Project 2

Astrological Toys

In this project you will write a series of Windows applications that look up and display astrological signs and dates. The applications that will make up the project are as follows. There are two preparation applications:

- This Time – a digital clock with the date on the window bar [In Lab Exercise]
- Sign Finder – say your birthday interval and get your astrological sign back [In Lecture Demonstration]

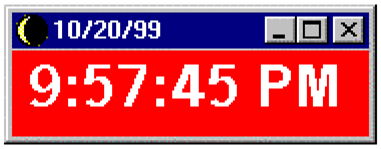
There are two parts to be turned in:

- Zodiac Range -- give your sign and receive the date interval containing your birthday [Part 1]
- Day Finder – generalizing the Zodiac Range to ask a series of questions to find the respondent's birthday using binary search [Part 2]

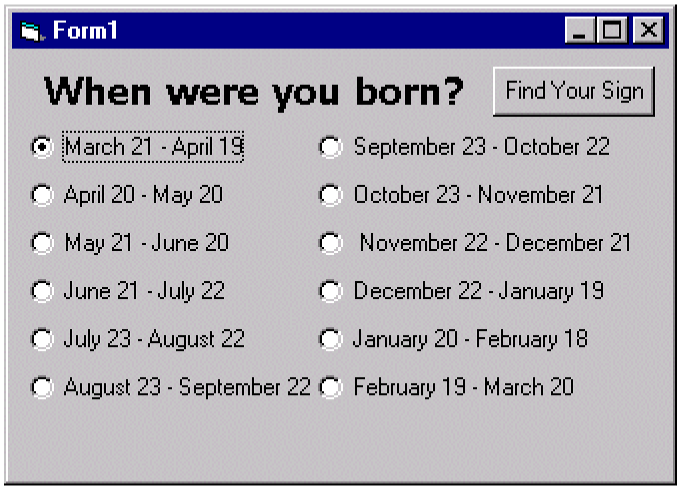
The projects are written in Visual Basic 6.0 and serve as the initial foray into algorithmic thinking, program design and the characteristics of VB6.

It is suggested you create a new folder for each project. This will assist you in keeping track of all of the various files produced as well as simplifying saving your work to the floppy disk.

This Time: This is a simple digital clock program with the date on the window bar. The GUI has the following form. The solution will be developed in Lab 8.



Sign Finder: The SignFinder program will be discussed in class. Sign Finder has the following graphic interface.

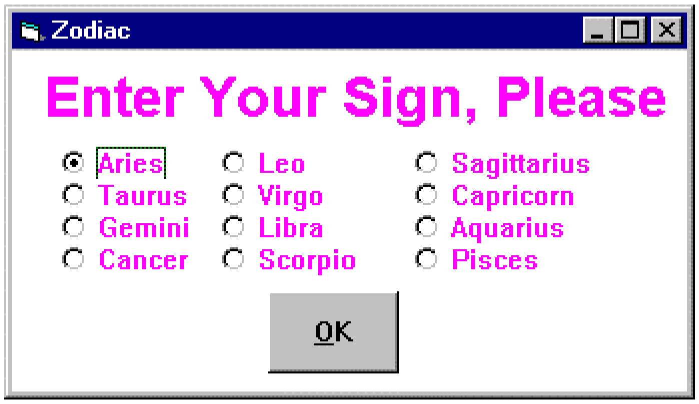


Part 1: Zodiac Range

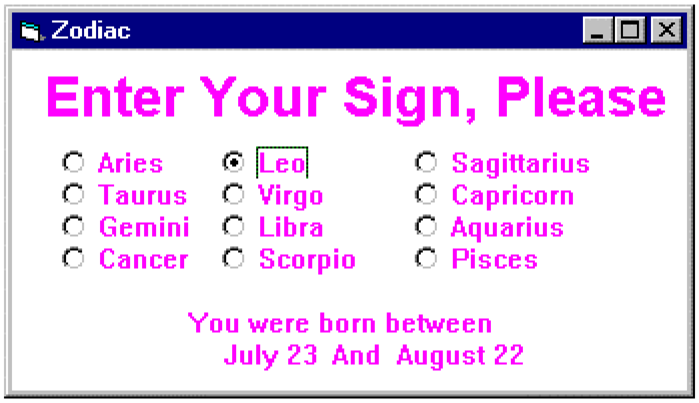
The objectives of this project are --

- Create your first VB application
- Develop experience with form creation and refinement
- Understand how a computation is expressed in a programming language
- Understand the flow of information through variables
- Understand the concept of "event programming"
- Practice using the Visual Basic IDE and managing project development

The application will present to the user a set of 12 radio buttons corresponding to the Zodiac signs, and when one is set, followed by a Click of OK, will print out the starting and ending dates of that sign. The main task is to construct the form to have the 12 radio buttons, one command button and five labels. Zodiac Range has the following graphic interface



The initial window for the Zodiac application.



