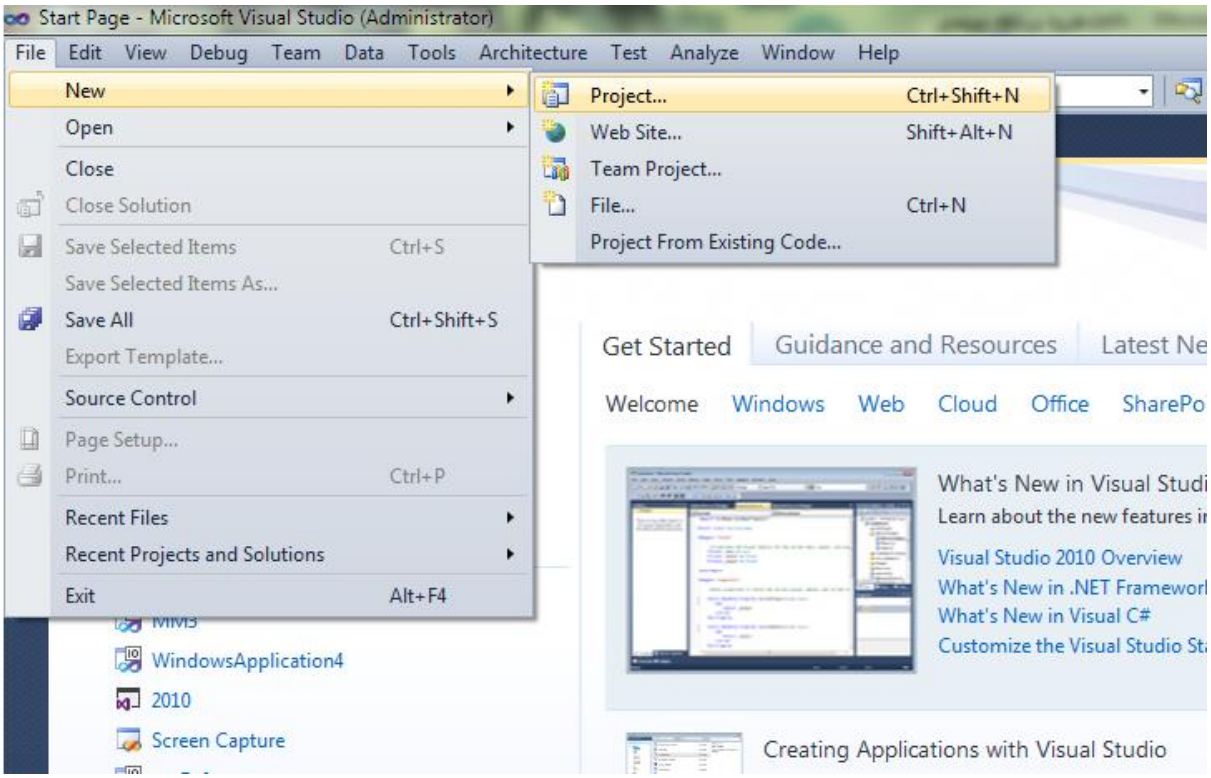
تمام دانشی که برای برنامه‌نویسی به زبان VB نیاز داریم حین اجرای یک برنامه چه رخ می‌دهد؟

نویسنده: محمد ربانی

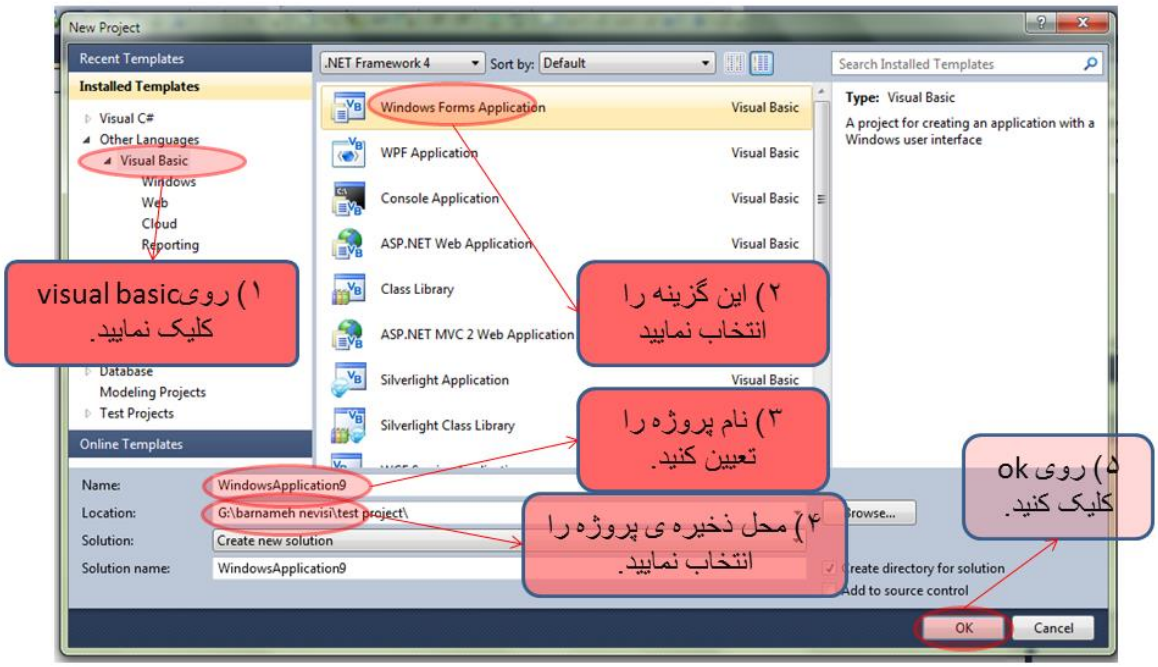
۱-۱- در زبان برنامه نویسی VB کدهای خود را کجا بنویسیم؟

ما برنامه‌ی خود را در پلت فرم Windows Form Application می‌نویسیم برای این کار مراحل زیر را طی می کنیم.

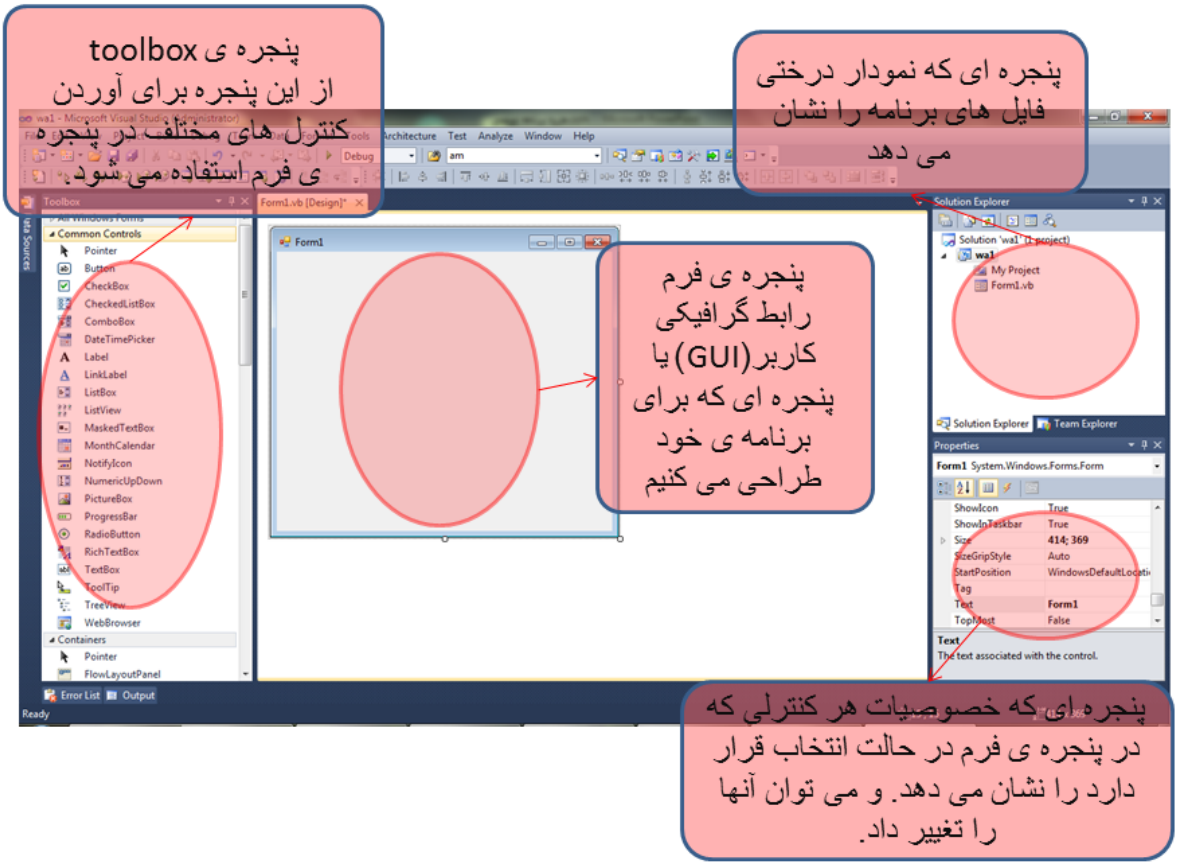
- ✓ بعد از اجرای visual studio از منوی فایل گزینه ی New و بعد گزینه ی Project را انتخاب نمایید (شکل ۱).
- ✓ حال پنجره‌ی new project باز می‌شود در این پنجره مطابق شکل ۲ عمل نمایید.
- ✓ اکنون وارد محیط Windows Form شده‌ایم. در این محیط سه پنجره‌ی مهم داریم که در شکل۳ مشخص شده‌است.



