

توانایی استفاده از انواع آرایه‌ها در ویژوال بیسیک

هدف‌های رفتاری

پس از مطالعه این واحد کاراز فراگیر انتظار می‌رود که:

- ۱- توانایی ایجاد انواع آرایه‌های یک بعدی، با ابعاد ثابت و متغیر را داشته باشد.
- ۲- توانایی ذخیره‌سازی و بازیابی داده‌ها به وسیله آرایه‌های یک بعدی را داشته باشد.
- ۳- توانایی ارسال آرایه‌ها به رویه‌ها را داشته باشد.
- ۴- توانایی فراخوانی یک آرایه با تعداد آرگومان‌های نامعین را داشته باشد.
- ۵- توانایی ایجاد و استفاده از آرایه‌های دوبعدی را داشته باشد.
- ۶- نحوه استفاده از توابع Ubound و LBound را بداند.
- ۷- نحوه استفاده از توابع Split، Filter و Join را بداند.
- ۸- توانایی مرتب کردن اعضای یک آرایه با روش‌های زیر را داشته باشد.
 - الف - مرتب‌سازی با روش حبابی
 - ب - مرتب‌سازی با روش انتخابی
- ۹- توانایی جستجوی اطلاعات را در آرایه‌ها با روش خطی و دودویی داشته باشد.

کلیات

تاکنون با روش‌های مختلف ذخیره‌سازی اطلاعات در حافظه اصلی کامپیوتر آشنا شده‌اید که استفاده از انواع متغیرها و خصوصیات کنترل‌ها از نمونه‌های کاملاً مشخص آن است.

گاهی در برنامه‌نویسی‌های واقعی لازم است تعداد زیادی داده را مورد پردازش قرار دهیم که به ناچار باید به تعداد مورد نظر متغیر، تعریف کرد؛ اما این روش، همواره قابل اجرا نیست، مثلاً فرض کنید می‌خواهید برنامه‌ای را طراحی کنید که باید اسامی هزار نفر از دانشجویان یک دانشگاه را دریافت کند در چنین حالتی با توجه به دانسته‌های قبلی باید هزار متغیر با اسامی مختلف تعریف کنید. آیا این روش منطقی است؟ اگر تعداد داده‌ها باز افزایش پیدا کند، چطور؟ اگر بخواهید یک اسم را در میان مجموعه اسامی پیدا کنید، چه اتفاقی می‌افتد؟ این‌گونه عملیات با روش‌های معمول یا امکان‌پذیر نیست یا از نظر تکنیکی، منطقی نخواهد بود.

برای حل این مشکل و طراحی چنین برنامه‌هایی در تمام زبان‌های برنامه‌نویسی از مفهومی به نام آرایه (Array) استفاده می‌شود. یک آرایه در واقع یک سری از چندین متغیر با یک نام مشابه است که به وسیله یک اندیس (یک عدد صحیح مثبت) از یکدیگر متمایز می‌شوند.

استفاده از آرایه‌ها باعث می‌شود تا کدهای برنامه ساده‌تر و کوتاه‌تر شود زیرا شما می‌توانید با استفاده از انواع حلقه‌ها و شماره اندیس‌ها، به هر یک از اعضای آرایه دسترسی پیدا کنید. آرایه‌ها نیز مانند متغیرهای معمولی دارای نوع داده هستند و تمام اعضای یک آرایه از یک نوع داده هستند البته می‌توانید به وسیله استفاده از نوع داده Variant انواع مختلفی از داده‌ها را در اعضای یک آرایه ذخیره کنید.

استفاده از آرایه‌ها در برنامه‌های بزرگ اجتناب‌ناپذیر است و بدون استفاده از آن‌ها انجام عملیات مرتب‌سازی و جستجوی داده کار بسیار مشکلی خواهد بود. ابعاد یک آرایه به وسیله دامنه پایینی و بالایی آن معین می‌شود و اعضای آرایه به‌طور پیوسته و پشت سر هم داخل این محدوده قرار می‌گیرند. ویژوال بیسیک برای هر یک از اعضای یک آرایه فضای جداگانه‌ای را در حافظه اختصاص می‌دهد بنابراین استفاده از آرایه‌هایی بزرگ‌تر از اندازه مورد نیاز، باعث اشغال حافظه بدون استفاده خواهد شد.

۱-۱۱ تعریف انواع آرایه در ویژوال بیسیک

در ویژوال بیسیک دو نوع آرایه وجود دارد: آرایه با ابعاد ثابت آرایه ایستا یا (Static Array) و آرایه با ابعاد متغیر (آرایه پویا یا Dynamic Array). ابتدا به نحوه تعریف آرایه‌ها با ابعاد ثابت می‌پردازیم:


۱-۱-۱ آرایه ایستا (Static Array)

برای تعریف آرایه با ابعاد ثابت می‌توانید از تمام روش‌هایی که تاکنون برای تعریف متغیرها به کار گرفته‌اید، استفاده کنید تنها تفاوتی که بین تعریف متغیر و آرایه وجود دارد تعیین ابعاد یک آرایه است. آرایه‌ها را می‌توانید به وسیله کلمات کلیدی Public، Private، Static و Dim در یک رویه یا بخش تعاریف ماژول فرم یا ماژول کد تعریف کنید.

برای تعریف یک آرایه با ابعاد ثابت می‌توانید یکی از روش‌های زیر را استفاده کنید:

نوع داده As (دامنه بالایی) نام آرایه [Dim | Public | Private | Static]

نوع داده As (دامنه بالایی To دامنه پایینی) نام آرایه [Dim | Public | Private | Static]

 **نکته** با توجه به مکان تعریف آرایه و کاربرد آن می‌توانید یکی از کلمات کلیدی موجود در [] را انتخاب کنید.

مثلاً برای تعریف یک آرایه از نوع Integer و با تعداد ۱۵ عضو از فرمان زیر استفاده می‌شود:

```
Dim no (14) As Integer
```

در ویژوال بیسیک به طور پیش فرض اولین اندیس آرایه‌ها از شماره صفر آغاز می‌شود بنابراین در مثال قبل با توجه به مقدار دامنه بالایی، اندیس‌های آرایه از صفر تا ۱۴ در نظر گرفته می‌شوند و در نتیجه تعداد اعضا ۱۵ خواهد بود.
و در تعریف یک آرایه به صورت زیر:

```
Public counters(20) As Double
```

آرایه‌ای با تعداد ۲۱ عضو و از نوع Double در حافظه آدرس‌دهی خواهد شد. روش دوم در تعریف یک آرایه با ابعاد ثابت استفاده از دامنه بالایی و پایینی است مثلاً برای تعریف آرایه‌ای (با ۱۵ عضو) که اندیس اول آن از یک شروع شود و اندیس آخرین عضو در آن ۱۵ باشد از فرمان زیر استفاده می‌شود:

```
Dim counters (1 To 15) As Double
```

و در تعریف آرایه sums که به این صورت انجام شده است:

Private sums (100 To 120) As Variant

آرایه sums دارای اولین عضو با شماره اندیس ۱۰۰ و آخرین عضو با شماره اندیس ۲۰ خواهد بود به عبارت دیگر ۲۱ عضو خواهد داشت. برای آن که با نحوه کار آرایه‌ها بهتر آشنا شوید به ذکر مثالی در این رابطه می‌پردازیم:



مثال ۱: رویه‌ای بنویسید که ده عدد را به صورت تصادفی ایجاد کرده و در آرایه‌ای ذخیره کند، سپس آرایه را نمایش دهد.

به این منظور یک رویه با نام myrandom به صورت زیر بنویسید:

Sub myrandom()

Dim i As Integer, sngno(9) As Single

Randomize

For i = 0 To 9

sngno(i) = Rnd

Next i

For i = 0 To 9

Print , "number("; i + 1; ")="; sngno(i)

Next i

End Sub

در رویه myrandom ابتدا آرایه‌ای با دامنه بالایی ۹ (۱۰ عضو) تعریف شده است سپس به وسیله یک حلقه For که مقدار شمارنده آن (i) از صفر شروع می‌شود اولین عضو آرایه یعنی sngno(0) به وسیله تابع Rnd مقداردهی می‌شود برای نمایش اعدادی که در آرایه sngno ذخیره شده‌اند نیز از یک حلقه استفاده شده است. البته شما می‌توانید به جای For از حلقه‌های دیگر نیز استفاده کنید اما این کار با استفاده از حلقه For آسان‌تر خواهد بود. در رویه myrandom تابعی به نام Rnd وجود دارد، به وسیله این تابع می‌توان اعداد تصادفی بین صفر و یک را ایجاد کرد. این تابع یک عدد از نوع Single را برمی‌گرداند و یک آرگومان اختیاری دارد که می‌تواند یک عدد از نوع Single یا یک عبارت عددی باشد. توجه داشته باشید که هر بار تابع Rnd فراخوانی می‌شود یک عدد تصادفی تولید خواهد شد البته اگر برنامه مجدداً اجرا و تابع فراخوانی شود همان اعداد

به صورت تکراری به دست می‌آیند؛ برای جلوگیری از چنین حالتی و این‌که همواره اعداد تولید شده یکسانی به دست نیایند، می‌توانید قبل از تابع Rnd از تابع Randomize استفاده کنید. تابع Randomize دارای یک آرگومان اختیاری نیز بوده که می‌تواند یک عدد یا عبارت عددی باشد.

