را در D قرار بده

۳.  $M \leftarrow 0$  (جمع نمرات) و  $T \leftarrow 0$  (جمع تعداد واحدها) و  $I \leftarrow 0$  (شمارنده تعداد درس)

۴. N (نمره) را دریافت کن

۵. V (تعداد واحد درس) را دریافت کن

۶.  $M \leftarrow M + (N * V)$

۷.  $T \leftarrow T + V$

۸.  $I \leftarrow I + 1$

۹. اگر  $I < D$  ، آنگاه برگرد به مرحله ۴

۱۰.  $A \leftarrow M / T$

۱۱. چاپ کن A را

۱۲. پایان

جدول Trace مربوط به الگوریتم فوق را برای داده های زیر رسم کنید .

درس	واحد	نمره					
فیزیک	۳	۱۰					
ریاضی	۳	۸					
مبانی کامپیوتر	۳	۱۸					
تربیت بدنی	۱	۲۰					
خروجی	A	V	N	I	T	M	D
12/80	12/80	3	10	0	0	0	4
		3	8	1	3	30	
		3	18	2	6	54	
		1	20	3	9	108	
				4	10	128	

**مثال ۲۰) الگوریتمی بنویسید که :**

- الف) شماره روز را گرفته و مشخص کند چند ماه و چند هفته و چند روز است؟  
 ب) اگر روز اول سال سه شنبه باشد ، روز متناظر با عدد وارد شده چه روزی است؟

**الف :**

۱. شروع
۲. تعداد روزها را در متغیر D قرار بده
۳. اگر  $D \leq 186$  ، آنگاه  $M \leftarrow D/31$  و  $D \leftarrow D\%31$  و برو به مرحله ۷
۴.  $N \leftarrow (D-186)/30$  (تعداد ماههای ۳۰ روزه)
۵.  $M \leftarrow N+6$  (تعداد کل ماهها)
۶.  $D \leftarrow D - ((N*30) + (6*31))$
۷.  $W \leftarrow D/7$
۸.  $D \leftarrow D\%7$
۹. چاپ کن M,W,D را
۱۰. پایان

**ب :**

۱. شروع
۲. تعداد روزها را در متغیر D قرار بده
۳. اگر  $D\%7=0$  ، آنگاه چاپ کن "سه شنبه است" و برو به مرحله ۱۰
۴. اگر  $D\%7=1$  ، آنگاه چاپ کن "چهارشنبه است" و برو به مرحله ۱۰
۵. اگر  $D\%7=2$  ، آنگاه چاپ کن "پنجشنبه است" و برو به مرحله ۱۰
۶. اگر  $D\%7=3$  ، آنگاه چاپ کن "جمعه است" و برو به مرحله ۱۰
۷. اگر  $D\%7=4$  ، آنگاه چاپ کن "شنبه است" و برو به مرحله ۱۰
۸. اگر  $D\%7=5$  ، آنگاه چاپ کن "یکشنبه است" و برو به مرحله ۱۰
۹. اگر  $D\%7=6$  ، آنگاه چاپ کن "دوشنبه است".
۱۰. پایان

جدول Trace مربوط به قسمت الف الگوریتم مثال ۲۰ را برای روزهای ۱۵۰ و ۲۷۳ رسم کنید .

$$D=150$$

$$D \leq 186, M=D/31$$

$$M=150/31=4$$

$$D \% 31 = D = 26$$

$$W=D/7, W=26/7=3$$

$$D \% 7 = D = 5$$

D	M	W	خروجی M,W,D
150 26 5	4	3	4,3,5

$$D=273$$

$$D > 186, N=(D-186)/30$$

$$N=(273-186)/30=2$$

$$M=N+6=2+6=8$$

$$D=D-((N*30)+(6*31))$$

$$D=273-((2*30)+(6*31))=27$$

$$W=D/7=27/7=3$$

$$D \% 7 = D = 6$$

D	M	N	W	خروجی M,W,D
273 27 6	8	2	3	8,3,6

**مثال ۲۱)** الگوریتمی بنویسید که بزرگترین مقسوم علیه مشترک دو عدد  $M$  و  $N$  را به روش نردبانی بدست آورد .

۱. شروع

۲. مقدار  $M$  و  $N$  را بگیر ( با فرض اینکه  $M$  از  $N$  بزرگتر است . )

۳.  $R \leftarrow M \% N$

۴. اگر  $R = 0$  ، چاپ کن  $N$  را و برو به مرحله ۶

۵.  $M \leftarrow N$  و  $N \leftarrow R$  و برو به مرحله ۳

۶. پایان

**مثال ۲۲)** الگوریتمی بنویسید که یک عدد را گرفته و فاکتوریل آن را حساب کند .

$$N! = 1 * 2 * 3 * \dots * N$$

۱. شروع

۲. مقدار  $X$  را بگیر

۳.  $I \leftarrow 1$  و  $F \leftarrow 1$

۴.  $F \leftarrow X * I$

۵.  $I \leftarrow I + 1$

۶. اگر  $I \leq X$  آنگاه برو به مرحله ۴

۷. چاپ کن  $F$  را

۸. پایان

**مثال ۲۳)** الگوریتمی بنویسید که حاصل ضرب  $X*Y$  را با استفاده از جمع انجام دهد .

$$X*Y = \frac{Y+Y+Y+\dots+Y}{X \text{ بار}}$$

۱. شروع

۲. مقدار  $X$  و  $Y$  را بگیر

۳.  $I \leftarrow 0$  و  $S \leftarrow 0$

۴.  $S \leftarrow S+Y$  و  $I \leftarrow I+1$

۵. اگر  $I < X$  آنگاه برو به مرحله ۴

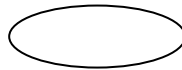
۶. چاپ کن  $S$  را

۷. پایان



## فلوچارت (Flowchart) یا نمودار گردش :

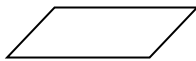
رسم فلوچارت ، یکی از روشهای نشان دادن حل مسئله با استفاده از اشکال ریاضی و هندسی است .  
 زیرا شکل مطالب را بهتر از متن انتقال می دهد و فهم و درک آن آسانتر است . برای نشان دادن  
 مراحل حل یک مسئله ، طبق قرار داد از اشکال زیر استفاده می کنیم :



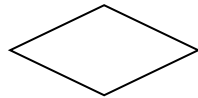
**بیضی :** برای نشان دادن حالت شروع و پایان



**مستطیل :** برای نشان دادن دستوراتی که حاوی اعمال محاسباتی هستند .



**متوازی الاضلاع :** برای دستوراتی که مقداری را از ورودی دریافت می کنند .



**لوزی :** برای نشان دادن شرط های موجود در یک الگوریتم



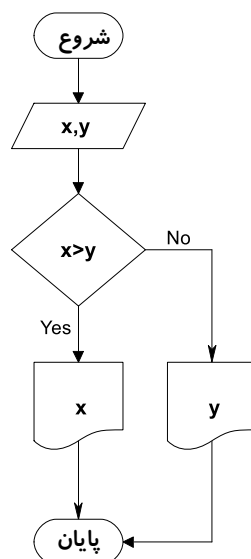
**فلش های جهت دار :** برای اتصال اشکال مختلف و نیز نشان دادن حلقه ها



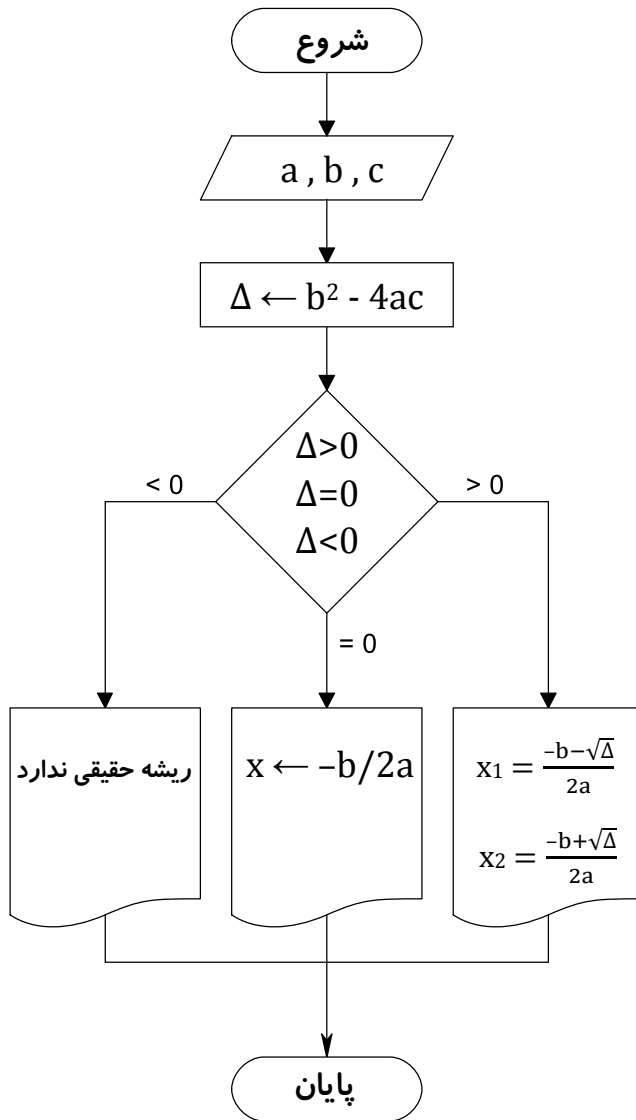
و از شکل روبرو برای **نشان دادن یک خروجی** استفاده می شود .