شکل -۱- انتخاب پروژه‌ی جدید

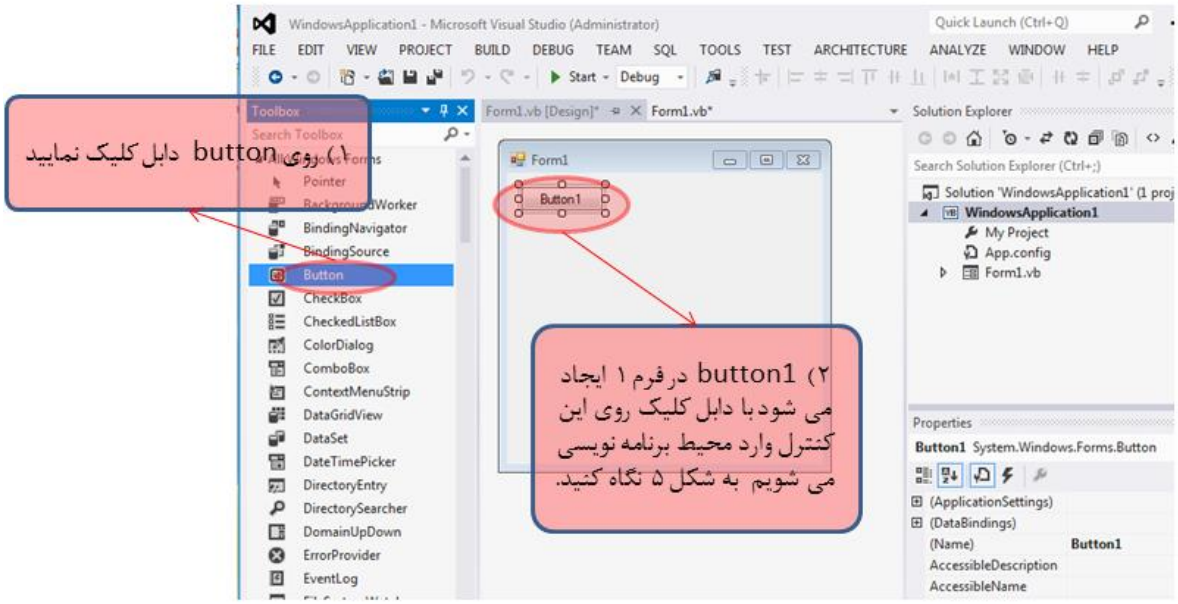


شکل -۲- انتخاب پروژه‌ی جدید



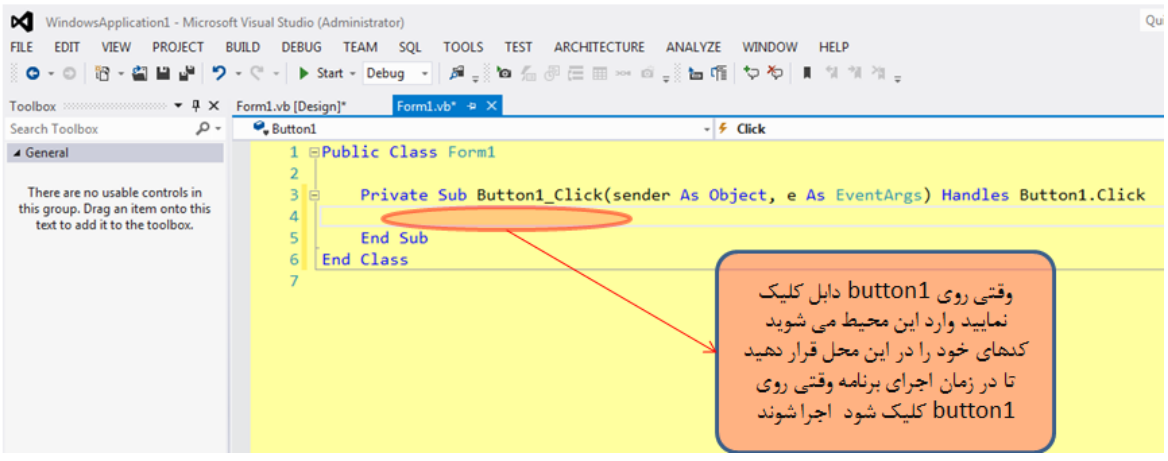
شکل -۳- محیط ویندوز فرم

حال سوال این است که : در محیط ویندوز فرم کدهای برنامه ی خود را کجا قرار دهیم؟
جواب: چون محیط ویندوز فرم یک محیط رویدادگرا است لذا ساده ترین راه این است که برنامه ی خود را در رویداد کلیک یک دکمه (button) بنویسیم. یعنی ساده ترین راه این است که برنامه ی ما با کلیک روی یک button اجرا شود برای این کار از پنجره ی toolbox روی button دابل کلیک نمایید با این کار یک دکمه به نام button1 در فرم ایجاد می شود اگر روی این دکمه (button1) دابل کلیک نمایید وارد محیط کد نویسی می شوید(شکل ۴).



شکل –۴– ایجاد button

حال وارد محیط برنامه نویسی می‌شویم مطابق شکل ۵ کدهای خود را بین خطوط private sub button1_click … و end sub بنویسید. برای اجرای برنامه کلید f5 را فشار دهید.



شکل –۵– رویداد کلیک button1

۱-۲- حال چگونه برنامه نویسی کنیم؟

اولین گام مهم این است که اعتماد به نفس داشته باشید. دوستی را می‌شناسم که ریاضیات ضعیفی دارد اما به خوبی برنامه‌نویسی می کند همچنین دوستی دارم که در ریاضیات قدرتمند است اما در برنامه نویسی ضعیف است تفاوت این دو دوست من در این است که اولی پیش خود می-گوید "من از پس برنامه نویسی بر میام کاری نداره که!" و دومی پیش خود می گوید "من نمی‌تونم برنامه نویسی کنم. یه جوریه! اصلا علاقه ندارم! " . پس به خودتان تلقیین کنید که می‌توانید برنامه نویس مجرب شوید. اطلاعات مورد نیاز برای برنامه نویسی بسیار کم است و در حقیقت کامپیوتر ۵ کار بیشتر انجام نمی‌دهد که در ادامه این کارها بررسی می‌شوند، شما باید دقیقا بدانید که با نوشتن هریک از این کارها در برنامه در زمان اجرا چه اتفاقی در کامپیوتر رخ می‌دهد سپس باید ذوق، هنر و خلاقیت سرشار خود را بکار گیرید و با ترکیب درست این ۵ کار هر مسئله‌ای را حل نمایید و از استعداد خود لذت ببرید. انتظار نداشته باشید که پاسخ تمام مسائل را همان لحظه‌ی اول بدانید بلکه روی مسائل باید فکر کنید تا بتوانید آنها را حل کنید. برخی مسائل ممکن است دیر حل شوند مثلا بعد از یک هفته فکر حل شوند اما وقتی مسئله ای حل شود مسائل مشابه نیاز به فکر چندانی ندارد. در حل مسائل به هیچ عنوان دنبال حل دیگران نروید که این یک خودفریبی است سعی کنید خودتان روی مسئله فکر کنید و آن را حل نمایید.

۱-۳- کارهایی که کامپیوتر می‌تواند انجام دهد:

a) تعریف متغییر
b) مقداردهی متغیر (محاسبه‌ی حاصل یک عبارت جبری و قرار دادن آن در یک متغیر)
c) گرفتن متغیر از ورودی و نمایش آن در خروجی
d) مقایسه بین متغیرها (دستورات شرطی)
e) ایجاد حلقه (بردن کنترل برنامه به محل دلخواهی از برنامه)

a) تعریف متغیر: منظور از تعریف متغیر در زبان برنامه نویسی این است که فضایی در ram رایانه برای یک متغیر در نظر گرفته شود. در زبان VB تعریف متغیر به صورت زیر است:

```
Dim نام متغیر as نوع متغیر
```

نام متغیر: نام متغیر را برای آن تعیین می کنیم که بعدا بتوانیم به آن اشاره کنیم که می‌تواند ترکیبی از حروف و اعداد باشد اما نباید با عدد شروع شود. مثلا نام متغیر می تواند a1 باشد یا s و یا ali. نوع متغیر: نوع متغیر را برای آن تعیین می کنیم که کامپیوتر بداند چه اندازه فضا باید برای

متغیر ما در ram در نظر بگیرد که برای اعداد صحیح به جای نوع متغیر می نویسیم integer، برای اعداد اعشاری می نویسیم double و برای متغیر از نوع حروف (کاراکتر) می نویسیم string برای مثال عبارت dim k1 as double زمانی که در برنامه اجرا شود کامپیوتر فضایی به اندازه ی double در ram خود برای متغیر اعشاری k1 در نظر می گیرد.

نکات:

۱) زمانی که متغیری را تعریف می کنیم اگر از نوع عددی باشد مثل double و integer مقدار اولیه برای آن صفر و اگر از نوع string باشد مقدار اولیه " " (یعنی تهی یا null) در نظر گرفته می‌شود.