تمرین:



رویه مثال ۱ را به‌گونه‌ای تغییر دهید تا بزرگ‌ترین عضو آرایه را به‌دست آورده و همراه اعضای آرایه نمایش دهد. سپس در یک پروژه آن را فراخوانی کنید و نتیجه را مشاهده نمایید.



مثال ۲: رویه‌ای بنویسید که دو آرایه ۵ عضوی یک بعدی از اعداد تصادفی را ایجاد کرده و حاصل ضرب آن‌ها را محاسبه نماید و در آرایه دیگری ذخیره کند.

Sub mymatrix()

```
Dim i As Integer, no1(4) As Single
```

```
Dim no2(4) As Single, no3(4) As Single
```

```
Randomize
```

```
For i = 0 To 4
```

```
no1(i) = Rnd
```

```
no2(i) = Rnd
```

```
Next i
```

```
For i = 0 To 4
```

```
no3(i) = no1(i) * no2(i)
```

```
Next i
```

End Sub

در این رویه از سه آرایه ۵ عضوی استفاده شده است و به‌وسیله اولین حلقه For اعضای دو آرایه no1 و no2 مقاداردهی شده‌اند سپس با استفاده از حلقه For دوم حاصل ضرب اعضای متناظر دو آرایه در عضو متناظر آرایه حاصل ضرب یعنی no3 قرار می‌گیرد. علاوه بر آرایه‌های عددی می‌توانید داده‌های رشته‌ای را نیز به‌صورت آرایه ذخیره کنید. در واقع فرق زیادی بین آرایه‌های رشته‌ای و عددی وجود ندارد فقط در مورد آرایه‌های رشته‌ای طول هر عضو می‌تواند ثابت یا متغیر باشد.



تمرین:

پروژه‌ای طراحی کنید که با استفاده از یک آرایه، هر عددی از مبنای ۱۰ را به مبنای ۲ تبدیل کند.



مثال ۳: پروژه‌ای طراحی کنید که اسامی ۱۰ نفر را دریافت کرده و در یک آرایه رشته‌ای ذخیره کند به علاوه بتوان اسامی ذخیره شده را روی یک فرم جداگانه مشاهده نمود. به این منظور عملیات زیر را به ترتیب انجام دهید:

۱ - برنامه ویژوال بیسیک را اجرا کرده و یک پروژه از نوع Standard EXE ایجاد کنید سپس یک فرم مطابق شکل ۱-۱۱ و جدول ۱-۱۱ ایجاد کنید و کنترل‌های آن را مطابق جدول ۲-۱۱ روی فرم قرار دهید. در این فرم از یک کنترل کادر متن برای دریافت داده‌ها و از دکمه Add برای ذخیره‌سازی اسامی در آرایه استفاده می‌شود به علاوه با دکمه Show می‌توان اسامی ذخیره شده در آرایه را در یک فرم دیگر مشاهده کرد.

جدول ۱-۱۱ خصوصیات فرم

مقدار	خصوصیت
frmarray	Name
Array	Caption



شکل ۱-۱۱

جدول ۲-۱۱ خصوصیات کنترل‌ها

کنترل / خصوصیت	Command Button	Command Button	Command Button	Text Box	Label
Name	cmdadd	cmdshow	cmdexit	txtname	lblen
Caption	Add&	Show&	Exit&	_____	:Enter Name

۲ - یک ماژول کد به پروژه اضافه کنید و در بخش تعاریف آن آرایه‌ای را با نام `strname` با ۱۰ عضو که هر عضو آن نیز توانایی دریافت ۲۰ کاراکتر را داشته باشد به صورت زیر تعریف کنید:

```
Public strname(9) As string* 20
```

رویداد Click کنترل دکمه Add را به این صورت تنظیم کنید:

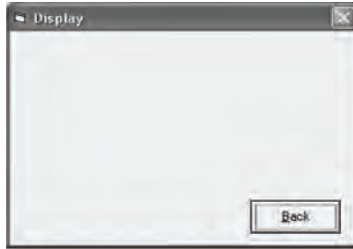
```
Private Sub cmdadd_Click()  
    Static i As Integer  
    If i < 10 Then  
        strname(i) = txtname.Text  
        i = i + 1  
    Else  
        MsgBox «Array is full»  
    End If  
End Sub
```

در این رویداد از یک متغیر i به‌عنوان شمارنده اندیس آرایه به‌صورت Static استفاده شده است تا به‌وسیله مقدار این متغیر بتوان در هر مرحله یک عضو در آرایه را با نامی که کاربر در کنترل کادر متن می‌نویسد پر کرد. اگر این متغیر به‌صورت محلی تعریف شود اسامی، همواره در اولین عضو آرایه ذخیره می‌شوند و در نتیجه، نامی که قبلاً در عضو اول آرایه ذخیره شده است از بین می‌رود. در ادامه اجرای رویداد، یک فرمان If مقدار i را کنترل می‌کند تا مقدار i از دامنه بالایی آرایه (یعنی ۹) تجاوز نکند در صورتی که مقدار i کنترل نشود در زمان رسیدن به مقدار 10 ، چون بالاترین مقدار اندیس آرایه ۹ است پیام خطای Subscript out of range نمایش داده می‌شود. بنابراین اگر $(i < 10)$ باشد آنگاه محتویات خصوصیت Text کنترل کادر متن در یکی از اعضای آرایه ذخیره می‌شود و سپس مقدار i یک واحد افزایش می‌یابد تا مرحله بعد، اندیس عضو بعدی آرایه آماده باشد، اما اگر کاربر 10 نام را وارد کند در هنگام ورود نام یازدهم با پیامی که به‌وسیله یک تابع MsgBox نمایش داده می‌شود از کامل شدن روند عملیات ورود داده مطلع می‌شود زیرا در رویداد Click دکمه فرمان Add نتیجه بررسی شرط $(i < 10)$ نادرست بوده و در نتیجه تابع MsgBox فراخوانی می‌شود.

۴ - یک فرم جدید با نام frmdisplay و عنوان Display مطابق شکل ۲-۱۱ به پروژه اضافه کرده سپس یک دکمه با نام cmdback و عنوان Back& روی آن قرار دهید. از این فرم برای نمایش اسامی ذخیره شده در آرایه استفاده می‌شود.

۵ - رویداد Click دکمه Show را در فرم frmarray به صورت زیر تنظیم کنید:

```
Private Sub cmdshow_Click()  
    frmarray.Hide  
    frmdisplay.Show  
End Sub
```



شکل ۱۱-۲

در این رویداد با استفاده از متد Hide فرم frmarray مخفی شده و با استفاده از متد Show فرم frmdisplay نمایش داده می شود.

۶ - رویداد دکمه cmdback در فرم frmdisplay به صورت زیر تنظیم کنید:

```
Private Sub cmdback_Click()  
    unload frmdisplay  
    frmarray.Show  
End Sub
```

در این رویداد با استفاده از دستور unload فرم frmdisplay بسته شده و با متد Show امکان بازگشت به فرم اول فراهم می شود.

۷ - پس از تنظیم رویداد دکمه فرمان cmdback در فرم frmdisplay دستورات زیر را در رویداد Activate همین فرم به صورت زیر تنظیم کنید:

```
Private Sub Form_Activate( )  
    Dim j  
    For j = 0 To 9  
        Print "NAME ( "; j ; " ) = "; strname(j)  
    Next j  
End Sub
```


دستورات در این رویداد باعث خواهند شد تا پس از فعال شدن فرم محتویات آرایه به وسیله یک حلقه نمایش داده شود.

۸ - پروژه و فرم frmarray را با نام Array و فرم frmdisplay را با نام Display ذخیره کنید سپس برنامه را اجرا نمایید و تعداد ۰ نام را وارد کرده و بعد از ورود هر نام، دکمه Add را کلیک کنید و بعد از پایان ورود داده‌ها به وسیله دکمه Show محتویات آرایه را مشاهده نمایید در پایان با کلیک روی دکمه Back به فرم اول بازگردید.

۹ - روی دکمه Exit کلیک کنید و به پنجره ویژوال بیسیک بازگردید.

تمرین:



برنامه قبل را به گونه‌ای تغییر دهید که در صورت عدم استفاده از تمام اعضای آرایه (خالی ماندن بعضی از اعضا) در هنگام نمایش اطلاعات در فرم cmddisplay, فقط تا آخرین اندیسی از آرایه که حاوی نام است، نمایش داده شود مثلاً اگر کاربر ۴ نام را وارد کرده است فقط همان ۴ نام نمایش داده شوند و از نمایش عناصر بعدی آرایه یعنی اندیس‌های بزرگ‌تر از ۳ خودداری شود.

در این جا لازم است که به ذکر نکته مهمی در رابطه با شماره اندیس اولین عضو در آرایه پردازیم. تاکنون وقتی یک آرایه را با ذکر مقدار اندیس بالایی آن تعریف می‌کنید شماره اندیس اولین عضو در آرایه از صفر شروع می‌شود، اما گاهی لازم است که این مقدار را با توجه به نیاز تغییر دهید، با استفاده از فرمان Option Base در بخش تعاریف می‌توانید مقدار اندیس اولین عضو را در آرایه‌ها تعیین کنید. شکل کلی فرمان به این صورت است:

Option Base n

n می‌تواند صفر یا یک باشد در صورت استفاده از مقدار صفر یا عدم استفاده از فرمان فوق اندیس آغازین در آرایه‌های برنامه صفر خواهد بود و اگر بخواهید اندیس آغازین در آرایه‌های برنامه از یک شروع شود مقدار n را در فرمان مزبور ۱ انتخاب کنید.