**مثال ۱)** فلوچارتی رسم نمائید که دو عدد را از ورودی گرفته و بزرگترین عدد را در خروجی چاپ

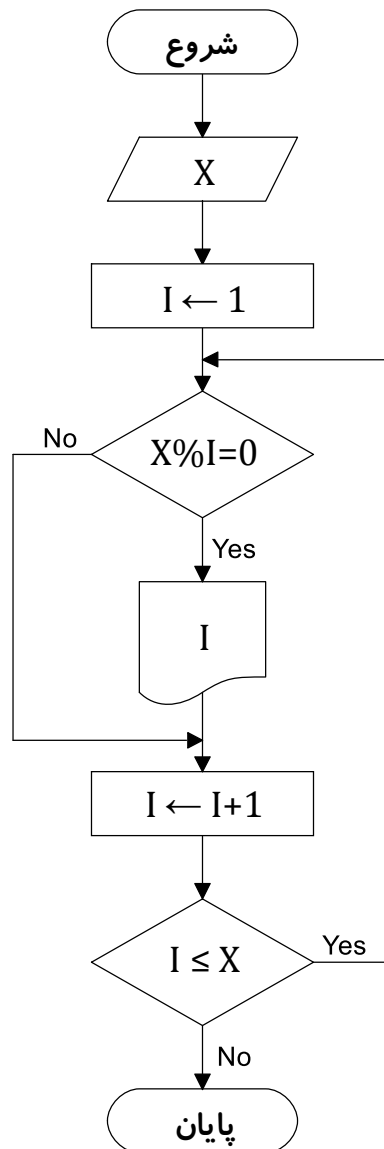
کند . ( مربوط به **مثال ۵** الگوریتم )



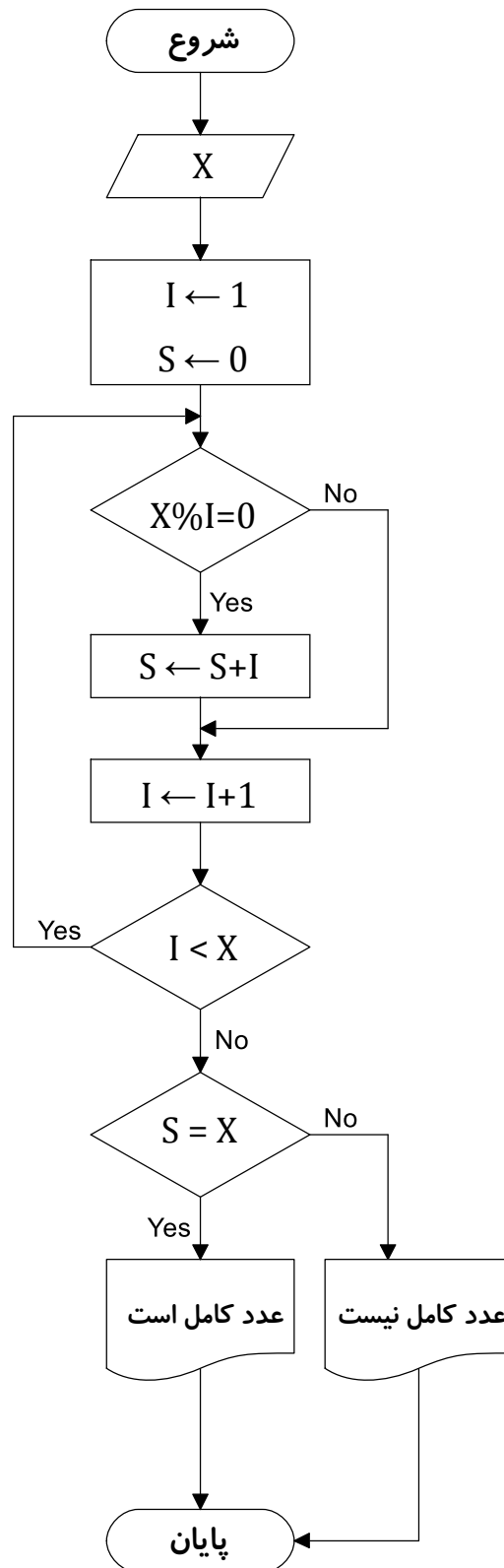
**مثال ۲)** فلوجارتهی رسم نمائید که ضرایب معادله درجه دوم  $(ax^2+bx+c=0)$  را گرفته و ریشه های آن را چاپ کند . ( مربوط به **مثال ۱۵** الگوریتم )



**مثال ۳)** فلوچارتی رسم نمائید که یک عدد را گرفته و مقسوم علیه های آن را چاپ کند .  
( مربوط به **مثال ۶** الگوریتم )

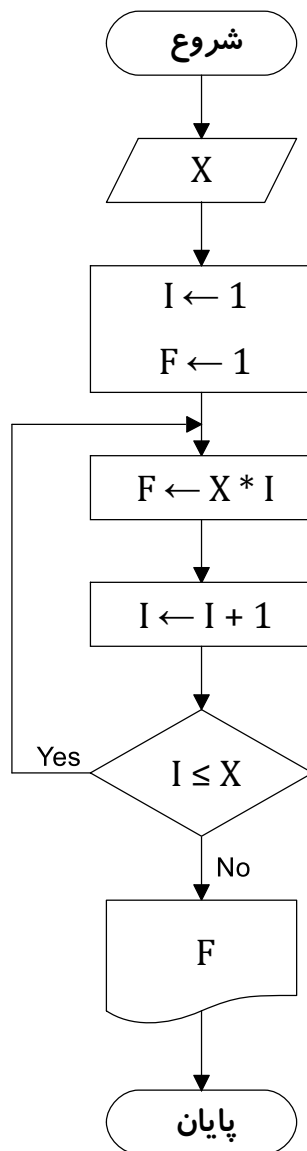


**مثال ۴)** فلوچارتی رسم نمائید که یک عدد را گرفته و مشخص کند که عدد کامل است یا خیر ؟  
( مربوط به مثال ۹ الگوریتم )



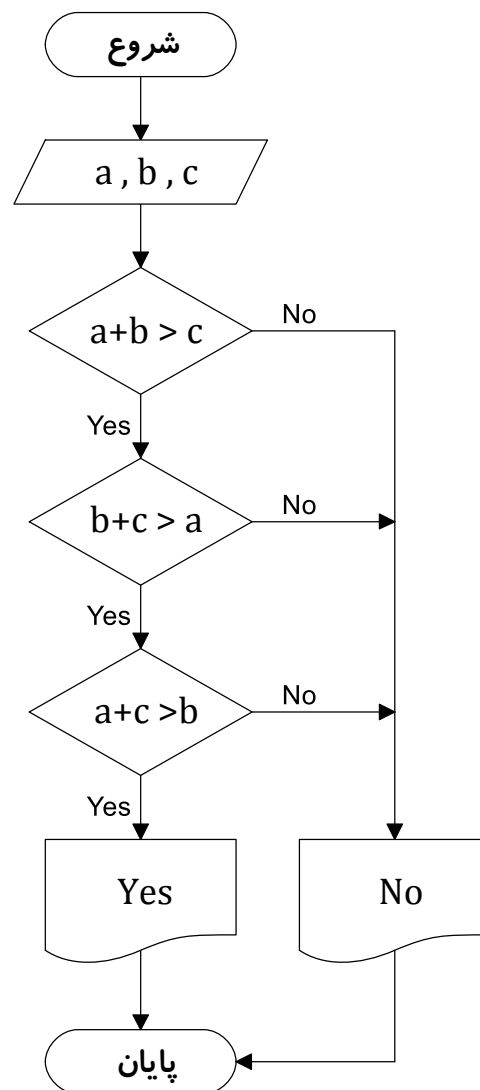
**مثال ۵)** فلوچارتی رسم نمائید که یک عدد را گرفته و فاکتوریل آن را چاپ نماید .