۲) برای تعریف چند متغیر در یک خط کافی است بین آنها از کاما استفاده کنیم مثلا

```
Dim a,b,c as double
```

b) مقداردهی متغیر:

برای مقداردهی متغیر از علامت = (مساوی) استفاده می شود و علامت تساوی به غیر از یک جا که بعدا اشاره می‌شود به معنای این است که حاصل طرف دوم تساوی را در طرف اول قرار ده و این تعریف با تعریف علامت تساوی در علم ریاضی متفاوت است. به مثال زیر توجه کنیم:

```
1        Dim a,b,c as double
2        a=5
3        b=6
4        c=a+b
5        a=a+7
6        c=(c+2)*a
```

فهمیدن اینکه در اجرای یک برنامه چه اتفاقی در رایانه رخ می‌دهد بسیار پر اهمیت است و هیچ کس نمی‌تواند بدون دانستن این مطلب حتی یک مثال ساده را خود به تنهایی حل کند و متقابلا هر کس این مطلب را درک کند پیچیده ترین مسائل را حل خواهد کرد و مشکلی در برنامه‌نویسی نخواهد داشت. پس با دقت توجه کنید که موقع اجرای برنامه‌ی بالا چه رخ می‌دهد. اول اینکه برنامه خط به خط و از خط اول اجرا می‌شود. ابتدا خط اول اجرا شده و فضایی در رم کامپیوتر برای متغیر a و b و c با مقدار اولیه‌ی صفر و به اندازه‌ی double ایجاد می‌شود حال خط دوم اجرا می‌شود و مقدار ۵ در متغیر a قرار می گیرد و مقدار صفر قبلی حذف می‌شود در خط سوم مقدار۶ در متغیر b ریخته می‌شود در خط چهارم ابتدا عبارت دوم تساوی محاسبه می-

شود و حاصل در c قرار می‌گیرد یعنی محتوای a که الان ۵ است با محتوای b که الان ۶ است جمع می‌شود و حاصل جمع یعنی ۱۱ در c ریخته می‌شود در خط ۵ ابتدا محتوای a که الان ۵ است با عدد ۷ جمع می‌شود وبعد حاصل آن یعنی ۱۲ در a ریخته می‌شود و محتوای قبلی a که ۵ بود از بین می رود در خط ۶ ابتدا حاصل عبارت دوم تساوی محاسبه می‌شود یعنی ابتدا محتوای c که در اینجا ۱۱ است با عدد ۲ جمع شده که می‌شود ۱۳ حال ۱۳ در محتوای a که در اینجا ۱۲ است ضرب شده و حاصل که ۱۵۶ می‌شود در c ریخته می‌شود و محتوای قبلی c که ۱۱ بود حذف می‌شود.

درحقیقت اگر یادتان باشد در ریاضیات دوران راهنمایی سوالی داشتیم به این صورت که مثلا "حاصل عبارت جبری $3X^2-5Y+4$ به ازای X=1 و Y=4 را حساب کنید." آنگاه ما در جواب می‌نوشتیم $3(1^2)-5(4)+4=-13$ خب الان در زبان برنامه نویسی و در یک عبارت دارای تساوی می‌توان فرض کرد که همین اتفاق برای طرف دوم (که در حقیقت یک عبارت جبری از متغیرهای تعریف شده در مسئله است) رخ می‌دهد یعنی مقادیر متغیرها که از خطوط قبل بدست آمده جایگزین متغیرها شده سپس حاصل آن عبارت جبری محاسبه شده و درنهایت در طرف اول که یک متغیر است ریخته می‌شود.

اگر یک روز کامل هم صرف فهمیدن اتفاق رخ داده شده در کامپیوتر حین اجرای این مثال بکنید باز هم ضرر نکرده‌اید پس اگر متوجه نشدید بار دیگر این مطلب را مطالعه کنید.

علامت اعمال ریاضی بین متغیرها در زبان VB

در جدول زیر عبارت ریاضی و معادل آن در VB نشان داده شده عبارت به رنگ قرمز کاربرد بیشتری در این درس دارند.

معادل ریاضی	معادل VB	معادل ریاضی	معادل VB
A+B	A+B	Sin(A)	Math.sin(A)
A-B	A-B	A^B	A^B
AxB	A*B	e^x	Math.exp(x)
A÷B	A/B	π	Math.PI
A^2	A^2	e	Math.E
$\sqrt{A}$	Math.sqrt(A)	|a|	Math.abs(a)
[A]	Int(A)	باقیمانده‌ی تقسیم A بر B	A mod B
$\log_{10} A$	Math.log10(A)	خارج قسمت تقسیم A بر B	Int(a/b)
lnA	Math.log(a)		

وقت آن رسیده که حالت کلی مقداردهی متغیر در زبان برنامه نویسی را بیان کنیم

یک عبارت جبری از متغیرهای تعریف شده = نام متغیر

و همواره به خاطر داشته باشیم که در زمان اجرای برنامه هر موقع که کامپیوتر با یک عبارت جبری مواجه شود آن عبارت جبری را به ازای مقادیری که متغیرها در آن لحظه دارند حساب می‌کند. زمانی که کامپیوتر جمله‌ی بالا را اجرا می‌کند ابتدا حاصل عبارت جبری را حساب کرده سپس آن را در متغیر طرف اول علامت مساوی قرار می‌دهد.

مثلا: برنامه‌ای که مساحت یک دایره به شعاع ۵ را محاسبه می‌کند.

```
1           Dim s, r as double
2           R = 5
3           S = Math.PI * r ^ 2
```

زمانی که برنامه‌ی بالا اجرا شود ابتدا در خط اول فضاهایی به اندازه‌ی double برای متغیرهای s و r ایجاد شده مقدار صفر در آنها قرار می‌گیرد سپس خط دوم اجرا شده و مقدار ۵ در r قرار گرفته و مقدار صفر قبلی حذف می‌شود در ادامه‌ی اجرا و در خط سوم ابتدا حاصل عبارت جبری طرف دوم تساوی حساب شده یعنی $\pi(5^2)=78.54$ حال این مقدارجدید (۷۸،۵۴) در متغیر s قرار می‌گیرد و مسلما مقدار قبلی s که صفر بود از بین می‌رود.

نکات:

۱) در مقداردهی متغیرهای نوع string باید کلماتی که می‌خواهیم در متغیر قرار دهیم را در داخل دابل کوتیشن قرار دهیم و یا به عبارت کلی در هر جای برنامه اگر بخواهیم یک عبارت غیر عددی بنویسیم باید آن را داخل دابل کوتیشن قرار دهیم (بغیر از اسامی متغیرها و کلمات کلیدی مثلا dim یک کلمه‌ی کلیدی است). مثلا:

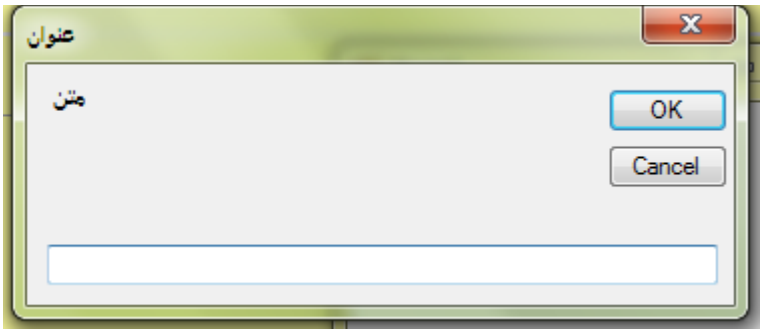
```
Dim s as string
S=”hello Ali”
```

C) گرفتن متغیر از ورودی و نمایش آن در خروجی

تا اینجا بعد از تعریف متغیر خودمان (برنامه نویس) برای یک متغیر مقدار جدید تعیین کردیم حال اگر بخواهیم در زمان اجرای یک برنامه کاربر(user) یک مقدار را برای یک متغیر تعیین کند یکی از راه‌ها در زبان VB این است که از پنجره‌ی inputbox استفاده کنیم. نحوه‌ی استفاده از آن به صورت زیر است.

نام متغیر = Inputbox("متن" , "عنوان")

زمانی که این عبارت در کامپیوتر اجرا می‌شود ابتدا پنجره‌ی inputbox (شکل ۶) باز می‌شود و عبارتی که به جای "متن" قرارگیرد در محل متن و عبارتی که به جای "عنوان" قرار گیرد در محل عنوان ظاهر می‌شود. بعد از این کامپیوتر منتظر می‌ماند تا کاربر مقداری را وارد کرده و سپس دکمه‌ی enter را فشار دهد به محض اینکه دکمه‌ی enter فشرده شود عدد وارد شده در متغیر طرف اول علامت تساوی قرار می گیرد.

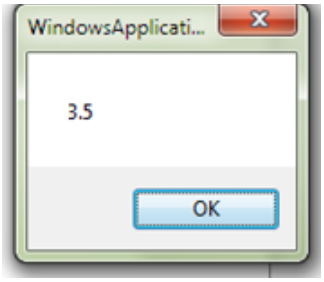


شکل ۶- پنجره‌ی inputbox

در یک برنامه تمامی تغییرات محتوایی متغیرها که در زمان اجرا رخ می‌دهد از دید کاربر مخفی است اگر بخواهیم در محلی از برنامه از مقدار یک متغیر باخبر شویم می‌توانیم آن متغیر را در خروجی نمایش دهیم برای انجام این کار یکی از راه‌ها در زبان VB استفاده از پنجره‌ی msgbox که مخفف messagebox است می‌باشد. نحوه‌ی استفاده از آن به صورت زیر است.

Msgbox(نام متغیر)

زمانی که این عبارت در کامپیوتر اجرا شود پنجره‌ی messagebox (شکل ۷) که حاوی مقدار متغیر است نمایش داده می‌شود و اجرای برنامه متوقف می‌شود تا اینکه کاربر روی دکمه‌ی ok موجود در این پنجره کلیک کند سپس ادامه‌ی برنامه اجرا می‌شود.