You were born between
July 23 And August 22

The window resulting from clicking "Leo" and "OK"

The overall logic is as follows. The form contains all of the text and controls shown in both displays above. For the initial display, the "You were born between" text, the dates and the "And" are all hidden. When the command (OK) button is pushed, the command button is hidden, and the "You were born between" text etc. are made visible.

Detailed Steps of What to Do

(a) Create a new project and a form template.
(b) Revise the form's name to "`frmZodiac`".
(c) Save the form and the project, selecting "`Zodiac`" as the project name.
(d) Adopt a new background color, and any other form-based customization that you think is appropriate, like an attractive font and font color.
(e) Double click on the form to bring up the code window.
(f) After the `Option Explicit` line or at the top of the code window if the `Option Explicit` is not shown, enter the variable declarations,

```
Dim loMo As String
Dim hiMo As String
Dim loDate As Integer
Dim hiDate As Integer
```

Notice how VB's "word completion" tries to assist after the "`As`" is entered. The two string variables will represent the two months that the astrological sign spans, and the two integer variables will represent the starting and ending days. So, for the Leo Zodiac sign, the variable `loMo` will be assigned the value "`July `" and `hiMo` will be assigned "`August `". The `loDate` variable will be assigned 23 and the `hiDate` will be assigned 22, since the Leo sign spans the interval July 23 to August 22.

Sign	Start Date	End Date
Aries	March 21	April 19
Taurus	April 20	May 20
Gemini	May 21	June 20
Cancer	June 21	July 22
Leo	July 23	August 22
Virgo	August 23	September 22
Libra	September 23	October 22
Scorpio	October 23	November 21
Sagittarius	November 22	December 21
Capricorn	December 22	January 19
Aquarius	January 20	February 18
Pisces	February 19	March 20

Table 1. Listing of astrological signs

(g) In the form, place the 12 option or radio buttons, naming each with a new name. The names should begin with the letters "`opt`" and be followed by the first three letters of the sign. The name for the Aries button will be `optAri`. Change the caption on the button to the intended Zodiac sign. Revise the background color, font and any other features of the button needed to make the application appear attractive.
(h) Once the buttons are defined, place labels for the text lines on the form:

```
"Enter your Zodiac sign"
"You were born between"
"starting date"
"and"
"ending date"
```

Name the labels `lblHead`, `lblBorn`, `lblSDate`, `lblAnd`, `lblEDate`, respectively. Revise the fonts, background colors and any other needed properties. The last four labels should be hidden, i.e. have their visibility property set to "`false`".

Click on each radio button to bring up the "click event" handler for that button. Customize it by setting the `loMo` to the starting month (followed by a space) in quotes, `hiMo` to the ending month (followed by a space) in quotes, `loDate` to the starting day for the sign, and `hiDate` for the ending day for the sign. The extra space is included so the day doesn't smash into the month when printed out. For example, the Aries "click event" handler would be

```
Private Sub optAri_Click()
    loMo = "March "
    hiMo = "April "
    loDate = 21
    hiDate = 19
End Sub
```

There is one further trick that can be used to make the text appear more natural when it is printed. Assume that the start date precedes the "` and `", which is followed by the end date. If these are all printed on the same line, and the start date is right justified, then the two dates will "hug" the "and" and make the text appear without spaces.

Start date right justified →	" and "	← End date left justified

(i) Finally, add a command button, renamed `cmdOK` and captioned `&OK`. Modify its visual properties appropriately. Click on the OK command button to bring up the cmdOK_Click click event handler. Set the start and end dates by changing the captions as follows:

```
lblSDate.Caption = loMo & loDate
lblEDate.Caption = hiMo & hiDate
```

Further, set the visibility of the `lblBorn`, `lblSDate`, `lblAnd`, `lblEDate`, to `True`, and visibility of cmdOK to be `False`.

The program should now run. Save it. Try it out. Show it to your friends, and notice their utter amazement that a computer could be made to do such an impressive feat, a skill once reserved only for astrologers!