( مربوط به مثال ۲۲ الگوریتم )

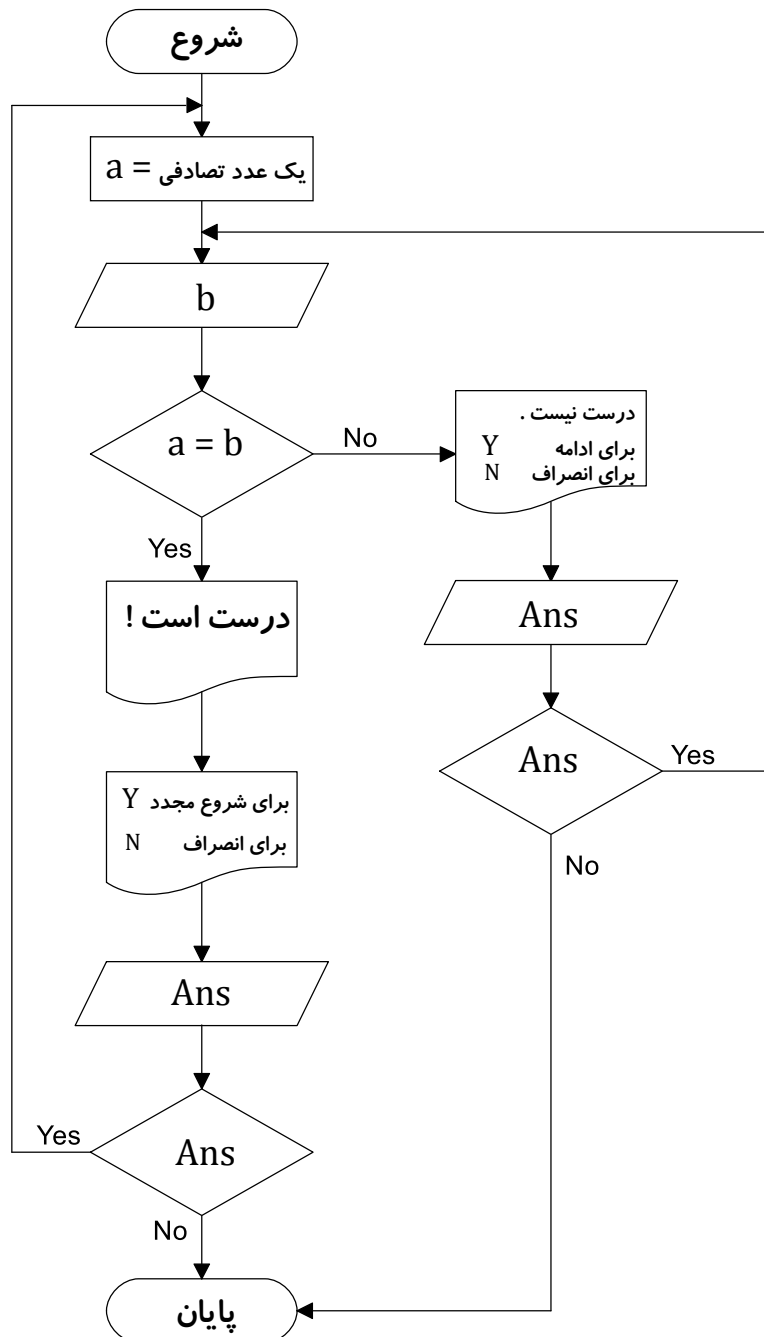


**مثال ۶)** فلوچارتی رسم نمائید که سه عدد را گرفته و مشخص کند که این سه عدد می توانند اضلاع یک مثلث باشند یا خیر ؟

راهنمایی : سه عدد زمانی یک مثلث تشکیل می دهند که مجموع دو ضلع از ضلع سوم بزرگتر باشد .



**مثال ۷)** فلوچارتی رسم نمائید که یک بازی را شبیه سازی نماید . ابتدا کاربر یک عدد را در نظر می گیرد ، سپس از شخص دیگری می خواهد که آن عدد را حدس بزند . آن شخص می تواند در هرجایی که خواست به بازی خاتمه دهد یا بازی را از نو شروع کند .



**مثال ۸)** فلوچارتی رسم نمائید که مقدار سینوس  $x$  را با استفاده از روش تیلور محاسبه نماید .  
الگوریتم باید مقدار  $x$  و تعداد جملات محاسباتی را از ورودی بگیرد و سینوس  $x$  را به عنوان خروجی چاپ کند .

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots -$$

۱. شروع

۲. تعداد جملات را در متغیر  $N$  ذخیره کن

۳. مقدار  $x$  را دریافت کن

۴.  $I \leftarrow 1$  و  $F \leftarrow 1$  و  $P \leftarrow x$  و  $N \leftarrow N-1$

۵.  $\text{sign} \leftarrow -1$  و  $S \leftarrow x$

۶.  $F \leftarrow F \cdot (I+1) \cdot (I+2)$

۷.  $P \leftarrow P \cdot x \cdot x$

۸.  $S \leftarrow S + (\text{sign} \cdot (P/F))$

۹.  $\text{sign} \leftarrow \text{sign} \cdot (-1)$  و  $N \leftarrow N-1$  و  $I \leftarrow I+2$

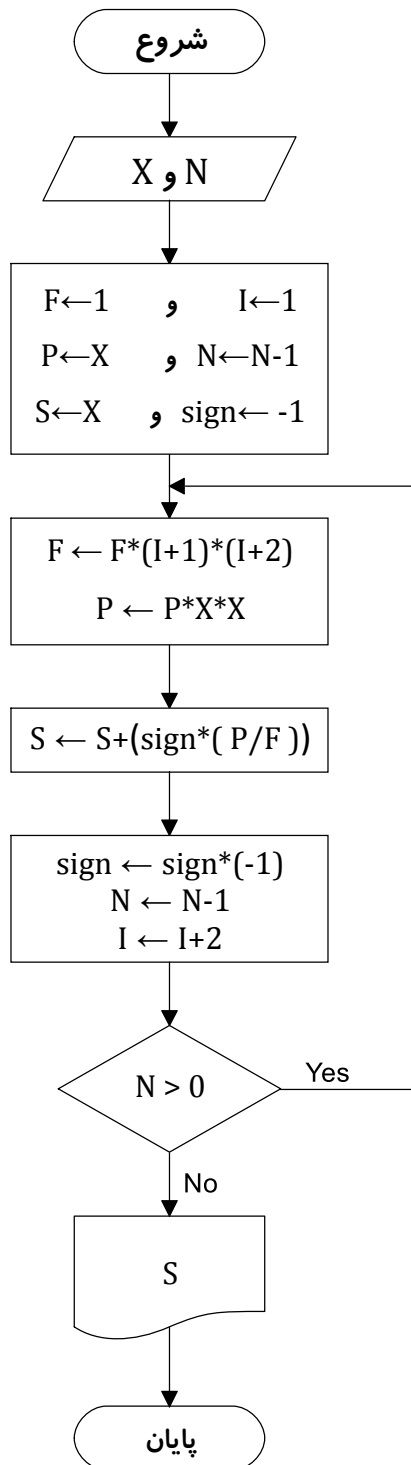
۱۰. اگر  $N > 0$  باشد برو به مرحله ۶

۱۱. چاپ کن  $S$  را

۱۲. پایان



## فلوچارت مربوط به الگوریتم صفحه قبل :



## زبان های برنامه نویسی :

برای اینکه یک کامپیوتر بتواند دستورات درخواستی یک کاربر را به درستی دریافت کرده و نتیجه را تولید نماید ، بایستی دستورات در قالب یک برنامه به کامپیوتر داده شود . زبان های برنامه نویسی برای این منظور طراحی شده اند . زبانهای برنامه نویسی از نظر نوع به سه دسته کلی تقسیم می شوند :

۱. زبانهای سطح بالا

۲. زبانهای سطح میانی

۳. زبانهای سطح پایین

هرچه به سمت زبانهای سطح بالا حرکت کنیم ، دستورات به زبان آدمی ( زبان محاوره ) نزدیک تر شده و فهم آن برای انسان آسانتر است . اما در حرکت به سمت زبانهای سطح پایین ، دستورات به صورت دنباله ای از صفر و یک هاست که درک آن برای ماشین آسان بوده ولی برای انسان دشوار است . زبانهایی مانند QBasic برای سیستم عامل Dos طراحی شده بود ، اما نرم افزارهایی که امروزه استفاده می شود ، قابلیت برنامه نویسی در سیستم عامل های گرافیکی مانند ویندوز را دارند .

## زبان برنامه نویسی ویژوال بیسیک (Visual Basic) :

ویژوال بیسیک (Visual Basic) یکی از زبانهای برنامه سازی است که در محیطی گرافیکی و تحت سیستم عامل ویندوز ، می توان با آن برنامه نویسی کرد .

## ویژگی های زبان برنامه نویسی VB :

۱. محیط برنامه نویسی گرافیکی دارد (GUI) ؛ در زبان VB ارتباط کاربر با محیط برنامه نویسی به صورت گرافیکی است که کار با محیط را آسانتر می کند ، به این ویژگی GUI گفته می شود .