شکل ۷- msgbox

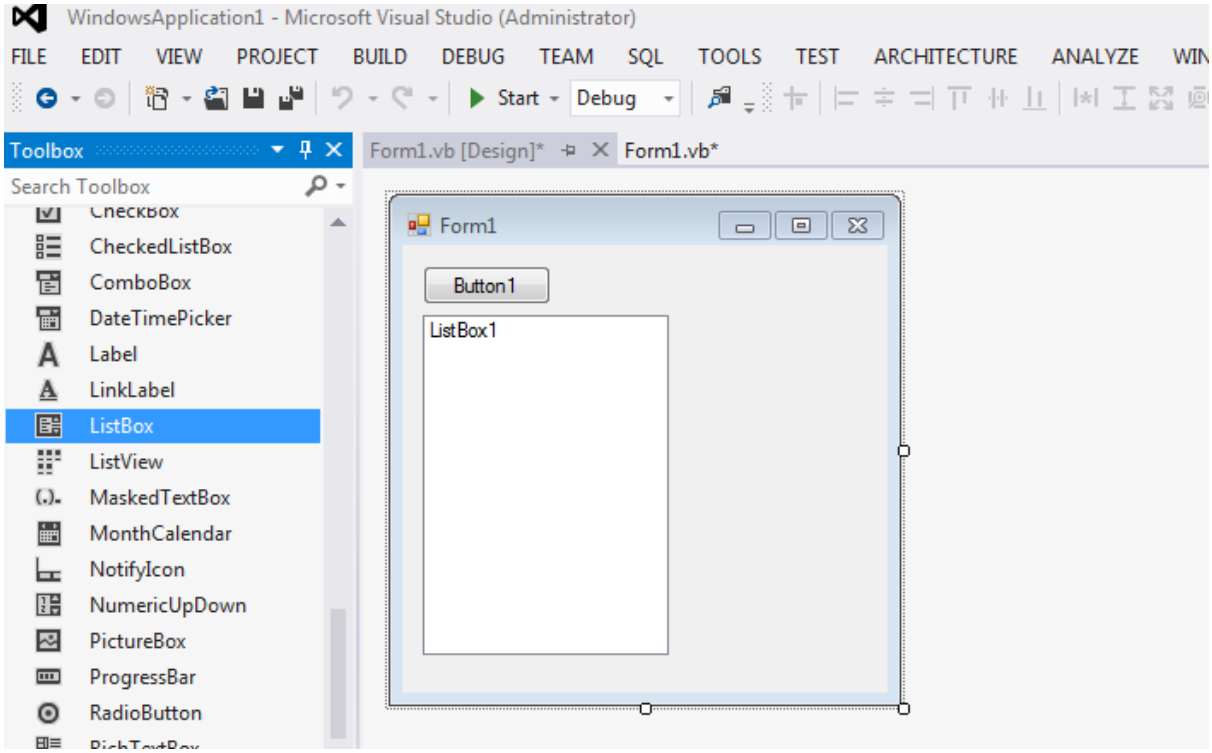
مثال: برنامه‌ای بنویسید که ضرایب a و b از معادله‌ی درجه اول aX+b=0 را دریافت کند و جواب معادله را محاسبه و چاپ نماید.

حل: حل معادله جزء کارهایی که کامپیوتر انجام می‌دهد نیست اما اگر ما خودمان معادله‌ی بالا را حل کنیم به عبارت X=-b/a می‌رسیم که X جواب معادله است. به عبارت X=-b/a دقت کنیم آیا این همان مقداردهی متغیر (که جزء کارهایی که کامپیوتر می‌تواند انجام دهد است) نیست؟ پس کاری که باید برای نوشتن برنامه‌ی این مسئله انجام دهیم این است که ابتدا سه متغیر به نام‌های a و b و X تعریف کنیم و طبق صورت مسئله a و b را از ورودی دریافت کند سپس b/a- را محاسبه کرده و در X قرار دهد و سرانجام X را نمایش دهد. پس

```
1          Dim a,b,X as double
2          A=inputbox(“”)
3          B=inputbox(“”)
4          X=-b/a
5          Msgbox(x)
```

از حل این مسئله درمی‌یابیم که کامپیوتر توانایی حل معادله را ندارد و این برنامه نویس است که روند حل کلی معادله را در برنامه پی ریزی می کند از این رو می گوییم کامپیوتر نمی‌تواند فکر کند.

یکی دیگر از راه‌های نمایش متغیر در خروجی استفاده از کنترل listbox است برای این کار ابتدا باید در محیط طراحی فرم یک کنترل listbox به فرم بیافزاییم این کار را مانند اضافه کردن button که قبلا توضیح دادیم انجام می‌دهیم (شکل ۸) کنترل listbox کنترلی است که می‌تواند ستونی از ایتم‌ها را لیست نماید. برای اینکه مثلا مقدار متغیر x در مثال قبل را به ایتم-های یک کنترل listbox اضافه کنیم می‌نویسیم listbox1.items.add(x) زمانی که این عبارت اجرا شود مقدار متغیر x به انتهای ایتم‌های موجود در لیست اضافه می‌شود.

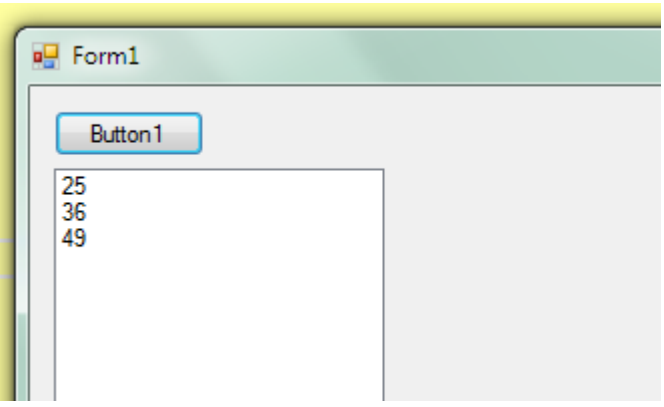


شکل –۸–کنترل لیست باکس

مثال۱: برنامه‌ای بنویسید که سه عدد را از ورودی دریافت کند و مربع آنها را در یک listbox نمایش دهد.

حل:یک button و یک listbox به فرم خود اضافه می‌کنیم و در رویداد کلیک button1 عبارات شکل ۹ را می‌نویسیم.

```
Dim a, b, c As Double
a = InputBox("")
b = InputBox("")
c = InputBox("")
a = a ^ 2
b = b ^ 2
c = c ^ 2
ListBox1.Items.Add(a)
ListBox1.Items.Add(b)
ListBox1.Items.Add(c)
```



شکل–۹– پاسخ مثال ۱ همراه با خروجی به ازای ورودی های ۵ و ۶ و ۷

اتصال متغیرها در زمان نمایش با استفاده از &

اگر بخواهیم محتوای چندین متغیر را در یک msgbox نشان دهیم مثلا بخواهیم دو متغیر x و y را در یک msgbox نشان دهیم برای این کار نمی‌توان نوشت msgbox(x,y) بلکه باید بین x و y از علامت & استفاده کرد و نوشت msgbox(x & y) اما با این کار محتوای دو متغیر بدون فاصله نمایش داده می‌شود مثلا اگر محتوای x مقدار 4 و y مقدار 7 باشد زمانی که msgbox(x & y) اجرا شود مقدار 47 نمایش داده خواهد شد حال اگر بخواهیم در موقع نمایش بین این دو متغیر فاصله و ویلگول باشد یعنی خروجی به صورت 7 , 4 باشد باید بنویسیم msgbox(x & " , " & y) در حقیقت در زمان اجرای این کد کامپیوتر محتوای x را با رشته‌ی فاصله ویلگول فاصله و آن را نیز با محتوای y اتصال داده و نمایش می‌دهد. این قاعده مخصوص نمایش در msgbox نیست و درهر نوع نمایشی مثل listbox هم قابل اجرا است.

d) مقایسه بین متغیرها (دستورات شرطی)

این قسمت را با یک مثال آغاز می کنیم.

مثال: برنامه‌ای بنویسید که x را از ورودی دریافت کند و f(x) را به صورت زیر محاسبه و چاپ نماید.

$F(x) = 3X^3$ x < 0
F(X) = Sin(x) X >= 0

حل: برای حل این مثال باید از دستور if استفاده کرد. طبق صورت مسئله ما باید ابتدا متغیرهایی به نام‌های x و fx تعریف کنیم سپس متغیر x را از ورودی دریافت کنیم و بعد اگر x<0 بود $3X^3$ را در متغیر fx قرار دهیم و اگر x>=0 بود sin(x) را در fx قرار دهیم و در آخر fx را نمایش دهیم. پس:

```
1          Dim x,fx as double
2          X=inputbox(“”)
3          If x<0 then
4            Fx=3*x^3
5          End if
6          If x>=0 then
7            Fx=math.sin(x)
8          End if
9          Msgbox(fx)
```

یک دستور if حداقل باید شامل سه کلمه‌ی کلیدی if,then,end if باشد.هر دستور if با if شروع می‌شود و با end if پایان می‌یابد در این مثال ما دوبار از دستور if استفاده کرده ایم یکی در خطوط ۳ تا ۵ و دیگری در خطوط ۶ تا ۸.

خیلی مهم ← در زمان اجرای این برنامه چه رخ می‌دهد؟ جواب: در زمان اجرا ابتدا در خط اول فضاهایی در رم کامپیوتر برای متغیرهای x و fx با مقدار اولیه‌ی صفر ایجاد می‌شود در خط دوم کامپیوتر پنجره‌ی inputbox را باز کرده و منتظر می‌ماند تا کاربر عددی را وارد کند و کلید enter را بفشارد فرض می کنیم که کاربر عدد ۴ را وارد کند و کلید enter را بزند حال عدد ۴ در متغیر X ریخته می‌شود برنامه وارد خط سه می‌شود در این خط محتوای متغیر x یعنی عدد ۴ با عدد صفر مقایسه می‌شود کامپیوتر از خود می‌پرسد آیا 4<0 درست است؟ جواب می-دهد false چون جواب false است کامپیوتر به بعد از end if، یعنی خط ۶ می‌رود در خط ۶ کامپیوتر از خود می‌پرسد آیا 4=>0 یعنی آیا ۴ بزرگتر یا مساوی ۴ است؟ جواب می‌دهد true چون جواب true است خط بعد را اجرا می کند یعنی خط ۷ در این خط کامپیوتر ابتدا sin(4) را حساب کرده سپس آن را در fx قرار می‌دهد حال به خط ۹ رفته و در اینجا مقدار fx که همان sin(4) است را نمایش می‌دهد.

توجه: برای درک صحیح نحوه‌ی اجرای برنامه‌ها، برنامه ها را درون یک باطله بنویسیم و در حین خواندن مطالب نحوه‌ی اجرا، حتما به خطوط برنامه نگاه کنیم سپس برنامه را در رایانه بنویسیم و خروجی را مشاهده کنیم تا مطالب را یاد بگیریم و حفظ نکنیم.