Grading Criteria for Zodiac Range

Part 1 of the Project is graded on the following criteria:

- Structure and organization of the form, including how attractive the layout is
- Operation of the application -- does it work
- Completeness -- do all aspects of it work

The project is **due 27 April 2001**. The electronic turn-in link will be found on the Projects page.

Part 2: Day Finder

The Day Finder, though a small extension to the Zodiac program, is significant because it introduces a powerful algorithm, binary search.

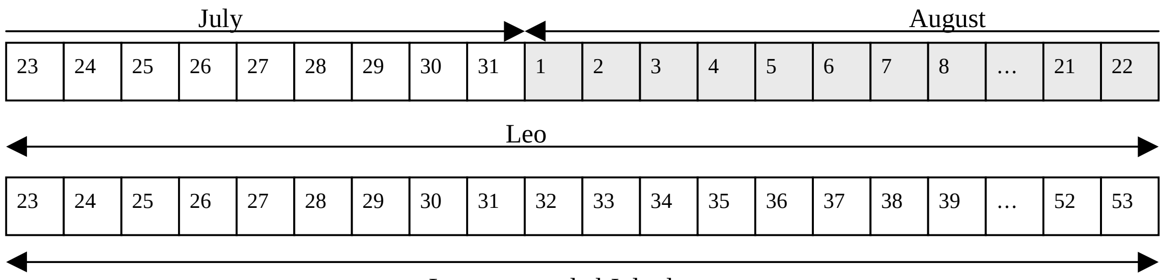
Searching for a day. The Day Finder is an extension of the Zodiac application, in which the user is asked questions that enable the program to say what day he or she was born on. So, Leo's are known by the Zodiac application to be born between July 23 and August 22. (Remember this includes both the starting and ending dates.) The task is to ask the user questions to determine which day was his or her actual birthday. One strategy would be to ask, "Were you born after July 23?" If the answer was "no", then the program could say, "You were born on July 23", but if the answer were "yes", the program would have to try another question, like "Were you born after July 24?" If the person were born on August 22, this strategy would take 30 questions to cover the 9 days in July (23-31) and the 22 days in August (1-22), which is actually 31 days, but we don't have to ask about the last day, since a "no" to "Were you born after August 21" implies an August 21 birthday, while a "yes" implies an August 22 birthday.

The Day Finder will use an intelligent searching strategy, called the *binary search algorithm*. Binary search can be used on any ordered objects, like letters or the dates of a Zodiac sign. It uses a series of questions, sometimes called *probes*, to eliminate half of the remaining possibilities with each question. For example, suppose I'm thinking of a letter, you may ask, "Is the letter between N and Z?", a question that is equivalent to asking if it is in the last half of the alphabet or "Is the letter after M?". Regardless of which answer is given, half of the alphabet is eliminated because a "yes" implies the letter is between N and Z, and a "no" implies it is between A and M. The binary search for the letter "L" goes as follows:

Question	Ans	Letters Eliminated
Is the letter after M?	N	A B C D E F G H I J K L M ~~N O P Q R S T U V W X Y Z~~
Is the letter after G?	Y	~~A B C D E F G~~ H I J K L M ~~N O P Q R S T U V W X Y Z~~
Is the letter after J?	Y	~~A B C D E F G H I J~~ K L M ~~N O P Q R S T U V W X Y Z~~
Is the letter after L?	N	~~A B C D E F G H I J~~ K L ~~M N O P Q R S T U V W X Y Z~~
Is the letter after K?	Y	~~A B C D E F G H I J K~~ L ~~M N O P Q R S T U V W X Y Z~~
The letter must be L		

Notice three points about the example: First, the questions always ask if the letter is *after* the probe letter. It is possible to ask *before* and *after* questions, but always asking the same type simplifies the algorithm. Second, the intervals sometimes have an odd number of letters and sometimes an even number. When it is an odd number, then the middle letter is the probe. When it is an even number, e.g. A-Z, then the probe is the last element of the first half of the interval. This is because of the "after formulation" of the questions. If the questions had been formulated as *before* questions, then the first element in the second half would be used. Third the binary search scheme requires only five questions to find any letter. Scientifically, and only for those interested in such things, the binary search takes $log_2\, n$ probes to find an item among n items, which is the minimum. The thing to remember is that this is the best way to find an item among an ordered sequence.

Finding a birthday works in the just the same way. The only complication is that the sequence of days changes months under the sign, e.g. Leo goes from July to August. This is not really a problem, however, if we change slightly how we think about dates. Imagine that the month didn't change, then we would look for the birthday from July 23rd through July 53rd. Of course, there is no July 53; that's what August 22 would be if the month didn't change.