۲. داشتن محیط برنامه نویسی یکپارچه (IDE) ؛ این ویژگی یعنی اینکه برنامه نویس می تواند بدون نیاز به استفاده از برنامه های کاربردی دیگر ، برنامه های تحت ویندوز خود را ایجاد ، اجرا و خطایابی نماید .

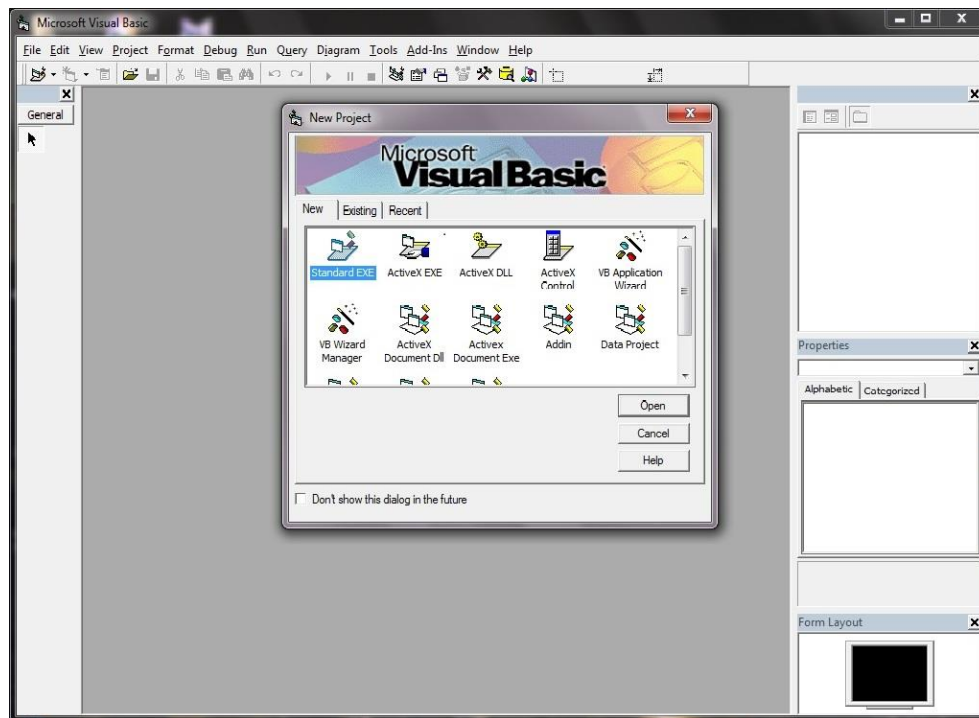
۳. ویژگی RAD ؛ بدین معنی که برای کاربردهای مختلف می توان از ابزارها (Tools) و ویزارد (Wizard) ، برنامه نویسی با سرعت زیاد انجام داد .

۴. استفاده از زبان پایه ای Basic ؛ کدهای زبان VB کاملاً منطبق بر دستورات زبان Basic می باشند .

۵. زبان برنامه نویسی شیء گرا (OOP) ؛ زبان هایی هستند که در آن برنامه نویس می تواند اشیای مختلفی را تعریف نماید و از اشیای تولید شده استفاده کند .

## آشنایی با محیط ویژوال بیسیک :

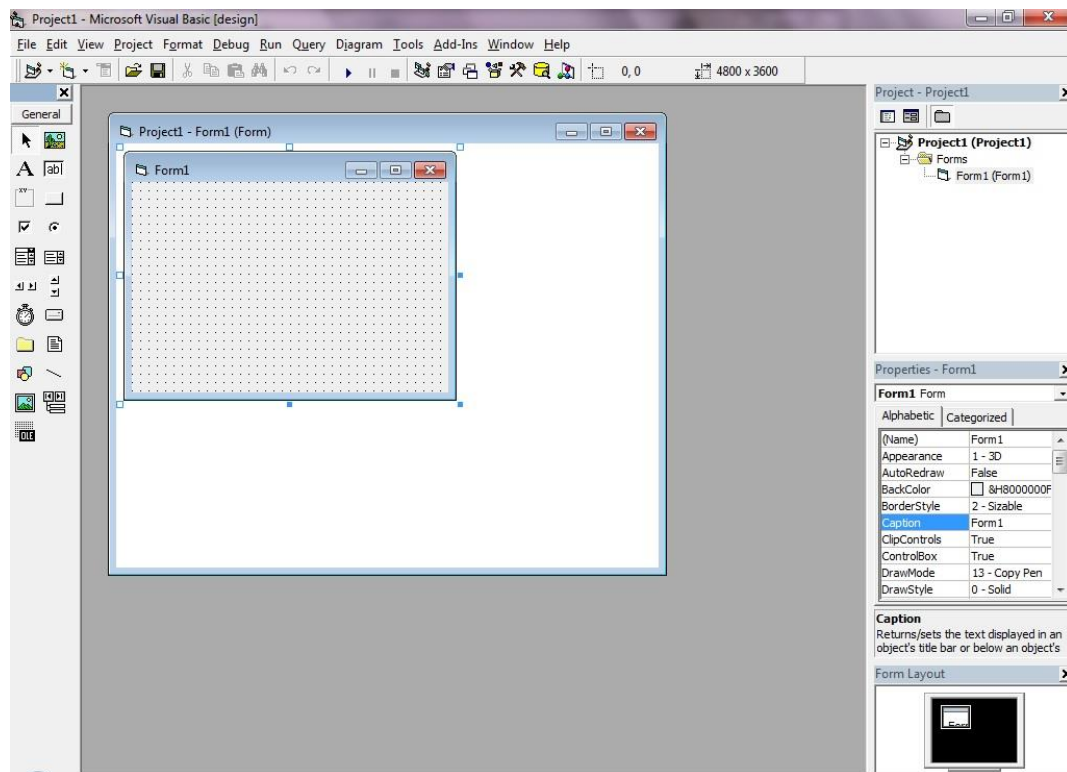
محیط ویژوال بیسیک ساده است . بعد از نصب و اجرای برنامه ویژوال بیسیک ۶ محیط ویژوال بیسیک در اولین اجرا به صورت زیر نمایش داده می شود :



کادر محاوره ای New Project به نمایش درمی آید که این کادر به برنامه نویس امکان انتخاب یکی از انواع برنامه هایی را می دهد که می توان در ویژوال بیسیک ایجاد کرد .



نوع Standard EXE که به طور پیش فرض در این کادر انتخاب شده است ، به برنامه نویس امکان می دهد که برنامه ی اجرایی استاندارد را ایجاد کند . با انتخاب نوع استاندارد برنامه ، محیط ویژوال بیسیک به شکل زیر به نمایش درمی آید :

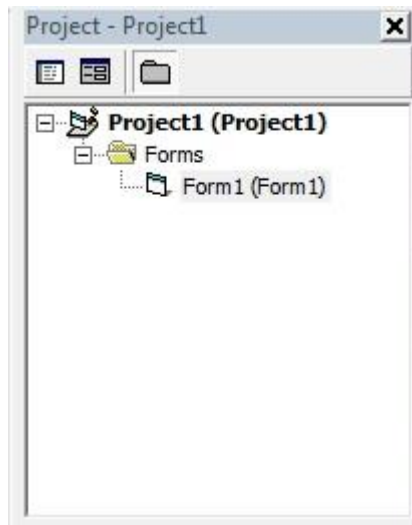


### تعریف پروژه (Project) در VB :

یک پروژه مجموعه ای از مستندات (Documents) یا مجموعه ای از فایل ها (Files) است که توسط این مجموعه می توانیم پروژه خود را سازماندهی نماییم .

### پروژه Standard EXE دارای پنجره های زیر است :

۱. پنجره پروژه (Project)
۲. پنجره سبک فرم (Form Layout)
۳. پنجره خصوصیات (Properties)
۴. جعبه ابزار (ToolBox)
۵. پنجره فرم (Form)



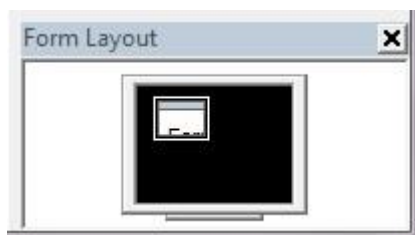
## ۱. پنجره Project :

پنجره ای است که به صورت سلسله مراتبی ، لیست پروژه ها و تمام آیتم های (عناصر) داخل پروژه را نشان می دهد . این پنجره به نام Project Explorer ( مرورگر پروژه ) نیز معروف است . در صورت عدم مشاهده این پنجره ، کلیدهای ترکیبی Ctrl+R را فشار دهید یا از منوی View گزینه Project Explorer را انتخاب کنید .

در این پنجره سه نوع دکمه متفاوت وجود دارد :

- دکمه View Code : با انتخاب این دکمه ، کد نوشته شده برای فرم نشان داده می شود .
- دکمه View Object : شکل ظاهری فرم فعال در پنجره Project را نشان می دهد .
- دکمه Toggle Folders : با هر بار فشردن این دکمه ، پوشه Forms به صورت متناوب ، گسترده شده و مخفی می شود .