نکته • از این فرمان فقط یک بار و در بخش تعاریف یکی از ماژول‌ها استفاده کنید.

• برای تعیین دامنه پایینی آرایه‌ها به جای استفاده از فرمان Option Base بهتر است در تعریف

آرایه‌ها دامنه پایینی و بالایی آرایه را تعیین کنید.

به عنوان مثال فرض کنید می خواهیم یک آرایه عددی با 10° عضو را به وسیله اعداد تصادفی مقداردهی کنیم (این مسأله قبلاً حل شده است اما از آن، برای بررسی عملکرد دستور Option Base استفاده می شود).

Sub myrandom()

Dim i As Integer, sngno(10) As Single

Randomize

For i = 1 To 10

sngno(i) = Rnd

Next i

End Sub

اگر دستور Option Base 1 استفاده شود در رویه myrandom جدید و در تعریف آرایه sngno به جای عدد ۹ عدد 10° استفاده می شود و مقدار پایانی در حلقه For نیز از ۹ به 10° تغییر می یابد و مقدار شروع شمارنده نیز از صفر به ۱ تبدیل می شود. توجه داشته باشید که در ذخیره سازی مقادیر هر دو حالت، مشابه هم عمل می شود؛ فقط در نحوه تعریف آرایه ها و استفاده از حلقه باید دقت کافی داشته باشید.

تمرین:



پروژه ای طراحی کنید که ده نمره درس های یک دانش آموز را در یک آرایه ذخیره کند سپس مجموع و معدل نمرات وی را محاسبه کرده و نمایش دهد (اندیس آرایه از عدد ۱ شروع شود).

۲-۱-۱۱ آرایه پویا (Dynamic Array)

گاهی اوقات ممکن است تعداد داده ها نامشخص باشد در نتیجه نمی توان تعداد اعضا آرایه را در زمان طراحی برنامه تعیین کرد بنابراین لازم است ابعاد آرایه در زمان اجرا تنظیم شود.

به این منظور ویژوال بیسیک نوع دیگری از آرایه ها را با نام آرایه پویا در اختیار شما قرار می دهد. ابعاد آرایه ای از این نوع را می توانید در زمان اجرای برنامه با توجه به نیازتان مکرراً تغییر دهید.

برای تعریف یک آرایه دینامیک، بهتر است مانند آرایه‌های ثابت عمل کرده اما از ذکر ابعاد آرایه خودداری کنید.
به‌عنوان مثال به دستور زیر توجه کنید:

```
Dim dynnumber ( ) AS Integer
```

اما در صورت استفاده از آرایه در این مرحله پیام خطا نمایش داده خواهد شد. برای قابل استفاده شدن آرایه‌های پویا، پس از تعریف آن باید ابعاد آن را با استفاده از دستور ReDim تعیین کنید. به‌عنوان مثال پس از تعریف آرایه dynnumber از نوع Integer ابعاد آن به صورت زیر تعیین می‌شود:

```
ReDim dynnumber (10)
```

اگر Option Base ۰ باشد آرایه دارای یازده عضو و اگر Option Base ۱ باشد آرایه دارای ۱۰ عضو (از ۱ تا ۱۰) خواهد بود.



- **نکته** در صورت استفاده از دستور ReDim مقادیر موجود در تمام اعضای آرایه از بین خواهد رفت بنابراین در استفاده مجدد از دستور ReDim با دقت کافی اقدام کنید.
- در صورتی که بخواهید مقادیر موجود در آرایه در زمان تغییر ابعاد آن حفظ شوند از کلمه کلیدی Preserve همراه با دستور ReDim استفاده کنید.
- آرایه‌های پویا را نیز می‌توانید به‌صورت Public، Private یا Static تعریف کنید.

به‌عنوان مثال به دستورات زیر توجه کنید: (با فرض این که Option Base 1 است).

```
Dim myarray( ) As Integer, i As Integer
```

```
ReDim myarray(5)
```

```
Next i
```

```
ReDim myarray(10)
```

```
For i = 1 To 10
```

```
Print myarray(i)
```

```
Next i
```

با استفاده از تعریف آرایه پویا آرایه myarray با ۵ عضو تعریف شده است سپس به‌وسیله یک حلقه For مقادیر ۱ تا ۵ در آرایه قرار گرفته‌اند. پس از حلقه For اول مجدداً دستور ReDim به‌کار گرفته شده است تا تعداد اعضای آرایه دو برابر شود پس از تغییر

ابعاد آرایه، حلقه For دوم مقادیر موجود در آرایه را نمایش می‌دهد اما استفاده دوباره از دستور ReDim، تمام مقادیر قبلی در آرایه را از بین می‌برد، در نتیجه فقط مقادیر صفر توسط حلقه نمایش داده می‌شود.

حال اگر به جای دستور ReDim myarray (10) از دستور ReDim Preserve myarray (10) استفاده کنید، حلقه For دوم پس از نمایش مقادیر ۱ تا ۵ برای اعضای قبلی، مقدار صفر را هم برای ۵ عضو جدید که اضافه شده‌اند، نمایش می‌دهد.



تمرین:

یک پروژه جدید طراحی کنید و دستورات فوق را در رویداد Click یک دکمه فرمان قرار داده و نتیجه را در دو حالت بحث شده بررسی کنید.



نکته • در صورت کاهش ابعاد یک آرایه به وسیله دستور ReDim مقادیر مربوط به اعضای حذف شده از بین می‌روند.

• به وسیله دستور ReDim نیز می‌توانید یک آرایه پویا Dynamic تعریف کنید شکل کلی این دستور برای تعریف یک آرایه پویا به صورت زیر است:

نوع داده As (دامنه بالایی) نام آرایه ReDim
مثلاً دستور زیر یک آرایه با ۲۰ عضو و از نوع رشته‌ای تعریف می‌کند.

ReDim fam (19) As String

• در صورت استفاده از دستور ReDim برای تغییر ابعاد یک آرایه ثابت در هنگام اجرای برنامه، پیام خطای Array already dimensioned نمایش داده می‌شود.



تمرین:

می‌خواهیم پروژه‌ای که در مثال ۳ طراحی شده است به گونه‌ای تغییر کند تا تعداد اسامی با توجه به نیاز و درخواست کاربر دریافت و در آرایه ذخیره شود. به این منظور پروژه Array را باز کنید و عملیات زیر را به ترتیب انجام دهید:

۳-۱-۱۱ دستور Erase

از این دستور برای حذف اعضا در آرایه‌های پویا و از بین بردن محتویات اعضا در آرایه‌های ثابت استفاده می‌شود. شکل کلی دستور Erase به صورت زیر است:

نام آرایه Erase

اگر این دستور روی یک آرایه پویا اجرا شود تمام مقادیر آن از بین خواهد رفت و شما می‌توانید مجدداً با استفاده از دستور ReDim از آرایه استفاده کنید اگر پس از حذف اعضای یک آرایه پویا، بدون استفاده از دستور ReDim سعی در دستیابی به اعضای آن داشته باشید پیام خطای Subscript out of range نمایش داده می‌شود.

اگر این دستور روی یک آرایه ثابت اجرا شود مقادیر موجود در آرایه حذف خواهند شد، اما فضاهای مربوط به اعضای آرایه در حافظه از بین نخواهند رفت.

به عنوان مثال به دستورات زیر توجه کنید (با فرض این که ۱ Option Base است):

```
Dim myarray1() As Integer, myarray2(5) As Integer
```

```
Dim i As Integer
```

```
ReDim myarray1(5)
```

```
For i = 1 To 5
```

```
    myarray1(i) = i
```

```
    myarray2(i) = i
```

```
Next i
```

```
Erase myarray1, myarray2
```

```
ReDim myarray1(5)
```

```
For i = 1 To 10
```

```
    Print myarray1(i)
```

```
    Print myarray2(i)
```

```
Next i
```

در این مجموعه دستورات از یک آرایه پویا (myarray1) و یک آرایه‌ای با ابعاد ثابت (myarray2) استفاده شده است. در اولین حلقه For، آرایه‌های myarray1 و myarray2 به ترتیب مقداردهی شده‌اند سپس به وسیله دستور Erase، آرایه myarray1 از حافظه کاملاً پاک شده اما آرایه myarray2 فقط مقادیر خود را از دست می‌دهد و در نتیجه حلقه For دوم فقط مقادیر صفر را برای هر دو آرایه نمایش می‌دهد. البته اگر دستور ReDim myarray1(5) پس از حذف آرایه مزبور استفاده نشود آرایه در حلقه For قابل شناسایی نبوده و پیام خطای Subscript out of range نمایش داده می‌شود.



تمرین:

پروژه Array را به گونه‌ای تنظیم کنید که شرایط زیر را داشته باشد:
الف- امکان افزایش اسامی پس از پرشدن آرایه وجود داشته باشد.
ب- در صورت لزوم بتوان اعضای آرایه را حذف کرده و اسامی جدید در آن ذخیره نمود.