Leo as extended July dates

The 53 can be found by adding the number of days in July, 31, to the last day of the sign, the 22nd, giving 31 + 22 = 53. This means that the search can look for the birthday between 23 and 53, and whenever a question must be asked about a date larger than 31, the day gets 31 subtracted from it and the month becomes August. We will call this the **Extended Month Technique**.

For example, to figure out the first question to ask, we find the mid point between July 23 and July 53 by subtracting the smaller endpoint from the larger:

$$53 – 23 = 30$$

and dividing by 2

30 / 2 = 15

If the number had been odd, the 0.5 would have been truncated (or rounded down). Then this number is added to the start of the Leo sign, the 23rd

23 + 15 = 38

which is a day in August since it is larger than the largest day in July, the 31st. Thus, July 38 is really August 7th, since 38-31 = 7. (See diagram above.) And, it can be checked that this is the middle of the Leo range, just by counting out the days. The result of this computation would be to ask the question:

Were you born after August 7?

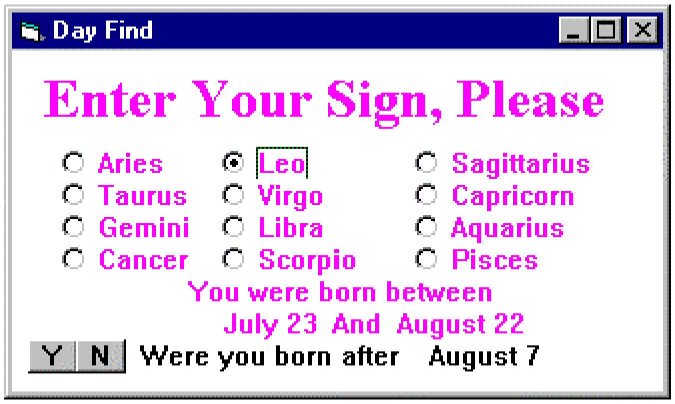
which to us is July 38. If the answer is "no," then the interval is 23 to 38. Remember, August 7 (= July 38) is still a possibility. If the answer is "yes" then the interval is 39 to 53. By repeatedly shrinking the interval we can reduce the interval size to one day, say 32 to 32. And that must be the person's birthday.

The questions to find the birthday August 1 are as follows, where Lo is the lower end of the interval, Hi is the upper end of the interval, Mid is the computed midpoint and Size is the size of the interval.

Lo	Hi	Size	Mid	Question	Ans	NewLo	New Hi
23	53	30	38	Were you born after August 7?	No	23	38
23	38	15	30	Were you born after July 30?	Yes	31	38
31	38	7	34	Were you born after August 3?	No	31	34
31	34	3	32	Were you born after August 1?	No	31	32
31	32	1	31	Were you born after July 31?	Yes	32	32
				You were born on August 1.			

Table 2: Example of using the Extended Month Technique

Notice that when the endpoints are the same, then they must be for the birthday being sought.


Day Find

Enter Your Sign, Please

Aries | [✓ Leo] | Sagittarius
Taurus | Virgo | Capricorn
Gemini | Libra | Aquarius
Cancer | Scorpio | Pisces

You were born between
July 23 And August 22

Were you born after August 7

Window asking the first question for the Day Finder application.

Encoding the Binary Search.

The objectives of this part of the project are:

- Create a non-trivial VB application
- Develop experience with more advanced form management
- Understand control flow through conditionals implementing complex logic

- Formulate and construct a dialog using a series of Click events
- Transfer an abstract algorithm into a practical program
- Increase familiarity with the details of binary search, including "end conditions"

The Day Finder will be an extension of the Zodiac program, so making a copy of that form is the way to begin. Use `Save As` in the `File` menu. Next, add four more variables to the integer declarations at the start of your program

```
Dim loMo As String
Dim hiMo As String
Dim loDate As Integer
Dim hiDate As Integer
Dim loEnd As Integer          New variable
Dim hiEnd As Integer          New variable
Dim midPt As Integer          New variable
Dim lastDay As Integer        New variable
```

The pair `loEnd` and `hiEnd` will bound the interval being searched`,` `midPt` will be used to represent the probe point in the center of the interval and `lastDay` will be the number of days of the month in which the sign begins to implement the Extended Month Technique.