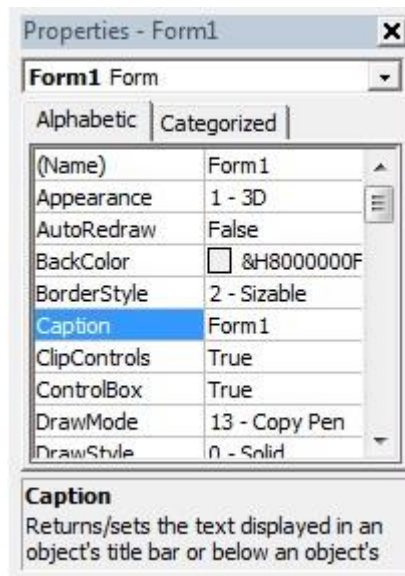
## ۲. پنجره Form Layout :



نشان دهنده ی موقعیت پنجره در زمان اجراست .

در صورت عدم مشاهده این پنجره ، از منوی View گزینه Form Layout Windows را انتخاب کنید .

### ۳. پنجره Properties :



این پنجره ، حاوی لیست خصوصیات یک فرم یا کنترل می باشد که در زمان طراحی می توان آنها را تغییر داد .

- اگر در پنجره Properties ، سربرگ Alphabetic انتخاب شده باشد ، خصوصیات براساس الفبا مرتب می شوند . ولی اگر گزینه Categorized انتخاب شود ، خصوصیات به صورت طبقه بندی شده نمایش داده می شوند .

### ۴. Toolbox (جعبه ابزار) :

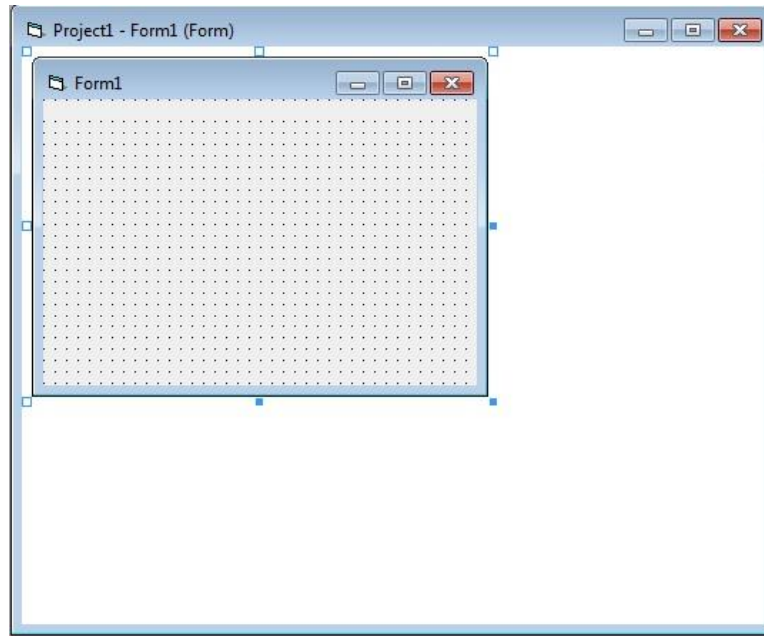


شامل کنترل هایی است که با انتخاب هر کدام و Drag کردن (کشیدن و رها کردن) آن بر روی فرم ، به داخل فرم اضافه می شود .

برای هر فرم به تعداد دلخواه می توان کنترل اضافه کرد ، به شرطی که هر کدام نام جداگانه ای داشته باشند .

## ۵. پنجره Form :

محیطی است که تمامی کارهای طراحی فرم و عملیات مربوط به هر کنترل ، در آن انجام می شود .



## انواع داده ها در ویژوال بیسیک :

انواع داده هایی که در VB مورد استفاده واقع می شوند به دو دسته کلی تقسیم می گردند :

۱. داده های عددی

۲. داده های غیر عددی



## ۱. داده های عددی :

داده های عددی در VB انواع زیر را دارند :

نوع داده	مقدار حافظه	محدوده اعداد
Byte	۱ بایت	0 تا 255
Integer	۲ بایت	-32768 تا 32767
Long	۴ بایت	تقریباً $\pm 2.1E9$
Single	۸ بایت	
Double	۸ بایت	
Currency	۸ بایت	
Decimal	۱۲ بایت	

## ۲. داده های غیر عددی :

داده های هستند که نمی توان روی آنها اعمال محاسباتی انجام داد و صرفاً برای ذخیره اطلاعات و در گزارشات و عملیات منطقی استفاده می شوند . این داده ها انواع زیر را دارند :

نوع داده	مقدار حافظه	محدوده اعداد
String (رشته)	طول رشته	از ۱ تا تقریباً ۶۵۴۰۰ نویسه
Date (تاریخ)	۸ بایت	از اول ژانویه ۱۰۰ تا ۳۱ دسامبر ۹۹۹۹
Boolean (منطقی)	۲ بایت	True یا False
Variant	طول رشته + ۲۲ بایت	مانند String
Variant	۱۶ بایت	هر عددی تا Double
Object	۴ بایت	معادل شیء تعریف شده

در ویژوال بیسیک اگر نوع داده ای تعریف نشود ، به طور پیش فرض از نوع Variant در نظر گرفته می شود . این نوع داده در موقعیت های مختلف ، نوع داده ی متفاوتی را می تواند در خود ذخیره کند . داده ای از نوع Variant را نمی توان به نوع دیگر تبدیل نمود زیرا در صورت نیاز به تبدیل ، این کار به طور اتوماتیک توسط ویژوال بیسیک انجام می شود .

## متغیر ها :

متغیر (Variable) مکانی از حافظه است که در آن می توان مقداری را ذخیره نمود .

هر متغیر دارای نام و نوع داده است .

## تعریف متغیر ها :

شکل کلی اعلان یک متغیر در VB با استفاده از کلمه کلیدی Dim به صورت زیر می باشد :

نوع داده As نام متغیر Dim

Dim A As Integer

## قواعد نامگذاری متغیر ها :

۱. نام متغیر حتماً باید با حرف شروع شود .
۲. نام متغیر حداکثر می تواند ۲۵۵ کاراکتر داشته باشد .
۳. فاصله میان حروف نام متغیر ، مجاز نیست .

## انواع متغیر های رشته ای :

در ویژوال بیسیک دو نوع متغیر رشته ای وجود دارد :

۱. رشته با طول ثابت

۲. رشته با طول متغیر

## ۱. رشته با طول ثابت :

اگر طول رشته ثابت باشد ، امکان تغییر طول وجود نداشته و زیر رشته ای با طول مشخص شده را می توان در آن قرار داد . برای تعریف رشته با طول ثابت به صورت زیر عمل می کنیم :

طول رشته \* `String As` نام متغیر `Dim`

`Dim S As String * 20`

## ۲. رشته با طول متغیر :

برای رشته های با طول متغیر ، طول رشته با شمردن تعداد کاراکترهایی که در رشته ذخیره شده است ، بدست می آید . برای تعریف رشته با طول متغیر به صورت زیر عمل می کنیم :

`Dim` نام متغیر `String As`

`Dim S As String`

• برای ذخیره متغیرهای رشته ای ، باید آنها را داخل علامت دابل کوتیشن ( " " ) قرار دهیم .

`S = "A"`

## کاراکترهای تعیین نوع متغیر :

برای تعیین نوع متغیر می توان از پسوند نوع داده نیز استفاده کرد . که این پسوندها (کاراکترها) به صورت زیر می باشند :

<code>Dim A As Integer</code>	→	<code>Dim A %</code>
<code>Dim A As Long</code>	→	<code>Dim A &amp;</code>
<code>Dim A As Single</code>	→	<code>Dim A !</code>
<code>Dim A As Double</code>	→	<code>Dim A #</code>
<code>Dim A As Currency</code>	→	<code>Dim A @</code>
<code>Dim A As String</code>	→	<code>Dim A \$</code>

- قبل از استفاده ی یک متغیر غیر Variant ، باید یکی از عبارات Static , Dim , Public , Private برای تعیین نوع متغیر مشخص شود .

- می توان چندین متغیر را یکجا تعریف نمود . مثلاً داریم :

Dim A As Integer , B As Double

Dim C, D, E As Integer

- در رشته های با طول ثابت ، اگر طول رشته از طول متغیر کمتر باشد ، فضاهای باقیمانده ، خالی مانده و هدر می شوند و اگر طول رشته ، از طول متغیر بیشتر باشد ، داده های اضافی دور ریخته می شوند .

- متغیر نوع Boolean برای نگهداری مفاهیمی مانند : 0 و 1 ، True و False ، on و off ، yes و no استفاده می شود و مقدار پیش فرض یک متغیر Boolean برابر False می باشد .