عبارت x<0 یک عبارت منطقی است و علامت > یک عملگرمقایسه‌ای. از توضیحات داده-شده فهمیدیم که ۱) حاصل یک عبارت منطقی یا درست (true) است یا غلط (false). ۲) کامپیوتر در مواجهه با یک عبارت منطقی مثل x<0 حاصل آن را محاسبه می کند به این ترتیب که مثلا در عبارت منطقی x<0 محتوای x در آن لحظه را با عدد صفر مقایسه می کند و درست یا غلط بودن آن مقایسه را برمی گرداند.

در ادامه در مورد عبارات منطقی بیشتر بحث می کنیم. اما الان حالات مختلف دستور if و اینکه در هر حالت در زمان اجرا در کامپیوتر چه رخ می‌دهد را بررسی می کنیم.

حالت ۱) if تک انتخابی :

```
If عبارت منطقی then
کارهایی که در صورت درست بودن عبارت منطقی باید صورت گیرد
End if
```

نحوه‌ی اجرا: ابتدا حاصل عبارت منطقی محاسبه می‌شود اگر حاصل true باشد خطوط بین then و end if اجرا می‌شود در غیر این صورت برنامه به خط بعد از end if می‌رود.

حالت ۲) if دو انتخابی:

```
If عبارت منطقی then
کارهای درست بودن
Else
کارهای غلط بودن
End if
```

نحوه‌ی اجرا: ابتدا حاصل عبارت منطقی محاسبه می‌شود اگر حاصل true باشد خطوط بین then و else (کارهای درست بودن) اجرا می‌شود و برنامه به خط بعد از end if می رود در غیر این صورت (غلط بودن عبارت منطقی) خطوط بین else و end if اجرا شده و برنامه به خط بعد از end if می‌رود .

حالت۳) if چندانتخابی :

```
1    If ۱عبارت منطقی then
2     کارهای درست بودن عبارت منطقی۱
3    Elseif ۲عبارت منطقی then
4     کارهای درست بودن عبارت منطقی۲
5    Elseif ۳عبارت منطقی then
6     کارهای درست بودن عبارت منطقی۳
7    .
8    .
9    .
```

```
10          Elseif  n عبارت منطقی then
11            n کارهای درست بودن عبارت منطقی
12          End if
```

نحوه‌ی اجرا: ابتدا حاصل عبارت منطقی ۱ محاسبه می‌شود اگر حاصل true باشد خط ۲ اجرا می‌شود و بعد برنامه به خط بعد از end if یعنی خط ۱۳ می‌رود (از if خارج می‌شود) اما اگر حاصل عبارت منطقی ۱ غلط باشد برنامه به خط ۳ می‌رود حال عبارت منطقی ۲محاسبه می‌شود اگر حاصل درست باشد خط ۴ اجرا شده و بعد برنامه به خط ۱۳ می‌رود و اگر حاصل عبارت منطقی ۲ غلط باشد خط ۶ اجرا شده و عبارت منطقی ۳ بررسی می گردد و...

عبارات منطقی:

در زبان برنامه نویسی دو نوع عبارت ریاضی استفاده می‌شود یکی عبارت جبری است که بحث شد و گفتیم که کامپیوتر در برخورد با یک عبارت جبری حاصل آن را به ازای مقادیر متغیرها در آن لحظه محاسبه می کند و دانستیم که یکی از جاهایی که ما می‌توانیم از یک عبارت جبری استفاده کنیم در مقداردهی متغیرها است. نوع دوم عبارت ریاضی که در برنامه نویسی بکار می-رود عبارت منطقی است.

صورت کلی یک عبارت منطقی به صورت زیر است.

(عبارت جبری۲) عملگر مقایسه‌ای (عبارت جبری۱) [1]

شما با عملگرهای مقایسه‌ای آشنا هستید و در حقیقت آنها را در دوران ابتدایی فرا گرفته‌اید .تمام عملگر مقایسه‌ای در زبان VB در جدول زیر آورده شده:

نام عملگرمقایسه‌ای	نماد در VB
کوچکتر است با	<
کوچکتر مساوی است با	<=
بزرگتر است با	>
بزرگتر مساوی است با	>=
مخالف است با	<>
مساوی است با	=

کامپیوتر در زمان اجرای برنامه و در مواجهه با حالت کلی یک عبارت منطقی چکار می‌کند؟
جواب: کامپیوتر حاصل آن را حساب می‌کند به این ترتیب که ابتدا حاصل عبارات جبری طرفین عبارت منطقی را حساب کرده سپس آن دو حاصل را با هم مقایسه می‌کند و حاصل این مقایسه را که یا true است و یا false برمی‌گرداند.

عملگرهای منطقی:

مهمترین عملگرهای منطقی AND و OR هستند. از این عملگرها برای ترکیب عبارات منطقی استفاده می‌شود. زمانی که دو عبارت منطقی با عملگر منطقی AND ترکیب شوند چه اتفاقی می‌افتد ؟به عبارتی زمانی که کامپیوتر با عبارت زیر مواجه شود چکار می‌کند؟

عبارت منطقی۱ AND عبارت منطقی۲

جواب: اولا اینکه ترکیب دو یا چند عبارت منطقی خود یک عبارت منطقی است یعنی حاصل آن یا درست است یا غلط. زمانی که عبارت بالا اجرا می‌شود ابتدا حاصل عبارت منطقی ۱ و حاصل عبارت منطقی ۲ حساب می‌شوند در نتیجه ۴ حالت رخ می‌دهد که جواب کلی در جدول۳ مشخص شده است.

عبارت منطقی۱	عبارت منطقی۲	عبارت منطقی۱ AND عبارت منطقی۲
true	true	**true**
true	false	**false**
false	true	**false**
false	false	**false**

جدول ۳

وبرای ترکیب دو عبارت منطقی با عملگر or جدول ۴ را داریم.

عبارت منطقی۱	عبارت منطقی۲	عبارت منطقی۱ OR عبارت منطقی۲
true	true	**true**
true	false	**true**
false	true	**true**
false	false	**false**

جدول ۴

خب سخت نگیرید مفهوم AND و OR همان مفهوم روزمره‌ی "و" و "یا" است. مثلا فرض کنید استادی به دانشجویانش بگوید علی و حسن باید مقاله بنویسند همه می‌فهمیم که هم علی و هم حسن باید مقاله بنویسند تا به گفته‌ی استاد عمل شودو اگر یکی از آنها مقاله ننویسد گفته‌ی استاد

عمل نشده و همینطور اگر استاد بگوید علی یا حسن مقاله بنویسند همه می‌فهمیم که اگر یکی از دانشجویان حسن یا علی و یا هردو مقاله بنویسند گفته‌ی استاد عمل شده و تنها اگر هیچ یک مقاله ننویسد به گفته‌ی استاد عمل نشده.

مثال: برنامه‌ای بنویسید که x را از ورودی دریافت کند و f(x) را به صورت زیر محاسبه و چاپ نماید.

$F(x) = \sqrt{2X}$ x < 0
$F(X) = x^3-4$ 0 <= x< 20
F(X) = 5 X >= 20

حل:

```
1        Dim x,y as double
2        X=inputbox(“”)
3        If x<0 then
4         Y=math.sqrt(2*x)
5        Elseif 0<=x and x<20 then
6         Y=x^3-4
7        Elseif x>5 then
8         Y=5
9        End if
         Msgbox(Y)
```

دستور if تو در تو

اگر درون یک دستور if از یک یا چند دستور if دیگر استفاده کنیم if تودرتو رخ می‌دهد

```
1  if  عبارت منطقی1  then
2      if  عبارت منطقی2  then
3        کارها
4      end if
5    کارها
6  end if
```

شکل ۱۰-یک نمونه if تودرتو

با توجه به این مطلب که هر دستور if شامل یک عبارت end if است در if تودرتو آنچه مهم است این است که باید بدانیم هر end if مربوط به کدام if است در شکل ۱۰ end if خط ۴ مربوط به if خط ۲ و end if خط ۶ مربوط به if خط ۱ است. پس شکل ۱۰ به این

صورت اجرا می‌شود در خط ۱ ابتدا عبارت منطقی۱ حساب می‌شود اگر حاصل false باشد برنامه به خط ۷ می‌رود و اگر حاصل true باشد برنامه به خط ۲ می‌رود در خط ۲ if دیگری داریم پس ابتدا حاصل عبارت منطقی ۲ بدست می‌اید اگر حاصل false باشد برنامه به خط ۵ می‌رود و اگر حاصل true باشد خط ۳ اجرا می‌شود و بعد خط ۵ اجرا می‌شود.

e) ایجاد حلقه

فرض کنید بخواهیم برنامه‌ای بنویسیم که مقادیر تابع $y=e^n$ را به ازای n=1,2,3,…, 1000 رادر یک listbox لیست کند. اگر بخواهیم بدون استفاده از حلقه‌ها این مسئله را حل کنیم باید چند هزار خط کد بنویسیم مثلا بنویسیم:

```
1          Dim n,y as double
2          Y=math.exp(1)
3          Listbox1.items.add(y)
4          Y=math.exp(2)
5          Listbox1.items.add(y)
6          Y=math.exp(3)
7          Listbox1.items.add(y)
8          Y=math.exp(4)
9          Listbox1.items.add(y)
.
.
.
2000       Y=math.exp(1000)
2001       Listbox1.items.add(y)
```

اما این راه حل منطقی نیست و اینجاست که ضرورت ایجاد حلقه خود را نشان می‌دهد حال ما این مسئله را با استفاده از حلقه‌ها حل می کنیم.

حالت۱) ایجاد حلقه با استفاده عبارت goto و حل مسئله

با استفاده از عبارت goto ما می‌توانیم کنترل برنامه را به ابتدای خط دلخواهی از برنامه ببریم برای این کار ابتدا باید یک نام برای آن خط قرار دهیم به این صورت که فرض کنید بخواهیم نام خطی را L10 قرار دهیم در ابتدای آن خط می‌نویسیم

```
L10 :
```

حال هر جای برنامه که بنویسیم

```
Goto L10
```

در زمان اجرا کنترل برنامه به ابتدای خط L10 منتقل می‌شود.

حال خوب به حل این مثال با استفاده از عبارت goto و نحوه‌ی اجرای آن دقت کنید.