Overall Logic

The Day Finder follows the Zodiac Range program. The program then asks a series of questions that implement a binary search over the range set by Zodiac Range until it gets so small as to span only a single day, which is the user's birthday. To program such a computation, break the task into a series of parts:

- Additional variable specifications
- Form modifications, including new buttons and text
- Program code to set up for the search and ask the first question [OK]
- Program code for the Yes/No buttons to ask additional questions [Y, N]
- Program code for the OK/Yes/No buttons to print out the guess in the right form [OK, Y, N]
- Program code to reduce the interval by half [Y, N]
- Program code for the Yes/No buttons to recognize when the end of the process and announce the result [Y, N]

Notice that except for form modifications and variable declarations the steps can all be associated with one or more events, given in brackets. This is a feature of Event Based Programming -- the whole process is arranged so that when an event like a button click happens, the program takes the proper action. Thinking of what that action should be is an easy way to understand the logic.

Detailed Logic

The following discussion amplifies on the five steps in the Overall Logic.

a) **Additional Variable Specifications.** The event handler for each sign's radio button, which already contains assignment statements for `loMo`, `hiMo`, `loDate` and `hiDate`, must have an assignment to `lastDay`, the last day of the month given by `loMo`. For example, for the Leo event handler, the `loMo` is "July ", and the last day of July is the 31st. So, `lastDay = 31` would be included in the Leo event handler. The purpose of keeping track of the last day of the month is so that the lower month can be "extended" as explained above.

b) **Form Modifications.** The form for the Day Finder is different from the form of the Zodiac Range. Add the items to the form, remembering that (a) they must be invisible initially, and (b) become visible at the time of the OK click. Setting them invisible initially is performed by changing their properties; making them visible at the OK Click is done by program code assigning to the visibility property.

c) **Search Set-up.** Once the form's visibility has been set, the binary search must be set up and the first question asked. The range will be bounded by `loEnd` and `hiEnd`, which should be assigned values based on the Extended Month Technique. The proper values to assign to `loEnd` and `hiEnd` are explained above; see Table 2 especially. Once the interval is initialized, the first question is asked.

d) **Compute the Probe.** Questions will be asked in OK (the first question) and in Y and N events. To find the probe requires the `midPt` to be computed, as described above. *It is important to note* that when dividing by 2 to compute the `midPt`, that integer division symbol (\) should be used rather than the normal division symbol (/), since doing so will get the rounding right. Notice that this computation will be performed in three different places in the program.

e) **Printing out the Guess.** The guess must be printed as "Were you born after " *month day*? The `midPt` value, which ranges over the Extended Month, can be used to determine the proper values for *month* and *day*. If `midPt` is less than or equal to `lastDay` then the month is the lower month (`loMo`), otherwise it is the higher month. Also, if `midPt` is less than `lastDay`, then `midPt` is the proper value for *day*. Otherwise, by the Extended Month Technique `midPt` is too large by the amount of `lastDay`, which must be subtracted off to get the right *day* value. To hook together the *month* word and the *day* number to form the guess to be displayed, use the concatenate operation (`&`) which is used to put two letter strings together. (The number will be converted to a letter string.) Assigning the result to the caption property of the text control used for guessing (introduced in (b) above), prints out the guess and waits for the user.

f) **Reducing the interval by half.** When the user clicks Y or N, he or she has given birthday information and the interval must be reduced before the next question is asked. Since the questions are "after" questions, a Y reply means that the lower half of the interval can be discarded. So, the `loEnd` should be moved up to just beyond the `midPt`, i.e. `midPt + 1`. If the reply is N, then the upper part of the interval is to be discarded, and `hiEnd` should be moved down to `midPt`. (Notice that there is an asymmetry here: if the birthday is *after* the `midPt`, then the `midPt` is thrown away, but if it is not after the midPt, the midPt must be kept.)

g) **Recognizing termination.** Any binary search ends when the interval has a size of 1, since that must be the item being sought. When after updating one of the end points (step f), they are the same, then the birthday has been found. At that point the text "Your birthday is " should replace the "Were you born after " text, and the day printed out.

h) **Explain how it works.** Finally, write a paragraph explaining, as to a friend, how the binary search works. Include discussion of the main ideas of the algorithm and how it has been converted into the specific program for finding a birthday given the Zodiac sign.

Grading Criteria

The Day Finder program will be graded according to the following criteria:

- The organization of the form revisions
- Operation of the application -- does it work and get the right answer
- Is the logic clearly specified
- Completeness -- do all aspects of the computation work
- Have control names been chosen thoughtfully
- Is the write-up clear and comprehensive

The project is **due 4 May 2001**. The electronic turn-in link will be found on the Projects page