## عملگرها در ویژوال بیسیک :

عملگرها ، نمادهایی هستند که اعمالی را روی مقادیر انجام می دهند . در ویژوال بیسیک عملگرها به سه دسته تقسیم می شوند :

۱. **عملگرهای محاسباتی :** برای انجام اعمال محاسباتی بر روی عملوند ها مورد استفاده قرار می گیرند .
۲. **عملگرهای رابطه ای :** برای مقایسه دو عملوند استفاده می شوند و نتیجه ی آن یک عبارت منطقی است.
۳. **عملگرهای منطقی :** برای عملیات تصمیم گیری یا دست کاری Bit ها به کار می روند . اگر حاصل عملگر منطقی درست باشد ، مقدار True و اگر حاصل نادرست باشد ، مقدار False را برمی گرداند .

- **عملگر مقدار دهی (انتساب) :** برای ذخیره ی یک مقدار در یک متغیر از علامت انتساب ( = ) استفاده می کنیم .

در جدول های صفحه بعد انواع عملگرها را به همراه نام و مثال هایی از هر کدام مشاهده می کنید ؛

## عملگرهای محاسباتی

عملگر	نام	مثال
+	جمع	$6 + 2 = 8$
-	تفریق	$6 - 2 = 4$
*	ضرب	$6 * 2 = 12$
/	تقسیم	$11 / 2 = 5.5$
\	تقسیم صحیح	$11 \setminus 2 = 5$
MOD	باقیمانده	$11 \text{ MOD } 2 = 1$
^	توان	$2 ^ 3 = 8$

## عملگرهای رابطه ای

عملگر	نام	مثال
<	کوچک تر	$2 < 6$
>	بزرگ تر	$6 > 2$
<=	کوچکتر مساوی	$2 <= 2$
>=	بزرگتر مساوی	$6 >= 6$
<>	نامساوی	$2 <> 6$
=	مساوی	$2 = 2$

## عملگرهای منطقی

عملگر	نام	مثال
NOT	نقیض	NOT X
AND	و	X AND Y
OR	یا	X OR Y
XOR	یای انحصاری	X XOR Y
EQV	هم ارزی	X EQV Y
IMP	مشمول	X IMP Y

عملگرهای منطقی اگر  $X, Y$  دارای ارزش درستی یا نادرستی باشند ، آنگاه عملگرهای مذکور با توجه به درستی و یا نادرستی عملوندهای  $X, Y$  یا درست خواهند بود و یا نادرست .

**عملگر NOT :** این عملگر نقیض عملوندش است . (مثلا اگر  $X$  درست (T) باشد ، آنگاه  $NOT X$  نادرست (F) خواهد بود و برعکس .

**عملگر AND :** نتیجه این عملگر وقتی درست است که هر دو عملوندش درست باشد .

**عملگر OR :** نتیجه این عملگر وقتی درست است که حداقل یکی از عملوندها دارای ارزش درستی باشد .

**عملگر XOR :** نتیجه این عملگر وقتی درست است که فقط یکی از عملوندهایش درست باشد .

**عملگر EQV :** نتیجه این عملگر وقتی درست است که دو عملوند دارای ارزش یکسانی باشند .

**عملگر IMP :** نتیجه این عملگر وقتی درست است که عملگر سمت چپ درست باشد .

جدول درستی عملگرهای منطقی							
$X \text{ IMP } Y$	$X \text{ EQV } Y$	$X \text{ XOR } Y$	$X \text{ OR } Y$	$X \text{ AND } Y$	$NOT X$	$Y$	$X$
T	T	F	T	T	F	T	T
F	F	T	T	F	F	F	T
T	F	T	T	F	T	T	F
F	T	F	F	F	T	F	F

• **عملگر & :** برای ترکیب رشته ها کاربرد دارد . با این عملگر می توان رشته و عدد را با هم ترکیب نمود .

"Ali" & "reza" = "Alireza"

## اولویت عملگرها :

منظور از اولویت این است که اگر در یک عبارت محاسباتی ، چندین عملگر کنار هم وجود داشته باشند ، ترتیب پردازش عملگرها به چه صورتی خواهد بود .

### اولویت عملگرهای محاسباتی :

عملگرها	ترتیب اولویت
( )	۱
^	۲
* /	۳
\	۴
MOD	۵
+ -	۶

اگر در یک عبارت ، عملگرهایی با اولویت یکسان وجود داشته باشند ، ویژوال بیسیک محاسبات را از چپ به راست انجام می دهد . به مثال زیر توجه کنید :

$$1*5^2+7/3.5*5^3$$

در این عبارت ، تقدم ( اولویت ) عملگر توان از همه بالاتر است ، بنابراین عدد ۵ اول به توان ۲ و عدد ۵ دوم به توان ۳ می رسد :

$$1*25+7/3.5*125$$

سپس عملگرهای ضرب ، تقسیم و ضرب به ترتیب از چپ به راست اعمال می شوند و در نهایت نوبت به عملگر جمع می رسد :

$$25+2*125 = 25+250 = 275$$

**اولویت عملگرهای منطقی :**

ترتیب اولویت	عملگرها
۱	NOT
۲	AND
۳	OR
۴	XOR
۵	EQV
۶	IMP

**اولویت کلی عملگرها :**

اگر در یک عبارت ، عملگرهای ریاضی ، رابطه ای و منطقی استفاده شوند ، ابتدا عبارات ریاضی ، سپس عملگرهای رابطه ای و در نهایت عملگرهای منطقی اعمال می شوند .



## خصوصیات اشیاء :

برای هر شی‌ای که در یک فرم وجود دارد ، خصوصیات مختلفی موجود است که می‌توان خواص آن شیء را تغییر داد .

برای هر شی‌ای خصوصیات مختص آن شیء وجود دارد که به چند نمونه از آن اشاره می‌کنیم :

۱. **خاصیت Name** : نام یک شیء را نشان می‌دهد و در هنگام برنامه نویسی برای دسترسی به

یک شیء از نام آن استفاده می‌شود .

۲. **خاصیت Caption** : این خاصیت مشخص می‌کند که متن یک شیء هنگام اجرای یک

برنامه به چه صورت باشد .

خصوصیات دیگری نیز مانند Font ، BackColor ، FillColor ، BorderStyle و ... وجود دارد .

## رویدادهای مربوط به اشیاء :

برای هر شیء رویدادهای مختلفی وجود دارد . منظور از رویداد (Event) مجموعه اتفاقاتی است که می‌تواند برای یک شیء اتفاق بیفتد . از جمله رویدادهایی که برای Form قابل تصور است ، عبارتند از : Click ، DoubleClick ، Activate ، Load ، Unload و ...

## خصوصیات مربوط به TextBox :

۱. **خاصیت Appearance** : این خصوصیت نحوه‌ی نمایش یک TextBox را مشخص

می‌کند . مقادیر این خصوصیت عبارتند از : 0-Flat و 1-3D . انتخاب 0-Flat باعث می‌شود

که TextBox به صورت ساده و انتخاب 1-3D باعث می‌شود که TextBox حالت سه‌بعدی

داشته باشد . اگر داخل برنامه بخواهیم کدی بنویسیم که خصوصیت Appearance را

عوض کند ، برای حالت ساده از مقدار صفر و برای حالت سه‌بعدی از مقدار یک استفاده

می‌کنیم . به صورت زیر :

TextBox : txt1.Appearance = 0

حالت ساده

TextBox : txt1.Appearance = 1

حالت سه‌بعدی

**۲. خاصیت Text:** خاصیت Text حاوی رشته‌ای است که کاربر در آن تایپ می‌نماید. به طور پیشفرض اسم هر TextBox به عنوان مقدار اولیه‌ی خاصیت Text در نظر گرفته می‌شود.

❖ مقادیر مربوط به کنترل TextBox از نوع رشته می‌باشند. یعنی اگر محتوای چندین TextBox را با هم جمع نمائیم، بجای جمع ریاضی، جمع رشته‌ای انجام می‌دهد. یعنی مقادیر TextBox ها را به هم می‌چسباند. اگر بخواهیم از مقادیر TextBox ها به عنوان عدد استفاده نمائیم، ابتدا باید مقدار TextBox ها را توسط تابع Val (رشته) به عدد تبدیل نموده و سپس عملیات ریاضی را انجام دهیم.

**۳. خاصیت MultiLine (چندخطی بودن):** اگر بخواهیم متن‌های طولانی را در TextBox وارد کنیم و در یک سطر جا نشود، در این حالت خاصیت MultiLine آن TextBox را True می‌نمائیم.

**۴. خاصیت ScrollBars:** اگر خاصیت MultiLine یک TextBox، True باشد، تمام سطرهای TextBox نمایش داده نمی‌شود. برای جابجا شدن بین سطرهای یک TextBox خاصیت ScrollBars را در یکی از سه حالت: افقی، عمودی یا هر دو نوع می‌توان قرار داد.