۲-۱۱ آرایه‌های چند بعدی

تاکنون با نحوه تعریف و چگونگی استفاده از آرایه‌های یک بعدی آشنا شدید اما در برنامه‌نویسی پروژه‌های واقعی، گاهی اوقات آرایه‌های یک بعدی نیز گره‌گشا نیستند و برنامه‌نویسی را با مشکلات متعددی روبه‌رو می‌کنند.

آرایه‌های دو بعدی از دو اندیس برای شناسایی اعضای خود استفاده می‌کنند. اندیس اول شماره سطر و اندیس دوم شماره ستون متناظر با آرایه را مشخص می‌کنند. برای روشن شدن بهتر موضوع به آرایه زیر توجه کنید، این آرایه دارای ۴ سطر و ۳ ستون است

جدول ۱۱-۳

شماره ستون \ شماره سطر	0	1	2
0	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
1	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$
2	$a_{2,0}$	$a_{2,1}$	$a_{2,2}$
3	$a_{3,0}$	$a_{3,1}$	$a_{3,2}$

در نوع از آرایه‌ها برای دسترسی به هر یک از عناصر آرایه از دو اندیس استفاده می‌شود مثلاً عنصر $a_{2,1}$ ، عضوی است که در سطر شماره ۲ و ستون شماره ۱ قرار دارد. برای تعریف آرایه دو بعدی می‌توانید یکی از این روش‌ها را به کار ببرید:

(دامنه بالایی ستون‌ها، دامنه بالایی سطرها) نام آرایه [Dim| Public| Private| Static]

نوع داده As (دامنه بالایی ستون‌ها، دامنه بالایی سطرها) نام آرایه [Dim| Public| Private| Static]

نوع داده As (دامنه بالایی ستون‌ها To دامنه پایینی ستون‌ها، دامنه بالایی سطرها To دامنه پایینی سطرها) نام آرایه [Dim| Public| Private| Static]

نکته با توجه به مکان تعریف و کاربرد آرایه می‌توانید یکی از کلمات کلیدی موجود در [] را انتخاب کنید.

به عنوان مثال فرض کنید می‌خواهید نمرات سه درس دانش‌آموزان یک کلاس ۵ نفره را به صورت یک آرایه دو بعدی بنویسید.

شماره ستون \ شماره سطر	ریاضی	فیزیک	شیمی	
	۰	۱	۲	
۰	۱۷	۱۴	۱۶	دانش‌آموز اول
۱	۱۲	۱۰	۸	دانش‌آموز دوم
۲	۲۰	۱۸	۱۵	دانش‌آموز سوم
۳	۱۴	۱۳	۱۹	دانش‌آموز چهارم
۴	۱۷	۲۰	۱۱	دانش‌آموز پنجم

اکنون می‌خواهیم این آرایه دو بعدی را تعریف کنیم بنابراین می‌توان یکی از موارد زیر را برای انجام این کار استفاده کرد:

Dim sng student (4,2) As Single

Dim sng student (0 To 4, 0 To 2) As Single

نکته توجه داشته باشید که قوانین دستور Option Base برای آرایه‌های دو بعدی نیز صادق است.

مثال ۴: رویه‌ای بنویسید که آرایه دو بعدی 4×4 را با اعداد تصادفی مقداردهی کرده سپس اعضای آن را نمایش دهد (با فرض Option Base 0).

Sub mymatrix()

Dim A(4, 4) As Integer, i As Integer, j As Integer

Randomize

For i = 0 To 3

For j = 0 To 3

A(i, j) = Int(100 * Rnd + 1)

Next j

Next i

For i = 0 To 3

For j = 0 To 3

Print " "; A("; i"; "; j"); "="; A(i, j)

Next j

Print

Next i

End Sub

رویه فرعی فوق در صورت اجرا، خروجی مشابه شکل زیر را خواهد داشت:

A(0,0)=0 A(0,1)=20 A(0,2)=17 A(0,3)=9

A(1,0)=2 A(1,1)= 0 A(1,2)=52 A(1,3)=53

A(2,0)=69 A(2,1)=44 A(2,2)=0 A(2,3)=31

A(3,0)=28 A(3,1)=87 A(3,2)=3 A(3,3)=0

در این رویه ابتدا آرایه دو بعدی A را تعریف کرده‌ایم سپس با استفاده از دو حلقه تو در تو و تابع Rnd عناصر آرایه مقداردهی شده‌اند. حلقه For اول به وسیله شمارنده i شماره سطر و حلقه دوم به وسیله شمارنده j شماره ستون را در آرایه کنترل می‌کنند بنابراین با هر بار اجرای حلقه اول، حلقه دوم سه بار تکرار می‌شود تا اعضای هر سطر را جهت مقداردهی به ترتیب آماده کند. به عنوان مثال زمانی که مقدار i برابر با صفر است حلقه دوم مقادیر z را به ترتیب روی مقادیر ۰، ۱، ۲ و ۳ تنظیم می‌کند. با خاتمه اجرای حلقه دوم، حلقه اول مقدار i را یک واحد افزایش می‌دهد و حلقه دوم مجدداً از z=۰ شروع شده و اعضای سطر دوم نیز به ترتیب مورد دسترسی قرار می‌گیرند. در پایان پس از مقداردهی آرایه، با استفاده از دو حلقه For تودرتو اعضای آرایه نمایش داده می‌شوند

تمرین:



پروژه‌ای طراحی کنید که بتواند نام، نام خانوادگی و نمره سه درس، پنج دانش‌آموز را در آرایه ذخیره کند و در ضمن بتوان هر لحظه اطلاعات دانش‌آموزان را مشاهده کرد.

۱۱-۳ توابع LBound و UBound

از این توابع برای دستیابی به دامنه پایینی و بالایی یک آرایه استفاده می‌شود. تابع LBound می‌تواند با دریافت نام یک آرایه، دامنه پایینی آن را به صورت یک عدد از نوع Long بازگرداند. به عبارت دیگر به وسیله این تابع می‌توانید شماره اولین اندیس را در آرایه به دست آورید. شکل کلی این تابع به این صورت است:

LBound (arrayname, dimension)

تابع LBound یک آرگومان اجباری و یک آرگومان اختیاری دارد، آرگومان arrayname در واقع نام آرایه‌ای است که می‌خواهید اندیس آغازین آن را به دست آورید. آرگومان دوم (dimension) در آرایه‌های دو و چند بعدی استفاده می‌شود و بعدی از آرایه که می‌خواهید اندیس آغازین آن را پیدا کنید به صورت یک عدد صحیح تعیین می‌شود. مقدار ۱ برای بعد اول (سطر)، ۲ برای بعد دوم (ستون) و الی آخر. در صورت عدم استفاده از این آرگومان، مقدار پیش فرض ۱ خواهد بود. به عنوان مثال به این دستورات توجه کنید:

```
Dim A(1 To 100, 10 To 13) As Integer
```

```
Dim B(10) As Integer
```

```
Print LBound(A, 1)
```

```
Print LBound(A, 2)
```

```
Print LBound(B)
```

در دستورات قبل یک آرایه دو بعدی (A) و یک آرایه یک بعدی (B) تعریف شده‌اند. نتیجه اجرای اولین و دومین دستور Print به ترتیب ۱ و ۱۰ خواهد بود و در صورت اجرای دستور آخر با توجه به عدم استفاده از آرگومان اختیاری و یک بعدی بودن آرایه مقدار صفر نمایش داده می‌شود. توجه داشته باشید که فرض بر این باشد که : Option Base 0 است.

تابع UBound می‌تواند با دریافت نام یک آرایه دامنه بالایی آن را به صورت یک عدد از نوع Long بازگرداند به عبارت بهتر عددی را که در تعریف آرایه به عنوان دامنه بالایی آرایه تعریف کرده‌ایم در اختیار شما قرار می‌دهد. شکل کلی این تابع به صورت زیر است:

UBound (arrayname ,dimension)

این تابع نیز یک آرگومان اجباری و یک آرگومان اختیاری دارد و عملکردی را مشابه آنچه در تابع LBound دارند به عهده می‌گیرند. به عنوان مثال به دستورات زیر توجه کنید:

```
Dim A ( 1 To 100, 10 To 13 ) As Integer
```

```
Dim B (10) As Integer
```

```
Print UBound(A, 1)
```

```
Print UBound(A, 2)
```

```
Print UBound(B)
```

اولین و دومین دستور Print به ترتیب مقادیر 10^0 و 13 را نمایش می‌دهند و دستور آخر نیز مقدار 10^0 را نمایش خواهد داد.

البته باید توجه داشت که دستور Option Base تأثیری روی تابع UBound نمی‌گذارد. به عنوان مثال به این دستورات توجه کنید:

```
Dim A(5) As Integer
```

```
For i = LBound(A) To UBound(A)
```

```
Print i
```

```
Next i
```

در دستورات قبل اگر Option Base 0 باشد حلقه For از مقدار صفر تا ۵ را نمایش می‌دهد و در واقع آرایه ۶ عضو خواهد داشت ولی اگر Option Base 1 باشد حلقه For از مقدار ۱ تا ۵ را نمایش می‌دهد و در نتیجه آرایه ۵ عضو خواهد داشت پس همان‌طور که دیدید دستور Option Base روی تابع UBound تأثیر نمی‌گذارد.