حل:

```
1           Dim n,y as double
2           L10:
3           N=n+1
4           y=math.exp(n)
5           listbox1.items.add(Y)
6           if n<1000 then
7            goto L10
8           end if
```

خیلی مهم --> نحوه‌ی اجرای برنامه: ابتدا در خط ۱ فضاهایی به اندازه‌ی double با مقدار اولیه‌ی صفر برای متغیرهای n و y ایجاد می‌شود در خط ۲ فقط یک نام‌گذاری داریم در خط ۳ ابتدا محتوای n که الان صفر است با عدد ۱ جمع شده و حاصل یعنی یک در n قرار می‌گیرد و محتوای قبلی n یعنی صفر از بین می‌رود در خط ۴ ابتدا e به توان محتوای n که الان یک است رسیده و حاصل در متغیر y ریخته‌می‌شود در خط ۵ مقدار محتوای y که الان e بتوان یک است در listbox لیست می‌شود در خط ۶ حاصل عبارت منطقی n<1000 محاسبه می‌شود به این صورت که محتوای n با عدد ۱۰۰۰ مقایسه می‌شود یعنی کامپیوتر از خود می‌پرسد آیا یک کوچکتر از ۱۰۰۰ است؟ چون حاصل true است خط ۷ اجرا می‌شود و کنترل برنامه به خط ۲ یا همان L10 می‌رود سپس خط ۳ دوباره اجرا می‌شود خوب دقت کنید ابتدا محتوای n که الان یک است با عدد یک جمع می‌شود وبعد حاصل که ۲ می‌شود در n قرار می‌گیرد و محتوای قبلی از بین می‌رود در خط ۴ ابتدا مقدار e^2 حساب شده و حاصل در متغیر y قرار می‌گیرد خط ۵ مقدار y را که الان e^2 است به لیست می‌افزاید در ادامه در خط ۶ عبارت 2<1000 بررسی می‌شود و چون حاصل true است خط ۷ دوباره اجرا می‌شود و کنترل برنامه به خط L10 می‌رود و بعد به خط ۳ می‌رود در این خط مقدار ۳ در n ریخته می‌شود و.... همینطور برنامه پیش می‌رود تا اینکه مقدار n به عدد ۱۰۰۰ می‌رسد حال وقتی خط ۶ اجرا می‌شود چون 1000<1000 نیست برنامه به خط بعد از end if یعنی خط ۹ می‌رود و اجرای برنامه متوقف می‌شود. تمام این مراحل در کسری از ثانیه رخ می‌دهد. حال می‌بینید که ۲۰۰۱ خط برنامه‌ی قبل در ۸ خط خلاصه شد. در این مثال دیدیم که عبارت n=n+1 هر بار که اجرا شود یک واحد به متغیر n اضافه می‌شود پس این عبارت یک شمارنده هم می‌تواند باشد که می‌تواند تعداد اجرای یک حلقه را بشمارد و بسیار کاربردی است.

حالت۲) ایجاد حلقه با استفاده از حلقه‌ی for… next برای حل مسئله

می‌دانیم که برای حل مثالمان ما نیاز به تولید اعداد یک تا ۱۰۰۰ برای قرار دادن هر یک در تابع $y=e^n$ داریم و این کار را در حل قبل با استفاده از عبارت n=n+1 انجام دادیم و دیدیم که هرباری که این عبارت (n=n+1) اجرا می‌شد یک واحد به n اضافه می‌شد اما حلقه‌ی for…next حلقه‌ای است که خودش اعداد را تولید می‌کند. حال خوب به حل و نحوه‌ی اجرا شدن این مثال با استفاده از این حلقه توجه کنیم.

حل:

```
1          Dim n,y as double
2          For n=1 to 1000
3            Y=math.exp(n)
4            Listbox1.items.add(y)
5          Next n
```

نحوه‌ی اجرای برنامه: زمانی که خط ۱ اجرا می‌شود فضاهایی به اندازه‌ی double با مقدار اولیه‌ی صفر برای متغیرهای n و y ایجاد می‌شود در خط دوم ابتدا مقدار ۱ در متغیر n قرار می‌گیرد سپس کامپیوتر بررسی می‌کند که آیا محتوای n از عدد ۱۰۰۰ عبور کرده یا نه یعنی عبارت منطقی 1000=>1 بررسی می‌شود و چون حاصل true است کامپیوتر خط بعدی را اجرا می‌کند یعنی خط ۳، در این خط مقدار e^1 محاسبه شده و حاصل در Y ریخته می‌شود در خط ۴ محتوای y یعنی همان e^1 به listbox اضافه می‌شود وقتی برنامه به خط ۵ می‌رسد ابتدا یک واحد به محتوای n اضافه می‌کند یعنی محتوای n می‌شود ۲ و محتوای قبلی از بین می‌رود سپس برنامه دوباره به خط ۲ می‌رود در اینجا دوباره کامپیوتر بررسی می‌کند که آیا محتوای n از عدد ۱۰۰۰ عبور کرده یا نه (اینجا دیگر ۱ در n قرار نمی‌گیرد یعنی عبارت n=1 تنها یکبار و آن هم در زمان ورود به حلقه اجرا می‌شود) یعنی عبارت منطقی 1000=>2 بررسی می‌شود و چون حاصل true است کامپیوتر خط بعدی را اجرا می‌کند یعنی خط ۳، در این خط مقدار e^2 محاسبه شده و حاصل در Y ریخته می‌شود در خط ۴ محتوای y یعنی همان e^2 به listbox اضافه می‌شود وقتی برنامه به خط ۵ می‌رسد ابتدا یک واحد به محتوای n اضافه می‌کند یعنی محتوای n می‌شود ۳ و محتوای قبلی از بین می‌رود سپس برنامه دوباره به خط ۲ می‌رود و… در آخرین باری که حلقه اجرا می‌شود محتوای n به عدد ۱۰۰۰ رسیده است حال در خط ۵ محتوای n می‌شود ۱۰۰۱ و برنامه به خط ۲ برمی‌گردد در خط ۲ عبارت منطقی 1000=>1001 بررسی

می گردد چون حاصل false است برنامه به خط بعد از next می‌رود یعنی خط ۷ و اجرای برنامه متوقف می‌شود.

سوال: در حل قبل دیدیم که n یا همان متغیر حلقه یک واحد یک واحد اضافه می‌شد آیا می‌توان کاری کرد که مثلا در هربار اجرای حلقه 0.2 واحد 0.2 واحد اضافه شود؟

جواب بله برای این کار باید بنویسیم :

```
For n=عدد شروع to عدد پایانی  step 0.2
  کارها
Next n
```

مثال: برنامه ای بنویسید که جملات دنباله‌ی زیر را به ترتیب در یک لیست باکس چاپ نماید.

0.9, 0.8, 0.7,…,-0.8, -0.9

حل:

```
1          Dim I as double
2          For i=0.9 to -0.9 step -0.1
3           Listbox1.items.add(i)
4          Next i
```

حالت کلی حلقه‌ی for… next:

```
Dim  متغیر حلقه as نوع متغیر
For متغیر حلقه = عبارت جبری to عبارت جبری step عدد اعشاری
 کارهای حلقه
Next متغیر حلقه
```

توضیح: همان‌طور که پیش‌تر گفته شد کامپیوتر در زمان اجرا در برخورد با یک عبارت جبری حاصل آن را به ازای محتوای متغیر های موجود در آن عبارت جبری در آن لحظه حساب می‌کند که یک عدد می‌شود پس حالت کلی بالا بدون هیچ مشکلی اجرا خواهد شد. دقت کنید که بعد از step نمی‌توان از یک عبارت منطقی استفاده کرد و تنها می‌توان یک عدد اعشاری قرار داد.

مثال: برنامه ای که دو عدد صحیح را از کاربر دریافت کند و اعداد صحیح بین این دو عدد را تولید و در یک listbox نمایش دهد.

حل:

```
1          Dim I,a,b as integer
2          A=inputbox(“”)
3          B=inputbox(“”)
```

```
4           For i=a+1 to b-1
5            Listbox1.items.add(i)
6           Next i
```

حالت ۳) ایجاد حلقه با استفاده از عبارت loop …do برای حل مسئله

اگر یک برنامه را به صورت زیر بنویسیم

```
1           Do
2            کارها
3           loop
```

در زمان اجرا وقتی خط ۱ (عبارت do) اجرا می‌شود، می‌توان فرض کرد که کامپیوترمحل do را در حافظه نگه می‌دارد در خط ۲ کارها انجام می‌شود زمانی که برنامه به خط ۳ یعنی عبارت loop برسد کامپیوتر به محل do که ذخیره شده بود یعنی خط ۱ برمی‌گردد پس دوباره به خط ۲ رفته و کارها انجام می‌شود و باز هم برنامه به عبارت loop می‌رسد و به خط ۱ برمی‌گردد و... این حلقه بینهایت بار اجرا می‌شود. برای اینکه بتوانیم تعداد اجرای حلقه را کنترل کنیم می‌توانیم از عبارات until یا while بعد از عبارت do یا loop استفاده کنیم.

حالت کلی استفاده از عبارت while

الف)

```
1           Do  while عبارت منطقی
2            کارها
3           Loop
```

ب)

```
1           Do
2            کارها
3           Loop while عبارت منطقی
```

همواره بعد از while یا until یک عبارت منطقی قرار می‌گیرد در حین اجرای برنامه زمانی که برنامه به عبارت while برسد حاصل عبارت منطقی مربوطه محاسبه می‌شود اگر حاصل false باشد برنامه به خط بعد از loop می‌رود (یعنی از حلقه خارج می‌شود) و اگر حاصل true باشد حلقه ادامه می‌یابد.

نحوه‌ی اجرای حالت الف: در خط اول ابتدا عبارت منطقی محاسبه می‌شود اگر حاصل درست باشد برنامه به خط بعدی می‌رود و خط ۲ اجرا می‌شود در خط ۳ برنامه به خط یک برمی‌گردد و دوباره حاصل عبارت منطقی محاسبه می‌شود اگر حاصل غلط شود برنامه به خط بعد از loop می‌رود وگرنه خط ۲ اجرا می‌شود و...