## خصوصیات رایج یک TextBox:

**Alignment:** مشخص می‌کند که متنِ کادر متنی به صورت چپ چین، وسط چین و یا راست چین درون کادر متنی نمایان گردد.

**BackColor:** رنگ زمینه‌ی کادر متنی را مشخص می‌سازد.

**BorderStyle:** مشخص می‌سازد که کناره‌های تک خطی، دور کادر متنی نمایان شود.

**Enabled:** مشخص می‌کند که کادر متنی فعال باشد یا خیر؟! اغلب زمانی که دیگر کادر متنی مورد نیاز نیست، خاصیت Enabled را زمان اجرا و از درون کد تغییر خواهید داد.

**Font:** کادر مکالمه‌ای Font را نمایش می‌دهد که در آن می‌توانید نام فونت، سبک و اندازه‌ی آن را تعیین کنید.

**ForeColor :** رنگ متن TextBox را تعیین می کند .

**Height :** ارتفاع TextBox را می توان با این خاصیت تعیین کرد .

**Left :** فاصله ی لبه ی چپ TextBox را تا لبه ی چپ Form تعیین می کند .

**Locked :** مشخص می سازد که کاربر می تواند متن درون TextBox را ویرایش کند یا خیر !?

**MaxLength :** تعداد کاراکتری که کاربر می تواند درون TextBox تایپ کند را مشخص می کند.

**MousePointer :** شکل اشاره گر ماوس را در هنگامی که کاربر ماوس را بر روی TextBox حرکت می دهد ، تعیین می کند .

**PasswordChar :** اگر هنگام تایپ کاراکتر در TextBox می خواهید به جای کاراکترهای اصلی، کاراکتر دیگری تایپ شود ، آن کاراکتر را می توانید توسط این خصوصیت تعیین کنید .

**TabIndex :** نوبت TextBox را در ترتیب Focus توسط کلید Tab مشخص می سازد .

**TabStop :** مشخص می سازد که TextBox می تواند Focus را دریافت کند یا خیر !?

**ToolTipText :** متنی که به عنوان توضیح ابزار در زمان اجرا ظاهر می شود .

**Top :** فاصله ی بین لبه ی بالایی TextBox و لبه ی بالایی پنجره Form را نگه می دارد .

**Visible :** تعیین می کند که کادر متنی (TextBox) نمایان باشد یا اینکه از دید کاربر مخفی بماند .

**Width :** پهنای TextBox را نگه می دارد .

## رویدادهای مربوط به TextBox :

**رویداد KeyPress :** این خاصیت هر زمانی که کلیدی در داخل TextBox فشرده شود ، اجرا خواهد شد .

از دیگر رویدادهای مربوط به TextBox می توان : GotFocus ، LastFocus ، KeyUp ، Key Down ، Validate و ... را نام برد .

## کنترل Label (برچسب) :

از کنترل Label معمولاً برای برچسب زدن به TextBox ها و یا سایر کنترل ها در صورت نیاز استفاده می شود .

از یک Label همانند کنترل های دیگر می توان استفاده نموده و خاصیت های آن را تعیین نمود . خاصیت Alignment برای ترازبندی Label بوده و آن را راست چین ، چپ چین و وسط چین می نماید خصوصیت TabIndex ترتیب Label ها را زمانی که چندین کنترل روی فرم باشد ، مشخص می کند. منظور از ترتیب این است که زمانی که کلید Tab فشرده می شود ، کدامیک از کنترل ها انتخاب شود. TabIndex از صفر شروع شده و به تعداد کنترل ها ادامه می یابد . خاصیت Accelerator در محیط ویژوال بیسیک ، برای Focus کردن روی برخی از موضوعات ، زیر حرف اول آنها خط کشیده می شود که با گرفتن کلید Alt و زدن حرف اول آنها می توان Focus را به آن موضوع منتقل نمود . اگر بخواهیم این خاصیت را اعمال کنیم ، کافیست کاراکتر & را قبل از متن خاصیت Caption در Label قرار دهیم .

## خصوصیات رایج یک Label :

**AutoSize :** هنگامی که برابر True باشد ، چنان چه عنوان درون برچسب برای اندازه ی جاری آن بسیار بزرگ باشد ، اندازه ی برچسب به اندازه ی متن بزرگ می شود .

**BackStyle :** مشخص می کند که پس زمینه از درون برچسب قابل رؤیت باشد یا اینکه برچسب ، متن ، گرافیک و رنگ پس زمینه را بپوشاند .

**Enabled :** مشخص می کند که برچسب فعال باشد یا خیر !؟

**Font :** نوع قلم استفاده شده در برچسب را نشان می دهد .

**ForeColor :** رنگ متن برچسب را مشخص می کند .

**MousePointer :** شکل اشاره گر ماوس را در هنگامی که کاربر ماوس را بر روی برچسب حرکت می دهد ، تعیین می کند .

## تابع ( ) MsgBox :

زمانی که لازم باشد از کاربر سؤالاتی پرسیده شده و یا پیام های خطا یا تذکری به کاربر نمایش داده شود ، می توان از تابع ( ) MsgBox استفاده کرد .

همیشه سعی می کنیم تابع ( ) MsgBox را به یک متغیر صحیح انتساب دهیم تا مقدار برگشتی توسط تابع ( ) MsgBox مشخص شود . آن مقدار ، مشخص کننده دکمه ای است که کاربر کلیک نموده است . قالب تابع ( ) MsgBox به صورت زیر است :

( [Title],[Buttons],[Prompt] ) = MsgBox ( "Prompt", [Buttons], [Title] ) = یک متغیر از نوع صحیح

❖ فیلد Prompt اجباری است ، اما بقیه فیلدها اختیاری می باشند .

❖ منظور از Prompt یک رشته است که درون یک جفت علامت نقل قول ( " " ) قرار می گیرد .

❖ Buttons مقدار عددی یا عبارتی اختیاری است که نوع دکمه های مورد نیاز در کادر پیام را

نشان می دهد و دارای مقادیر زیر است :

عبارت	مقدار	شرح
VbOkOnly	0	فقط دکمه OK را نشان می دهد .
VbOkCancel	1	دکمه OK و Cancel را نشان می دهد .
VbAbortRetryIgnore	2	دکمه Abort ، Retry ، Ignore را نشان می دهد .
VbYesNoCancel	3	دکمه Yes ، No و Cancel را نشان می دهد .
VbYesNo	4	دکمه Yes و No را نشان می دهد .
VbRetryCancel	5	دکمه Retry و Cancel را نشان می دهد .

❖ فیلد Title عنوان پنجره ی پیام را نشان می دهد .

❖ مقدار بازگشتی تابع ( ) MsgBox معمولاً برای تصمیم‌گیری در برنامه استفاده می‌شود .  
یعنی اگر مثلاً OK فشار داده شد ، کار X انجام شود ولی اگر کلید Cancel فشار داده شد ،  
کار Y انجام شود . کاربر هر دکمه ای را که فشار دهد ، عددی مطابق جدول زیر در آن  
متغیر صحیح ذخیره خواهد شد :

مقدار برگشتی	کلید فشار داده شده
1	OK
2	Cancel
3	Abort
4	Retry
5	Ignore
6	Yes
7	No

به مثال زیر توجه کنید :

```
Dim a As Integer
a = MsgBox ("به ویژوال بیسیک خوش آمدید", vbOKOnly, "Welcome")
MsgBox a
```

دستور فوق ، ابتدا کادر زیر را نشان خواهد داد :



و سپس ، بعد از انتخاب دکمه OK توسط کاربر ، مقدار متغیر a (مقدار برگشتی توسط تابع) را نشان خواهد داد :



## تابع ( ) InputBox :

در مواردی که لازم است سؤالاتی از کاربر پرسیده شده و پاسخ آن دریافت شود ، از تابع ( ) InputBox استفاده می شود . تابع ( ) InputBox پیامی را نشان داده و به کاربر امکان می دهد تا اطلاعاتی را تایپ نماید . شکل استفاده از تابع ( ) InputBox به صورت زیر است :

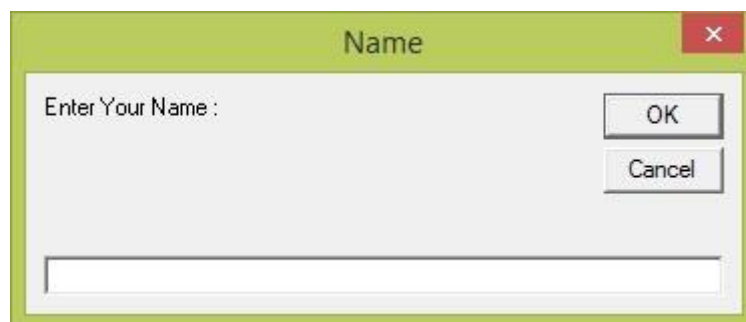
( ) InputBox ( "Prompt",[Title],[Default] ) = یک متغیر از نوع رشته

- ❖ پارامترهای Prompt و Title ، شبیه پارامترهای ( ) MsgBox هستند .
- ❖ پارامتر Default مقدار پیش فرض فیلد ورودی را مشخص می کند .
- ❖ آرگومان های [Xpos] و [Ypos] مختصات محل پنجره را نشان می دهد که اگر ذکر نشوند، پنجره در وسط صفحه خواهد بود .
- ❖ متغیر ، مقدار برگشتی تابع ( ) InputBox را در خود ذخیره می کند که نوع برگشتی از نوع رشته است .