۴-۱۱ تابع Split

به وسیله این تابع می‌توان محتویات یک متغیر رشته‌ای را به صورت زیر رشته‌های جداگانه‌ای در یک آرایه یک بعدی ذخیره کرد. مقدار بازگشتی این تابع یک آرایه یک بعدی است. شکل کلی این تابع به صورت زیر است

Split (expression, delimiter, limit, compare)

این تابع دارای یک آرگومان اجباری است. expression می‌تواند یک عبارت رشته‌ای باشد. آرگومان delimiter یک آرگومان اختیاری است و به‌طور پیش‌فرض یک کاراکتر فاصله " " است. در واقع این آرگومان کاراکتری را که برای جدا کردن زیر رشته‌ها در آرگومان expression مورد استفاده قرار می‌گیرد معین می‌کند.

آرگومان limit دومین آرگومان اختیاری است و تعداد زیر رشته‌های مورد نظر را که باید بازگشت داده شوند تعیین می‌کند در صورت عدم استفاده از این آرگومان مقدار آن ۱- خواهد بود که به معنی بازگشت دادن تمام زیررشته‌هاست.

آرگومان compare نیز اختیاری بوده و تعیین می‌کند که آیا تابع بین حروف بزرگ و کوچک آرگومان delimiter با کاراکترهای expression تفاوت قائل شود یا خیر. مقادیری که این آرگومان می‌تواند دریافت کند در جدول ۴-۱۱ آرایه شده‌اند.

جدول ۴-۱۱ مقادیر مربوط به آرگومان Compare

ثابت رشته‌ای	ثابت عددی	توضیح
vbBinaryCompare	۰	تابع بین حروف کوچک و بزرگ تفاوت قائل می‌شود.
vbTextCompare	۱	تابع بین حروف کوچک و بزرگ تفاوت قائل نمی‌شود.

نکته مقدار پیش‌فرض برای آرگومان compare صفر است.



برای روشن شدن بهتر مطلب به رویه رویداد زیر توجه کنید:

```
Private Sub cmdsplit_Click ()
```

```
    Dim strname As String, stresult() As String
```

```
    Dim i As Integer
```

```
    strname = "microsoft visual basic 6"
```

```
    stresult = Split(strname)
```

```
    (For i = 0 To UBound(stresult)
```

```
        Print stresult(i)
```

Next i

End Sub

در دستورات فوق دستور Split براساس فضاهاى خالی « موجود در رشته strname, کلمات موجود در متغیر رشته‌ای strname را به صورت جداگانه در آرایه strresult ذخیره می کند که به وسیله یک حلقه For مطابق شکل ۱۱-۳ نمایش داده می شود.

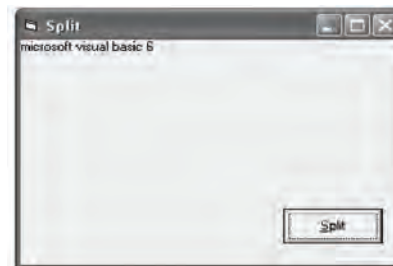


شکل ۱۱-۳

اکنون فرض کنید تابع Split را به صورت زیر تغییر دهید:

strresult = Split (strname , "A")

در این صورت کاراکتری که برای جدا کردن زیررشته‌ها استفاده می شود کاراکتر A خواهد بود. اما همان طور که در شکل ۱۱-۴ مشاهده می کنید رشته strname بدون توجه به کاراکتر A بدون تغییر باقی می ماند چون در این دستور از آرگومان compare استفاده نشده است مقدار پیش فرض صفر است در نتیجه تابع بین کاراکتر A در آرگومان دوم با کاراکتر a در رشته strname تفاوت قائل شده است.



شکل ۱۱-۴

حال اگر دستور قبل را به صورت (1) «A», strresult = Split (strname, تغییر دهید خروجی برنامه به صورت شکل ۱۱-۵ خواهد بود یعنی تابع بین کاراکتر A در آرگومان دوم با کاراکتر a در رشته strname تفاوتی قائل نمی شود و strname به سه زیر رشته تقسیم می شود.

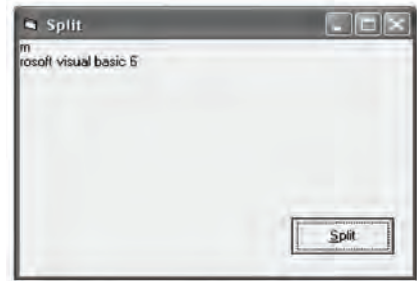
در صورتی که بخواهید تعداد معینی از زیررشته‌ها را جدا کنید می‌توانید از آرگومان سوم در تابع Split استفاده کنید. مثلاً اگر مثال قبل را به این صورت تغییر دهید:

```
strresult = Split(strname,"ic",2,1)
```

خروجی برنامه مطابق شکل ۱۱-۶ خواهد بود.



شکل ۱۱-۵



شکل ۱۱-۶

نکته در صورتی که آرگومان limit در تابع Split، ۱ باشد تمام رشته expression به صورت یک زیررشته بازگشت داده خواهد شد.

تمرین:



پروژه‌ای طراحی کنید که یک رشته و کاراکتر دلخواه را دریافت کند و رشته را براساس کاراکتر موردنظر به چند زیررشته تقسیم کرده و نمایش دهد.

۱۱-۵ تابع Join

عملکرد این تابع برعکس تابع Split است. به وسیله این تابع می‌توانید اعضای یک آرایه رشته‌ای را به صورت یک مقدار رشته‌ای واحد درآورید. شکل کلی این تابع به صورت زیر است:

Join (sourcearray, delimiter)

این تابع دارای یک آرگومان اجباری و یک آرگومان اختیاری است. آرگومان sourcearray، نام آرایه رشته‌ای مورد نظر است که می‌خواهید مقادیر موجود در آن را به صورت یکپارچه در یک متغیر رشته‌ای ذخیره کنید.

آرگومان delimiter یک آرگومان اختیاری است که کاراکتر جداکننده را بین کلمات در رشته‌ای که تابع بازگشت می‌دهد تعیین می‌کند. در صورت عدم استفاده از این آرگومان، کاراکتر فضای خالی " " مورد استفاده قرار می‌گیرد.
به‌عنوان مثال به دستورات زیر توجه کنید:

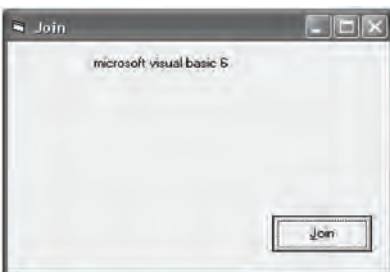
```
Private Sub cmdjoin_Click()  
    Dim strname(3) As String, strresult As String  
    strname(0) = "microsoft"  
    strname(1) = "visual"  
    strname(2) = "basic"  
    strname(3) = "6"  
    strresult = Join(strname)  
    Print  
    Print , strresult  
End Sub
```

در صورت اجرای دستورات فوق خروجی مشابه شکل ۷-۱۱ به دست می‌آید همان‌طور که می‌بینید مقادیر موجود در آرایه strname به‌وسیله تابع Join در متغیر strresult به‌صورت یک رشته واحد ذخیره شده‌اند و از کاراکتر فاصله خالی " " برای ایجاد فاصله بین مقادیر آرایه استفاده شده است.

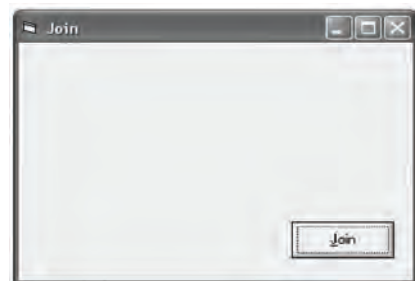
اکنون تابع Join را در مثال قبل به‌صورت زیر تغییر دهید:

```
strresult = Join ( strname , " # " )
```

در این صورت خروجی دستورات به‌صورت شکل ۸-۱۱ خواهد بود.



شکل ۷-۱۱



شکل ۸-۱۱



تمرین:

پروژه‌ای طراحی کنید که تعدادی کلمه را به صورت جداگانه دریافت کند سپس با استفاده از تابع Join آن‌ها را به یک رشته واحد تبدیل نموده و نمایش دهد.

۶-۱۱ نحوه ارسال آرایه‌ها به رویه‌ها

تاکنون با نحوه تعریف آرایه و چگونگی استفاده از آن‌ها آشنا شده‌اید اما گاهی اوقات لازم است تا آرایه‌ها را به یک رویه فرعی یا تابع ارسال کنید. در این جا با ذکر یک مثال با چگونگی انجام این کار آشنا می‌شوید.



مثال ۵: می‌خواهیم تابعی بنویسیم که با دریافت یک آرایه عددی از نوع Integer با ۱۰ عضو، کوچک‌ترین عضو را پیدا کرده و بازگرداند. ابتدا به تعریف تابع می‌پردازیم، در اینجا لازم است آرگومان ورودی به صورت آرایه استفاده شود بنابراین در هنگام تعریف آرگومان از دو پرانتز () استفاده می‌شود.

```
Public Function minelement(myarray() As Integer) As Integer
```

```
Dim i As Integer, min As Integer
```

```
min = myarray(0)
```

```
For i = 1 To 9
```

```
    If min > myarray(i) Then min = myarray(i)
```

```
Next i
```

```
minelement = min
```

```
End Function
```

همان‌طور که در تعریف تابع می‌بینید برای تعریف یک آرگومان از نوع آرایه فقط کافی است دو پرانتز در ابتدای نام آرایه ذکر کنید. برای فراخوانی این تابع نیز می‌توانید از دستور زیر استفاده کنید:

```
Print minelement (myarray ( ) )
```



نکته آرایه‌ها به‌طور پیش فرض به صورت ارسال با مرجع (Call By Reference) به رویه ارسال می‌شوند.