نحوه‌ی اجرای حالت ب: در خط اول یعنی عبارت do برنامه به خط بعدی می‌رود و خط ۲ اجرا می‌شود. در خط سوم ابتدا عبارت منطقی محاسبه می‌شود اگر حاصل درست باشد برنامه به خط یک برمی‌گردد و اگر حاصل غلط شود برنامه به خط بعد از loop می‌رود و...

حالت کلی استفاده از عبارت until که درست عکس while است

الف)

```
1        Do  until عبارت منطقی
2        کارها
3        Loop
```

ب)

```
1        Do
2        کارها
3        Loop until عبارت منطقی
```

زمانی که برنامه به عبارت until برسد حاصل عبارت منطقی مربوطه محاسبه می‌شود اگر حاصل true باشد برنامه به خط بعد از loop می‌رود (یعنی از حلقه خارج می‌شود) و اگر حاصل false باشد حلقه ادامه می‌یابد.

نحوه‌ی اجرای حالت الف: در خط اول ابتدا عبارت منطقی محاسبه می‌شود اگر حاصل غلط باشد برنامه به خط بعدی می‌رود و خط ۲ اجرا می‌شود در خط ۳ برنامه به خط یک برمی‌گردد و دوباره حاصل عبارت منطقی محاسبه می‌شود اگر حاصل درست شود برنامه به خط بعد از loop می‌رود وگرنه خط ۲ اجرا می‌شود و...

نحوه‌ی اجرای حالت ب: در خط اول یعنی عبارت do برنامه به خط بعدی می‌رود و خط ۲ اجرا می‌شود. در خط سوم ابتدا عبارت منطقی محاسبه می‌شود اگر حاصل غلط باشد برنامه به خط یک برمی‌گردد و اگر حاصل درست شود برنامه به خط بعد از loop می‌رود و...

نکته: چون until درست عکس while است به خاطر سپردن تنها یکی کافی است.

حل برنامه‌ای که مقادیر تابع $y=e^n$ را به ازای n=1,2,3,…, 1000 رادر یک listbox لیست نماید با استفاده از حلقه‌ی do…loop :

```
1        Dim n as integer, y as double
2        Do while n<1000
3         N=n+1
4         Y=math.exp(n)
5         Listbox1.items.add(Y)
6        loop
```

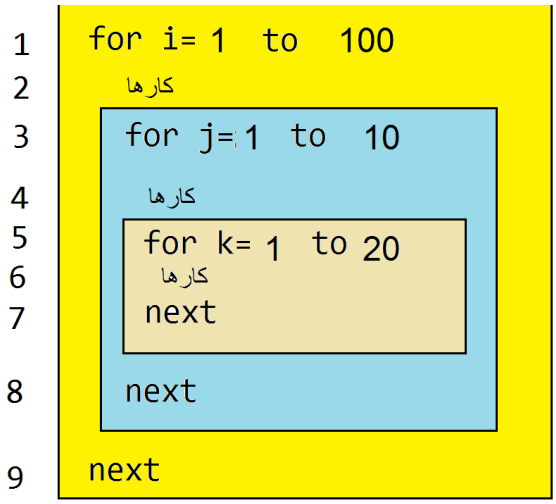
ویا

```
1        Dim n as integer, y as double
2        Do until n>999
3         N=n+1
4         Y=math.exp(n)
5         Listbox1.items.add(Y)
6        loop
```

نحوه‌ی اجرای حل اول : در خط اول فضاهایی برای متغیرهای n و y به اندازه ی integer و double با مقدار اولیه‌ی صفر در رم ایجاد می‌شود در خط دوم ابتدا عبارت منطقی 0<1000 محاسبه می‌گردد چون حاصل درست است خط سوم اجرا می‌شود در خط سوم ابتدا محتوای n با یک جمع شده و حاصل یعنی ۱ در n ریخته می‌شود در خط چهارم مقدار e^1 محاسبه شده در y ریخته می‌شود در خط پنجم محتوای y به لیست اضافه می‌شود در خط ششم برنامه به خط ۲ برمی‌گردد سپس ابتدا عبارت منطقی 1<1000 محاسبه می‌گردد چون حاصل درست است خط سوم اجرا می‌شود در خط سوم ابتدا محتوای n با یک جمع شده و حاصل یعنی 2 در n ریخته می‌شود در خط چهارم مقدار e^2 محاسبه شده در y ریخته می‌شود در خط پنجم محتوای y به لیست اضافه می‌شود در خط ششم برنامه به خط ۲ برمی‌گردد و... برنامه بار آخری که به خط ۶ می‌رسد مقدار n به عدد ۱۰۰۰ رسیده است حال برنامه به خط ۲ برمی‌گردد پس ابتدا عبارت 1000<1000 محاسبه می‌گردد چون حاصل غلط است برنامه به خط ۷ رفته و اجرای برنامه متوقف می‌شود.

حلقه‌های تودرتو

در برخی مسائل برنامه نویس مجبور است درون یک حلقه از یک یا چند حلقه‌ی دیگر استفاده کند. دانستن نحوی اجرا شدن برنامه در این موارد برای برنامه نویس بسیار اهمیت دارد درک نادرست این مطلب باعث عدم اجرای مورد انتظار برنامه خواهد شد.



شکل-۱۱- یک نمونه حلقه‌ی for تودرتو

برای نمونه نحوه‌ی اجرای شکل ۱۱ را توضیح می‌دهیم در ابتدا دقت کنیم که next خط ۹ مربوط به for خط ۱ ، next خط ۸ مربوط به for خط ۳ و next خط۷ مربوط به for خط ۵ است. در خط ۱ عدد ۱ در i ریخته‌می‌شود و چون 1<=100 درست است خط ۲ اجرا می‌شود سپس درخط ۳ ابتدا ۱ در j ریخته شده و چون 1<=10 درست است خط ۴ اجرا می‌شود در خط ۵ هم ابتدا عدد ۱ در k ریخته شده و چون 1<=20 درست است خط ۶ اجرا می‌شود در خط ۷ ابتدا یک واحد به k اضافه شده و برنامه به خط ۵ برمی‌گردد و عبارت 2<=20 بررسی می‌شود چون حاصل درست است خط ۶ اجرا می‌شود دوباره در خط ۷ مقدار k به ۳ رسیده سپس برنامه به خط ۵ می‌رود و عبارت 3<=20 بررسی می‌گردد و... بعد از اینکه ۲۰ مرتبه این حلقه اجرا شود در خط ۷ مقدار k به عدد ۲۱ می‌رسد و برنامه به خط ۵ برگشته و عبارت 21<=20 بررسی می‌شود چون حاصل false است برنامه از حلقه‌ی شامل k خارج شده و به خط ۸ می‌رود در اینجا یک واحد به j افزوده می‌شود و برنامه به خط ۳ رفته عبارت 2<=10 بررسی می‌شود چون حاصل true است خط ۴ اجرا می‌شود در خط ۵ دوباره ابتدا عدد ۱ در k ریخته شده (دقت کنیم که عبارت k=1 هربار که ما از حلقه خارج شویم و دوباره وارد شویم یکبار اجرا می‌شود یعنی عبارت k=1 یک بار در زمان ورود به حلقه اجرا می‌شود و همین‌طور عبارات j=1 و i=1

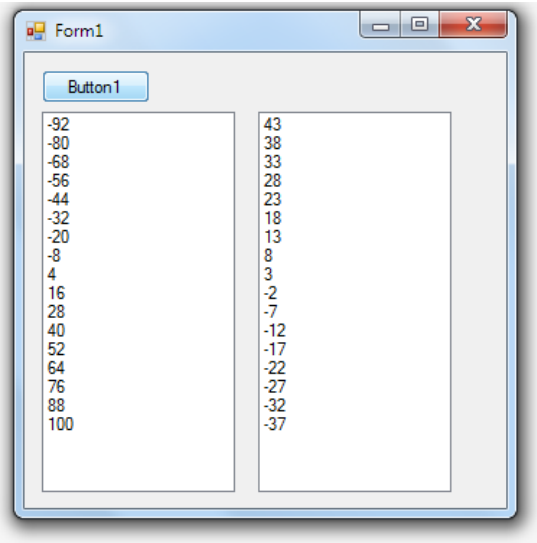
در حلقه‌های دیگر، یک بار در زمان ورود به حلقه اجرا می‌شوند) و چون 1<=20 درست است خط ۶ اجرا می‌شود در خط ۷ ابتدا یک واحد به k اضافه شده و برنامه به خط ۵ برمی‌گردد و عبارت 2<=20 بررسی می‌شودو... باز هم این حلقه ۲۰ مرتبه انجام می‌شود و بعد برنامه به خط ۸ می‌رود یک واحد به j اضافه شده برنامه به خط ۳ رفته عبارت 3<=10 بررسی می‌شود و... خلاصه اینکه حلقه‌ی j، ۱۰ مرتبه اجرا می‌شود و در تمامی این ۱۰ مرتبه حلقه‌ی k، ۲۰ مرتبه اجرا خواهد شد الان تازه برنامه به خط ۹ می‌رسد و یک واحد به متغیر i اضافه می‌شود سپس برنامه به خط ۱ برگشته و عبارت 2<=100 بررسی می‌شود چون درست است دو باره خط ۲ و بعد خط ۳ در خط ۳ دوباره حلقه‌ی j، ۱۰ مرتبه اجرا شده و در تمامی این مرتبه ها حلقه‌ی k، ۲۰ مرتبه تکرار می‌شود دو باره در خط ۹ یک واحد به i اضافه می‌شود و برنامه به خط ۱ برمی‌گرددو... در کل در این قطعه کد حلقه‌ی i، ۱۰۰ مرتبه تکرار می‌شود که در تمامی این تکرارها حلقه‌ی j، ۱۰ مرتبه که باز هم در تمامی تکرارهای حلقه‌ی j حلقه‌ی k، ۲۰ مرتبه تکرار می‌شود پس در این کد کارهای خط ۲، ۱۰۰ مرتبه، کارهای خط ۴، 100*10=1000 مرتبه و کارهای خط ۶، 100*10*20=20000 مرتبه اجرا خواهد شد. دقت کنیم که اگر این قطعه کد را به صورت شکل ۱۲ بنویسیم دیگر حلقه‌های تودرتو نداریم و هر حلقه جداگانه تکرار می‌شود کارهای خط ۲، ۱۰۰ مرتبه، کارهای خط ۵، ۱۰ مرتبه و کارهای خط ۸، ۲۰ مرتبه تکرار می‌شود.دقت کنیم که تفاوت این دو شکل (۱۱ و ۱۲) تنها در محل عبارت next است پس محل next، end if و loop بسیار اهمیت دارد.

```
1  for i= 1  to  100
2     کارها
3  next
4  for j= 1  to  10
5     کارها
6  next
7  for k= 1  to 20
8     کارها
9  next
```