به مثال زیر توجه کنید :

```
Dim Name As String
Name = InputBox ("Enter Your Name : ", "Name")
MsgBox "Hello " & name, vbOKOnly, "HelloBox"
```

دستور فوق ، ابتدا کادر ورودی زیر را نشان خواهد داد :



و سپس ، بعد از وارد کردن یک نام توسط کاربر (مثلاً Ramin) کادر پیام زیر را نشان خواهد داد :



## دستور شرطی IF :

بعضی مواقع لازم است یک دستور به شرطی اجرا شود که شرایط خاصی برقرار باشد . برای این منظور در زبان VB از ساختار شرطی IF استفاده می شود که نحوه ی کار آن به صورت زیر است :

دستورات **Then** شرط **If**

منظور از شرط ، یک رابطه ی مقایسه ای یا منطقی است که اگر ارزش آن درست باشد ، شرط برقرار بوده و دستور بعد از **then** اجرا می شود . ولی اگر ارزش شرط نادرست باشد ، دستور بعد از **then** اجرا نمی شود .

❖ اگر بعد از **then** فقط یک دستور داشته باشیم ، از فرم :

دستور **Then** شرط یا شرط ها **If**

استفاده می کنیم . اما اگر بعد از **then** چند دستور داشته باشیم ، از ساختار زیر استفاده می شود :

**Then** شرط یا شرط ها **If**

دستورات

**End If**

به مثال زیر توجه کنید :

```
Dim Number As Integer
Number = InputBox ("Enter Number : ")
If Number < 10 Then MsgBox "Number Is Small"
```

برنامه فوق ، یک عدد را با استفاده از تابع **InputBox()** از کاربر گرفته و اگر عدد کوچکتر از ۱۰ باشد ، پیام "Number Is Small" را نشان می دهد .



## کامل کردن IF با Else :

در نوع ساده‌ی دستور IF ، فقط پردازش حالت درست شرط انجام می‌شد ولی در نوع کامل IF ، حالت نادرست شرط هم پردازش خواهد شد . قسمت Else اجرای حالت نادرست شرط را بر عهده دارد .

**Then** شرط یا شرط ها **If**

دستور یا دستورات

**Else**

دستور یا دستورات

**End If**

به مثال زیر که تکمیل شده‌ی مثال قبلی است ، توجه کنید :

```
Dim Number As Integer
```

```
Number = InputBox ("Enter Number : ")
```

```
If Number < 10 Then
```

```
    MsgBox "Number Is Small"
```

```
Else
```

```
    MsgBox "Number Is Large"
```

```
End If
```

در برنامه فوق ، ابتدا با استفاده از تابع Inputbox ، عدد از ورودی دریافت شده و سپس با استفاده از متد Val به نوع داده‌ی عددی تبدیل می‌شود و در نهایت درون متغیر Number قرار می‌گیرد . در ادامه ، مقدار این متغیر بررسی شده و پیغام مناسب چاپ خواهد شد .

## دستور ElseIf :

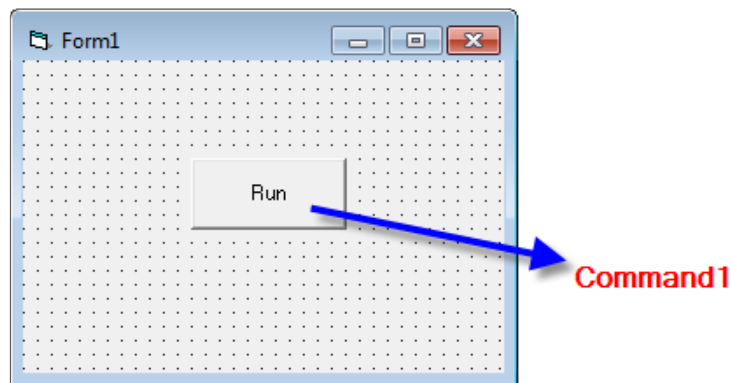
اگر یک دستور IF را بعد از Else قرار دهید ، به آنها دستورهای IF تودرتو می گویند . با استفاده از این ساختار می توان چندین بلوک از جملات را تعریف کرد که تنها یکی از آنها اجرا شود .

```
If      Then شرط اول
    جملات بلوک اول
Else If Then شرط دوم
    جملات بلوک دوم
Else If ...
    :
Else    جملات بلوک آخر
End If
```

مثال) برنامه‌ای بنویسید که نمره‌ی یک دانش آموز را از ورودی دریافت کرده و سپس بر اساس جدول زیر ، رتبه‌ی وی را نمایش دهد . (توجه داشته باشید که نمره‌ی دانش آموز باید در محدوده صفر تا بیست باشد.)

رتبه	نمره
A	بین ۱۵ تا ۲۰
B	بین ۱۰ تا ۱۵
C	بین ۵ تا ۱۰
D	کمتر از ۵

برای نوشتن این برنامه ، ابتدا پروژه جدیدی را در ویژوال بیسیک ایجاد کرده و فرمی مانند شکل زیر طراحی کنید :



سپس روی دکمه‌ی Command1 دابل کلیک کنید و کدهای زیر را وارد نمایید :

```
Private Sub Command1_Click()
Dim N As Integer
N = InputBox ("Enter Student's Number :")
If N > 20 Or N < 0 Then
    MsgBox "Number Is False"
Else
    If N < 5 Then
        MsgBox "D"
    ElseIf N > 5 And N <= 10 Then
        MsgBox "C"
    ElseIf N > 10 And N <= 15 Then
        MsgBox "B"
    ElseIf N > 15 Then
        MsgBox "A"
    End If
End If
End Sub
```

## دستور شرطی IIF :

این دستور بر اساس نتیجه‌ی آزمایش یک عبارت منطقی (شرط) ، یک مقدار عددی یا رشته‌ای را برمی گرداند . ساختار این دستور به صورت زیر است :

( دستور برای نتیجه نادرست , دستور برای نتیجه درست , شرط ) IIF

در دستور فوق ، شرط ، مورد ارزیابی قرار می گیرد . اگر ارزش آن True باشد ، دستور نتیجه درست و اگر ارزش آن False باشد ، دستور نتیجه نادرست اجرا می شود .

به عنوان مثال :

```
IIf (N > 10, Display = True, Display = False)
```

## تابع شرطی Switch :

این تابع ، یک لیست از عبارت ها را ارزیابی می کند و یک مقدار Variant یا مقدار مربوط به اولین عبارتی را که درست است ، برمی گرداند . ساختار این تابع به شکل زیر است :

( [ مقدار n , عبارت n ] , ... , [ مقدار ۲ , عبارت ۲ ] , مقدار ۱ , عبارت ۱ ) Switch

به مثال زیر توجه کنید :

```
Dim A As String
A = Switch (CityName = "Tabriz", "Azeri", CityName = "Tehran", "Persian")
```

برنامه فوق ، نام شهر را گرفته و زبانی که در آن شهر صحبت می شود را برمی گرداند . مثلاً در صورتی که مقدار متغیر CityName برابر Tabriz باشد ، متغیر a با Azeri مقداردهی می شود .

## حلقه تکرار For :

برای اجرای دستورهای مشخصی از برنامه به دفعات معین ، از حلقه تکرار For می توان استفاده کرد . این حلقه زمانی به کار می رود که تعداد دفعات تکرار مشخص باشد . در حلقه For از یک متغیر که شمارنده نامیده می شود ، استفاده می شود که هر بار با استفاده از حلقه ، افزایش و یا کاهش می یابد . شکل کلی این دستور به صورت زیر است :

[ تعداد پرش step ] مقدار پایان To مقدار شروع = متغیر شمارنده For

مجموع دستور ها

Next [متغیر]

قطعه برنامه ی زیر ، اعداد ۱ تا ۱۰ را به صورت صعودی روی فرم چاپ می کند ؛

```
For X = 1 To 10
```

```
Print X
```

```
Next X
```

اگر مقدار افزایشی شمارنده (تعداد پرش) ذکر نشود ، به صورت پیش فرض این مقدار برابر با یک خواهد بود .

قطعه برنامه ی زیر ، اعداد زوج بین ۱۰ تا ۲۰ را روی فرم چاپ می کند ؛

```
For I = 10 To 20 Step 2
```

```
Print I
```

```
Next I
```