۷-۱۱ روش‌های مرتب‌سازی آرایه‌ها

در برنامه‌نویسی واقعی معمولاً لازم است تا اطلاعات دریافت شده یا نتیجه محاسبات را به صورت مرتب شده در اختیار کاربران قرار دهید. به این منظور می‌توان از آرایه‌ها استفاده کرد.

تاکنون روش‌های متعددی برای مرتب کردن داده‌های موجود در یک آرایه طراحی و معرفی شده‌اند که هر یک نقاط ضعف و قدرتی نیز دارند. از انواع روش‌های معروف در مرتب‌سازی اطلاعات می‌توان به روش مرتب‌سازی حبابی (Bubble Sort)، روش مرتب‌سازی Shell Sort، روش مرتب‌سازی سریع (Quick Sort)، روش مرتب‌سازی انتخابی (Selection Sort) و مرتب‌سازی به روش درج (Insertion Sort) اشاره کرد. بعضی از روش‌های نامبرده دارای الگوریتم پیچیده‌ای هستند که از حوصله این کتاب خارج است بنابراین دو روش مرتب‌سازی حبابی و مرتب‌سازی انتخابی (Selection Sort) ارائه می‌شود.

۱-۷-۱۱ روش مرتب‌سازی حبابی (Bubble Sort)

این روش ساده‌ترین روش مرتب‌سازی است و هر عضو از آرایه با اعضای بعدی آرایه مقایسه می‌شود و در صورت نیاز، عمل تعویض صورت می‌گیرد. در این روش چون هر عضو با سایر اعضای بعد از آن مقایسه می‌شود در آرایه‌های بزرگ زمان زیادی برای مرتب شدن داده‌ها مصرف می‌شود بنابراین توصیه می‌شود در صورت کوچک بودن ابعاد یک آرایه از این روش استفاده کنید. در زیر دستوراتی را می‌بینید که می‌تواند یک آرایه عددی از نوع صحیح با ۱۰ عضو را که قبلاً پر شده است به روش حبابی و به صورت صعودی مرتب کند (با فرض این که 1 Option Base است).

```
Private Sub cmdsort_Click()
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Dim temp As Integer
```

```
For i = 1 To 9
```

```
For j = 1 To 9
```

```
If myarray(j) > myarray(j + 1) Then
```



```
temp = myarray(j)
myarray(j) = myarray(j + 1)
myarray(j + 1) = temp
```

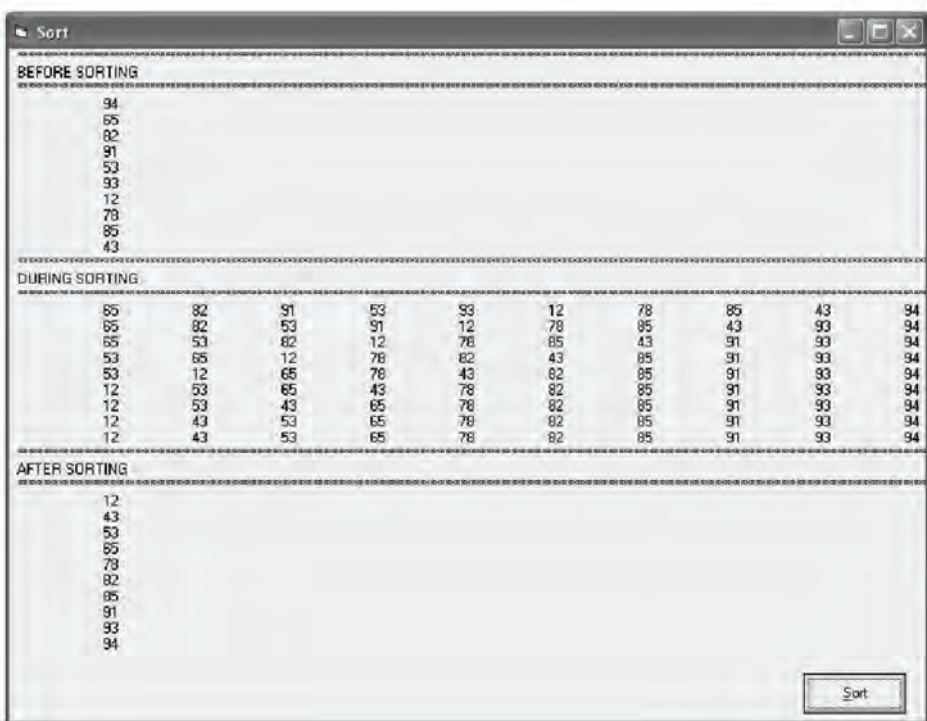
End If

Next j

Next i

End Sub

در این رویداد عمل مرتب‌سازی با دو حلقه که $n-1$ بار تکرار می‌شوند (n تعداد اعضای آرایه است) انجام می‌شود. در واقع به‌وسیله یک If در داخل حلقه دوم مقدار هر عضو با عضو بعدی مقایسه می‌شود و در صورت بزرگ‌تر بودن عضو که اندیس آن j است مقدار آن با مقدار عضو که اندیس آن $j+1$ است تعویض می‌شود. با ادامه روند عملیات، این کار آنقدر تکرار می‌شود تا آرایه مرتب شود؛ می‌توانید روند انجام عملیات را در شکل ۹-۱۱ مشاهده کنید: در ابتدا، مقادیر موجود در آرایه را قبل از مرتب شدن و در انتها، همان آرایه را به‌صورت مرتب شده می‌بینید در بخش میانی نیز اعضای آرایه را در هر بار اجرای حلقه اول (حلقه با شماره‌دهنده i) مشاهده می‌کنید. اگر به روند تغییر مکان اعداد توجه کنید نحوه مرتب‌سازی را کاملاً درک خواهید کرد.



شکل ۹-۱۱

تمرین:



پروژه‌ای طراحی کنید که ده عدد را از کاربر دریافت کند و پس از ذخیره‌سازی در یک آرایه، آرایه را به صورت نزولی مرتب کرده و نمایش دهد.

۲-۷-۱۱ روش مرتب‌سازی انتخابی

در این روش ابتدا کوچک‌ترین عضو آرایه تعیین و با عضو اول آرایه جابه‌جا می‌شود. در مرحله بعد کوچک‌ترین عضو آرایه در بین عضو دوم تا آخرین عضو آرایه پیدا شده و با عضو دوم آرایه جابه‌جا می‌شود و به همین ترتیب الی آخر تا آرایه مرتب شود.

```
Private Sub cmdsort_Click()
```

```
Dim length As Integer
```

```
Dim index As Integer
```

```
Dim min As Long
```

```
length = UBound(myarray)
```

```
For i = 1 To length
```

```
min = myarray(i)
```

```
index = i
```

```
For j = i + 1 To length
```

```
If min > myarray(j) Then
```

```
min = myarray(j)
```

```
index = j
```

```
End If
```

```
Next j
```

```
If index <> i Then
```

```
temp = myarray(i)
```

```
myarray(i) = myarray(index)
```

```
myarray(index) = temp
```

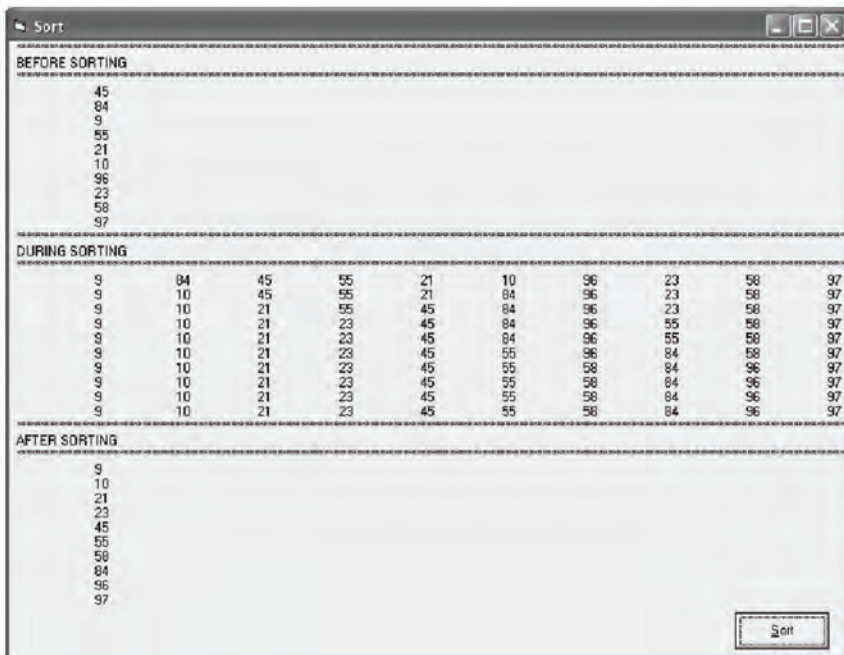
```
End If
```

```
Next i
```

```
End Sub
```

در این رویداد با استفاده از دو حلقه For تودرتو عملیات مرتب‌سازی انجام می‌شود در ابتدا نخستین حلقه For اولین عضو آرایه را به عنوان کوچک‌ترین مقدار در متغیر min ذخیره می‌کند. سپس حلقه For دوم این مقدار را با سایر اعضا یعنی اعضای دوم تا آخرین عضو مقایسه می‌کند اگر مقدار عضوی از مقدار min کمتر باشد مقدار کوچک‌تر را در متغیر min و اندیس آن را در متغیر index ذخیره می‌کند و پس از خاتمه اجرای این حلقه اگر مقدار کوچک‌تری پیدا شده باشد (index < i) آنگاه با استفاده از یک دستور If جای عضو اول با عضوی کوچک‌تر که اندیس آن در متغیر index ذخیره شده است عوض می‌شود.