شکل ۱۲- حلقه ی for متوالی

مثال: برنامه ای بنویسید که حاصل معادله دو مجهولی 5n+12m=56 که m و n عضو اعداد صحیح بوده و 101>n>-101 همچنین 51>m>-51 را نمایش دهد(مقادیر n در یک listbox و مقادیر m در listbox دیگر).

```
1          Dim n,m as integer
2          For n=-100 to 100
3          For m=-50 to 50
4           If (5*n+12*m)=56 then
5            Listbox1.items.add(n)
6            Listbox2.items.add(m)
7           End if
8          Next
9          Next
```



شکل -۱۶- خروجی برنامه

آرایه‌ها:

آرایه‌ها جزء کار اول یعنی تعریف متغیرها هستند اما چون مسائل آرایه‌ها بیشتر با حلقه‌ها همراه است در این پست مطرح می‌شود. برای تعریف یک ماتریس در برنامه نویسی از آرایه ها استفاده می‌شود.

تعریف ماتریس یک بعدی:

```
Dim نام ماتریس(تعداد درایه) as نوع متغیر
```

مثلا عبارت dim a(10) as double زمانی که اجرا شود در رم رایانه یک ردیف ده عضوی که اندازه‌ی هر عضو آن به اندازه‌ی double است ایجاد می‌شود مانند شکل زیر.

عضو۱۰	عضو ۹	...	...	....	...	...	...	عضو۲	عضو۱

مقدار اولیه‌ی تمامی اعضا صفر خواهد بود حال اگر بخواهیم مثلا مقدار 3.14 را در عضو ۹ قرار دهیم می نویسیم a(9)=3.14 .

نکته: مقدار دهی یک عضو از آرایه‌ی یک بعدی در حالت کلی به صورت زیر است:

عبارت جبری۲=(عبارت جبری۱)نام ارایه

رایانه در برخورد با این کد ابتدا حاصل عبارت جبری۱ را حساب می کند (که باید یک عدد طبیعی باشد) فرض کنیم حاصل عدد ۴ باشد سپس حاصل عبارت جبری ۲ را حساب می کند فرض کنیم حاصل 5.23 باشد در این صورت مقدار5.23 را در عضو ۴ آرایه قرار می‌دهد.

مثال: یک آرایه‌ی ۱۰۰۰ عضوی تعریف کنید و عناصر زوج آن را به ترتیب با توان های عدد ۲ پر کنیدیعنی در عنصر ۲ عدد ۲ قرار بگیرد در عنصر ۴ عدد ۴ در عنصر ۶ عدد ۸ در عنصر ۸ عدد ۱۶ و....

حل:

```
1          Dim a(1000),i as integer
2          For i=2 to 1000 step 2
3           a(i)=2^i
4          next
```

تعریف ماتریس دو بعدی:

Dim نام ماتریس (تعداد سطر ، تعداد ستون) as نوع متغیر

مثلا عبارت dim a(3,6) as double زمانی که اجرا شود در رم رایانه یک ماتریس با سه سطر و ۶ ستون که اندازه‌ی هر درایه‌ی آن به اندازه‌ی double است ایجاد می‌شود مانند شکل زیر.

عضو (1,1)	عضو (1,2)	...	...	...	عضو (1,6)
عضو (2,1)	...	...	...	...	...
عضو (3,1)	...	...	...	...	عضو (3,6)

مقدار اولیه‌ی تمامی اعضا صفر خواهد بود حال اگر بخواهیم مثلا مقدار 3.14 را در عضو (3,4)قرار دهیم می نویسیم a(3,4)=3.14 .

نکته: مقدار دهی یک عضو از آرایه‌ی دو بعدی در حالت کلی به صورت زیر است:

عبارت جبری۳=(عبارت جبری۲ ، عبارت جبری۱)نام ارایه

رایانه در برخورد با این کد ابتدا حاصل عبارت جبری۱ را حساب می‌کند (که باید یک عدد طبیعی باشد) فرض کنیم حاصل عدد ۳ باشد سپس حاصل عبارت جبری ۲ را حساب می‌کند (که باید یک عدد طبیعی باشد) فرض کنیم حاصل عدد ۴ باشد سپس حاصل عبارت جبری ۳ را حساب میکند فرض کنیم حاصل 5.23 باشد در این صورت مقدار5.23 را در عضو (3,4) آرایه قرار می‌دهد.

مثال: برنامه‌ای بنویسید که سه ماتریس 10×10 را ایجاد کند سپس اعضا ماتریس اول و دوم را از ورودی (کاربر) دریافت کرده وبعد مجموع ماتریس اول و دوم را در ماتریس سوم قرار دهد.

نکته: ماتریس حاصل جمع از مجموع نظیر به نظیر درایه های دو ماتریس بدست می‌آید.

حل:

```
1         Dim a(10,10),b(10,10),c(10,10) as double
2         Dim i,j as integer
3         For i=1 to 10
4          For j=1 to 10
5           a(i,j)=inputbox(“”)
6           b(i,j)=inputbox(“”)
7           c(i,j)=a(i,j)+b(i,j)
8          Next
9         Next
```

نحوه‌ی اجرا: در خط اول رایانه فضایی ۱۰۰عضوی برای متغیر a ، فضایی ۱۰۰ عضوی برای متغیر b و فضایی ۱۰۰ عضوی برای متغیر c که هر عضو آنها به اندازه‌ی double است با محتوای اولیه‌ی صفر ایجاد می‌کند. در خط دوم رایانه فضایی برای i وفضایی برای j به اندازه‌ی integer با محتوای اولیه‌ی صفر ایجاد می‌کند. در خط سوم ابتدا عدد ۱ جایگزین محتوای صفر متغیر i شده سپس عبارت منطقی 1<=10 حساب می‌شود چون حاصل true است خط بعدی اجرا می‌شود (اگر حاصل false بود برنامه به بعد از next خط ۹ یعنی خط ۱۰ می‌رفت) در خط۴ ابتدا عدد ۱ جایگزین محتوای صفر متغیر j شده سپس عبارت منطقی 1<=10 حساب می‌شود چون حاصل true است خط بعدی اجرا می‌شود. در خط ۵ پنجره‌ی inputbox باز شده و برنامه منتظر می‌ماند تا کاربر عددی را وارد کرده کلید enter را بزند فرض می‌شود کاربر عدد 43 را وارد کند وenter را بزند عدد 43 در عضو (1,1) متغیر a قرار می‌گیرد (عضو (1,1)چون الان محتوای iوj یک هستند). خط ۶ هم مانند خط ۵ اجرا می‌شود با این تفاوت که مقدار عدد ورودی کاربر مثلا 78 در در عضو (1,1) متغیر b قرار می-

گیرد. در خط ۷ ابتدا محتوای عضو (1,1) متغیر a با محتوای عضو (1,1) متغیر b جمع شده یعنی 43+78 و حاصل که عدد 121 است در عضو (1,1) متغیر c قرار می‌گیرد. در خط ۸ چون این next مربوط به for خط ۴ است یک واحد به متغیر j اضافه شده برنامه به خط ۴ می‌رود و عبارت منطقی 10=>2 حساب می‌شود حاصل درست است پس خط ۵ اجرا می‌شود در خط ۵ اگر کاربرعدد 27 را وارد کند و enter را بزند عدد 27 در عضو (1,2) متغیر a قرار می‌گیرد. درخط ۶ هم به فرض عدد 4 در عضو (1,2) متغیر b قرار می‌گیرد. در خط ۷ ابتدا 4+27 حساب شده و حاصل که عدد 31 است در عضو (1,2) متغیر c قرار می‌گیرد. در خط ۸ یک واحد به متغیر j اضافه شده برنامه به خط ۴ می‌رود و عبارت منطقی 10=>3 حساب می‌شود حاصل درست است پس خط ۵ اجرا می‌شود و به همین ترتیب اعضای (1,3)، (1,4)، (1,5)... و (1,10) سه متغیر a و b و c پر می‌شود تا اینکه در خط ۸ مقدار متغیر j به ۱۱ برنامه به خط ۴ می‌رود عبارت منطقی 10=>11 بررسی می‌شود حاصل false است برنامه به خط ۹ می‌رود این next مربوط به for خط ۳ پس یک واحد به i اضافه می‌شود برنامه به خط ۳ می‌رود عبارت منطقی 10=>2 بررسی می‌شود حاصل صحیح است برنامه به خط ۴ می‌رود و دوباره برنامه وارد حلقه‌ی j می‌شود پس دوباره ابتدا مقدار 1 در j قرار گرفته و حلقه ۱۰ مرتبه تکرار می‌شود این‌بار اعضای (2,1)، (2,2)، (2,3)... و (2,10) سه متغیر a و b و c پر می‌شود و برنامه به خط ۹ رفته مقدار i به ۳ می‌رسد ... برنامه به همین روال پیش میرود تا در خط ۹ مقدار i به ۱۱ می‌رسد و به خط ۳ برمی‌گردد چون دیگر 10=>11 غلط است برنامه به خط ۱۰ رفته و خاتمه می‌یابد.