در این مرحله با رسیدن Next i اجرای دستورات به ابتدای حلقه اول منتقل می‌شود. در این مرحله همان عملیات برای عضو دوم آرایه اجرا خواهد شد و به همین شکل آرایه مرتب می‌شود (شکل ۱۰-۱۱).



شکل ۱۰-۱۱

۸-۱۱ روش‌های جستجوی داده‌ها در آرایه‌ها

گاهی اوقات لازم است تا داده‌ای را میان مجموعه‌ای از اطلاعات موجود جستجو

کرده و مورد دستیابی قرار دهیم. یکی از دلایل استفاده از آرایه‌ها جستجوی سریع‌تر و راحت‌تر داده‌ها در میان انبوهی از اطلاعات است.

جستجوی داده در یک آرایه مانند مرتب‌سازی از روش‌های مختلفی امکان‌پذیر است که از مهم‌ترین آن‌ها می‌توان روش جستجوی خطی و روش جستجوی دودویی را نام برد.

۱-۸-۱ روش جستجوی خطی (Linear Search)

این روش یکی از ساده‌ترین روش‌های جستجو است در این روش برای پیدا کردن یک مقدار، مقدار مربوطه را یک به یک با اعضای آرایه مورد مقایسه قرار داده و در صورت پیدا شدن اطلاعات مورد نظر جستجو خاتمه می‌یابد و اگر مقدار مورد نظر در هیچ یک از اعضای آرایه پیدا نشود در آن صورت نتیجه جستجو منفی خواهد بود و مقدار مربوطه در آرایه وجود نخواهد داشت.

به‌عنوان مثال فرض کنید می‌خواهیم یک عدد را در آرایه‌ای با ۲۰ عضو جستجو کنیم. تابع mysearch می‌تواند با دریافت یک عدد و آرایه مربوطه، عدد مربوط را در آرایه جستجو کند و در صورت وجود عدد در آرایه، مقدار Yes و در غیر این صورت مقدار No را برگرداند.

```
Function mysearch(myarray() As Integer, mynumber As Integer) _ As Boolean
```

```
Dim i As Integer
```

```
For i = 0 To 19
```

```
    If mynumber = myarray(i) Then
```

```
        mysearch = True
```

```
    Exit Function
```

```
End If
```

```
mysearch = False
```

```
Next i
```

```
End Function
```

در این دستورات با استفاده از حلقه For و یک دستور If تک‌تک اعضای آرایه با مقدار متغیر mynumber مقایسه می‌شوند و در صورت پیدا شدن مقدار مشابه، مقدار درست (True) برگشت داده می‌شود و پس از آن به وسیله Exit Function خروج از تابع اتفاق می‌افتد.

اما اگر تمام اعضای آرایه بررسی شوند و مقداری برابر با mynumber پیدا نشود با بازگشت دادن مقدار نادرست (False) این مسأله را مشخص می‌کند.
به‌عنوان مثالی دیگر فرض کنید می‌خواهیم یک نام را در یک آرایه رشته‌ای با ۲۰ عضو مورد جستجو قرار دهیم، در این صورت تابع mysearch به این صورت در می‌آید:

```
Function mysearch(myarray() As String, myname As String) As Boolean
```

```
Dim i As Integer
```

```
For i = 0 To 19
```

```
    If StrComp(myarray(i), myname, 1) Then
```

```
        mysearch = True
```

```
    Exit Function
```

```
End If
```

```
mysearch = False
```

```
Next i
```

```
End Function
```

در تابع فوق نیز نحوه انجام عملیات مانند تابع قبل است و فقط برای پیدا کردن نام مورد نظر (myname) در آرایه از تابع StrComp استفاده می‌شود.

تمرین:



پروژه‌ای طراحی کنید که ده عدد را که به صورت تصادفی تولید شده‌اند در آرایه قرار دهد و به کاربر نشان دهد سپس کاربر عدد دلخواهی را در یک کنترل کادر متن وارد کند و با کلیک روی دکمه فرمان Search به روش جستجوی خطی به جستجوی عدد مورد نظر پردازد.

۲-۸-۱۱ روش جستجوی دودویی (Binary Search)

روش جستجوی خطی در آرایه‌های بزرگ، بسیار کند و وقت‌گیر است، روش جستجوی دو دویی با استفاده از این راه‌حل ساده، تعداد مقایسه‌ها را به مقدار زیادی کاهش می‌دهد.

در این روش ابتدا آرایه را مرتب کرده سپس عضو میانی آرایه با مقدار مورد جستجو، مقایسه می‌شوند، اگر مقدار مورد جستجو کوچک‌تر از عضو میانی باشد جستجو در

قسمت پایینی عضو میانی صورت می‌گیرد و اگر مقدار مورد جستجو بزرگ‌تر از عضو میانی باشد جستجو در قسمت بالایی عضو میانی انجام می‌گیرد سپس همین اعمال برای نیمه پایینی و یا بالایی در آرایه انجام می‌شود و بدین صورت تا زمان پیدا شدن مقدار مورد نظر یا با نصف شدن تعداد اعضای باقیمانده عملیات ادامه می‌یابد.
بنابراین می‌توانید تابع Binary Search را به این صورت بنویسید:

Public Function mysearch(myarray() As Integer, mynum As Integer) As Boolean

Dim i As Integer, intlow As Integer

Dim inthigh As Integer, intmid As Integer

intlow = LBound(myarray)

inthigh = UBound(myarray)

Do While (inthigh >= intlow)

intmid = Int((intlow + inthigh) / 2)

If mynum = myarray(intmid) Then

mysearch = True

Exit Function

ElseIf myarray(intmid) < mynum Then

intlow = intmid + 1

Else

inthigh = intmid - 1

End If

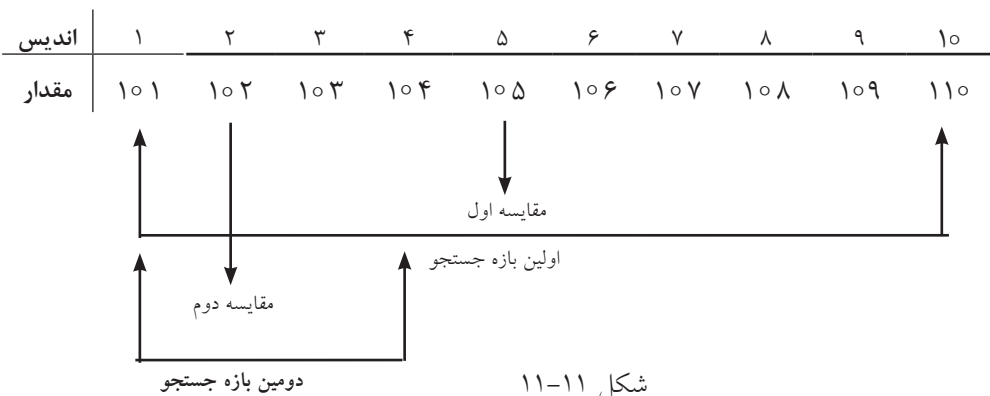
Loop

mysearch = False

End Function

این تابع با دریافت یک آرایه و عدد مورد نظر برای جستجو تعریف می‌شود پس از تعریف متغیرهای مورد نظر، مقدار اندیس بالایی و پایینی در آرایه را به وسیله توابع LBound, UBound در متغیرهای inthigh و intlow قرار می‌دهد. سپس یک حلقه Do while عملیات جستجو را انجام می‌دهد. ابتدا مقدار اندیس میانی در آرایه محاسبه می‌شود برای این کار اندیس پایینی با بالایی جمع شده و مقدار صحیح تقسیم آن‌ها بر عدد ۲ توسط تابع Int به دست می‌آید.

فرض کنید که آرایه دارای ۲۰ عضو است و اندیس پایینی ۱ است. با این شرایط مقدار `intmid` برابر با ۵ خواهد بود. پس از محاسبه مقدار اندیس عضو میانی به وسیله فرمان `if` تساوی عضو میانی (`myarray(5)` با عدد مورد جستجو `mynum` بررسی می‌شود، در صورت مساوی بودن آن‌ها با یکدیگر مقدار `True` بازگشت داده خواهد شد و به وسیله فرمان `Exit Function` اجرای برنامه به محل فراخوانی منتقل می‌شود اما اگر تساوی برقرار نباشد شرط موجود در `ElseIf` بررسی می‌شود. اگر در این مرحله مقدار عضو (`myarray(5)` کوچک‌تر از مقدار `mynum` باشد چون آرایه مرتب است بنابراین مقدار اندیس پایینی به وسیله فرمان `intlow = intmid + 1` به مقدار بعد از اندیس عضو میانی (یعنی ۶) تغییر پیدا خواهد کرد و حلقه در اجرای دوباره دستورات، عناصر موجود در بین اندیس ۶ و ۱۰ را مورد جستجو قرار می‌دهد؛ اما اگر شرط موجود در `ElseIf` غلط باشد بنابراین عنصر مورد جستجو در نیمه پایینی آرایه قرار دارد و با استفاده از بخش `Else` مقدار اندیس بالایی یکی کمتر از اندیس میانی (یعنی ۴) خواهد شد و در اجرای دوباره حلقه، عناصر موجود در بین اندیس ۱ و ۴ مورد جستجو قرار می‌گیرند، به این ترتیب با کوچک‌تر شدن بازه جستجو، در صورتی که مقدار مورد نظر در آرایه کشف نشود مقادیر `intlow` و `inhigh` به گونه‌ای تغییر می‌کنند که نتیجه شرط `inhigh >= intlow` نادرست شود که در نتیجه اجرای حلقه خاتمه یافته و مقدار `False` بازگشت داده می‌شود مثلاً فرض کنید که می‌خواهیم عدد ۲۰ را در آرایه ۱۰ عضوی (مقدار اعضای آن از ۱۰۱ تا ۱۰ هستند) جستجو کنیم. عملکرد تابع به این صورت نمایش داده می‌شود:

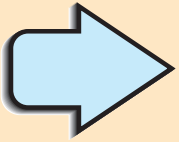


شکل ۱۱-۱۱

تمرین:



پروژه تمرین قبل را با روش جستجوی دودویی انجام دهید.



Learn in English

Array Variables

Much of the time, you just want to assign a single value to a variable you've declared.

A variable containing a single value is a scalar variable.

Other times, it's convenient to assign more than one related value to a single variable.

Then you can create a variable that can contain a series of values.

This is called an array variable. Array variables and scalar variables are declared in the same way,

except that the declaration of an array variable uses parentheses () following the variable name.

In the following example, a single-dimension array containing 11 elements is declared:

```
Dim A(10) As Integer
```


واژه‌نامه

Array	آرایه
Binary Search	جستجوی دودویی
Bound	مرز، محدوده
Bubble Sort	مرتب‌سازی حبابی
Convenient	راحت، مناسب
Delimiter	حائل
Dimension	اندازه، بعد
Dynamic	پویا
Element	عنصر
Erase	پاک کردن
Expression	عبارت
Join	متصل کردن
Limit	حد، اندازه
Linear Search	جستجوی خطی
Random	تصادفی
Related	مربوط
Scalar	عددی
Selection Sort	مرتب‌سازی انتخابی
Split	از هم جدا کردن
Subscript	اندیس، زیرنویس

خلاصه مطالب

- از آرایه‌ها برای ذخیره‌سازی و نگهداری آسان‌تر داده‌ها، مرتب‌سازی و جستجوی سریع اطلاعات استفاده می‌شود.
- آرایه‌ها به دو دسته کلی، آرایه‌ها با ابعاد ثابت و آرایه‌های پویا یا با ابعاد متغیر تقسیم می‌شوند.
- به وسیله فرمان Option Base می‌توان شماره اولین عنصر آرایه را تعیین کرد.
- در آرایه‌های پویا (Dynamic) تعداد اعضای آرایه با توجه به تعداد مورد نیاز معین می‌شوند اما در آرایه‌های ثابت تعداد اعضای آرایه بعد از تعریف آن قابل تغییر نیست.
- برای قابل استفاده شدن یک آرایه پویا پس از تعریف آرایه از دستور ReDim برای تعیین ابعاد آن استفاده می‌شود.
- در صورتی که بخواهید مقادیر موجود در یک آرایه پویا در زمان تغییر ابعاد آن از بین

نرود از کلمه کلیدی Preserve استفاده کنید.

- به وسیله فرمان Erase می توان آرایه های پویا را حذف کرد.
- مقادیر موجود در یک آرایه با ابعاد ثابت به وسیله فرمان Erase از بین خواهند رفت اما فضاهای مربوط به اعضای آرایه در حافظه باقی می ماند.
- هنگام ارسال آرایه ها به رویه ها به طور پیش فرض از حالت ارسال با مرجع استفاده می شود.
- روش های مرتب سازی آرایه ها عبارتند از: مرتب سازی حبابی، مرتب سازی به روش Shell Sort، روش مرتب سازی Quick Sort، روش مرتب سازی انتخابی (Selection Sort) و روش مرتب سازی Insertion Sort.
- برای جستجوی اطلاعات در یک آرایه می توان از روش جستجوی خطی یا دودویی استفاده کرد.
- توابع UBound و LBound به ترتیب دامنه بالایی و پایینی آرایه مورد نظر را محاسبه می کنند.
- تابع Split می تواند یک عبارت رشته ای را به صورت زیر رشته های جداگانه در یک آرایه یک بعدی ذخیره کند.
- تابع Join برخلاف تابع Split، می تواند مقادیر رشته ای موجود در یک آرایه رشته ای را به صورت یک عبارت رشته ای واحد درآورد.

آزمون نظری

۱- در ویژوال بیسیک اندیس اول یک آرایه به‌طور پیش فرض از چه شماره‌ای آغاز می‌شود؟

الف- ۱ ب- صفر ج- ۱ د- ۲

۲- پس از تعریف یک آرایه پویا به‌وسیله کدام دستور آرایه قابل استفاده می‌شود؟

الف- Dim ب- Static ج- Erase د- ReDim

۳- به‌وسیله کدام فرمان می‌توان شماره اندیس اولین عضو آرایه را تعیین کرد؟

الف- Option Compare Text ب- Option Explicit

ج- Option Base د- Option Keyword

۴- آرایه (۵, ۱۰) No چند عضو دارد؟ (Option Base 1)

الف- ۳۶ ب- ۵۰ ج- ۴۵ د- ۴۰۰

۵- فرمان Erase می‌تواند آرایه‌های را حذف کند.

الف- پویا ب- با ابعاد ثابت

ج- رشته‌ای د- گزینه‌های الف و ب صحیح هستند.

۶- اگر (۱۰) name باشد حاصل عبارت UBound (name) چیست؟

الف- ۸ ب- ۹ ج- ۱۰ د- ۱۱

۷- آرایه (Color (1 To 5, 3 To 6) چند عضو دارد؟

الف- ۲۰ ب- ۲۵ ج- ۳۰ د- ۳۵

۸- کدام تابع می‌تواند اعضای یک آرایه رشته‌ای را به یک عبارت رشته‌ای واحد تبدیل کند؟

الف- Split ب- Join ج- Ubound د- Preserve

۹- در صورتی که بخواهیم در زمان تغییر ابعاد یک آرایه پویا مقادیر قبلی آرایه از بین

نرود از کدام کلمه کلیدی همراه با دستور ReDim استفاده می‌کنیم؟

الف- Dim ب- ParamArray ج- Para د- Preserve

۱۰- کدام تابع می‌تواند یک عبارت رشته‌ای را به‌صورت کلمات جداگانه در یک آرایه

ذخیره کند؟

الف- Split ب- Join ج- Preserve د- Filter

۱۱- Which of the following answer can be use to create a single dimension array With 5 elements?

a- Dim no (5) As Integer

b- Dim no (4) As Integer

c- Dim no [5] As Integer

d- Dim no [4] As Integer

- ۱۲- آرایه را تعریف کنید و انواع آن را نام ببرید.
- ۱۳- دستور Erase, Split و Join را با ذکر مثال توضیح دهید.
- ۱۴- نحوه تعریف و استفاده از آرایه‌های استاتیک و پویا را با ذکر مثال بیان کنید.
- ۱۵- توابع UBound و LBound را با ذکر مثال توضیح دهید.
- ۱۶- نحوه مرتب سازی آرایه‌ها با روش حبابی و انتخابی را بیان کنید.
- ۱۷- کاربرد دستور Option Base را با ذکر مثال توضیح دهید.
- ۱۸- نحوه ارسال آرایه‌ها به رویه‌های تابعی و فرعی را بیان کنید.
- ۱۹- انواع روش‌های جستجوی داده‌ها در آرایه‌ها را توضیح دهید.

آزمون عملی

- ۱- رویه‌ای بنویسید که یک آرایه 5×5 را با اعضای تصادفی ایجاد کرده سپس مجموع هر سطر و هر ستون آرایه را به‌طور جداگانه به همراه خود آرایه نمایش دهد.
- ۲- رویه‌ای بنویسید که دو آرایه دلخواه را دریافت کرده و آرایه حاصل ضرب آن دو را به همراه دو آرایه نمایش دهد.
- ۳- رویه‌ای بنویسید که دو آرایه یک بعدی دلخواه را دریافت کرده و سپس آرایه حاصل جمع، حاصل ضرب و حاصل تفریق آن دو را در آرایه‌های جداگانه‌ای ذخیره کند.
- ۴- رویه‌ای بنویسید که یک آرایه عددی با ۲۰۰ عضو را به‌صورت نزولی مرتب کرده و نمایش دهد.