

**The Heritage Computer Challenge
2002
Heritage High School
Newport News, Virginia**



Visual Basic

Instructions

The problems for this contest appear in order of difficulty. The maximum number of points you can earn for each problem is indicated under its title.

Problems are designed in the format used by The Great Computer Challenge, held annually each Spring at Old Dominion University. Some of these problems were actually used at the Great Computer Challenge in previous years.

The folder and project file names are provided to each project. You may name your form and module files as you wish. Just make sure that they are saved in the specified folder. Saving files in the wrong place will result in loss of points which may negatively effect your placement in this contest.

Above all, make sure you do NOT save any files on drive C. Doing so will result in a score of 0 (zero) for that project because your programs will be evaluated from a different computer than the one you are using now.

***** Problems *****

1. Quadrant Quest (10 points)
 2. Three Sailors and a Monkey (10 points)
 3. The Game of Nim (20 points)
 4. Dialer (30 points)
-

Quadrant Quest

(10 points)

All files go in - **k:\hcc\quadrant**
Project file name: **quadrant.vbp**

An angle of rotation (using degrees) in standard position in the coordinate plane may have any measure, positive, negative or zero.

Have the user input an angle between -360 and 360 degrees inclusive, integers only. Invalid values should be rejected.

Determine in which quadrant the terminal side of an angle with the given measure lies or, if the terminal side lies on an axis, which part of which axis (positive x axis, negative y axis, etc.).

Three Sailors and a Monkey

(10 points)

All files go in - **k:\hcc\sailors**

Project file name: **sailors.vbp**

Three sailors, shipwrecked with a monkey on a desert island, have gathered on one day a pile of coconuts that are to be divided early the next day. Sometime during the night, one sailor arises, divides the pile into three equal parts, and finds one coconut left over, which he gives to the monkey. He then hides his share, and returns the remaining coconuts to a single pile. Later during the same night, each of the other two sailors arises separately and repeats the performance of the first sailor. In the morning all three sailors arise, divide the pile into three equal shares, and find one coconut left over, which they give to the monkey.

Write a program in Visual Basic that will compute how many coconuts were in the original pile. Since there is more than one correct answer, the program should consider all coconut piles in the range of **1 to 1000**. The output should be displayed on the screen and consist of the following:

- a. The number of coconuts in the original pile.
- b. The number of coconuts after each sailor removes a third.

One correct answer is 79 and may be used to check the correctness of the program. Output for this pile could look like the following:

```
Coconuts in the original pile    79
Coconuts after the first sailor  52
Coconuts after the second sailor 34
Coconuts after the third sailor  22
```

Hint: the best solutions will display their output in a textbox with scrollbars so that the user can see all the solutions.

The Game of Nim

(20 points)

All files go in - **k:\hcc\nim**

Project file name: **nim.vbp**

The traditional game of Nim is played in the following manner:

There are three to five rows of stones arranged as shown (number of stones in each row may vary):

○	○	○								3
○										1
○	○	○	○	○	○	○	○			7
○	○	○	○	○						5

Let A and B be the two players. The players play according to the rules below:

- A. The game begins by first deciding which of two ways (called the "win option") a player will win:
 1. taking the last stone
 2. forcing the opponent to take the last stone (by leaving only one stone on the board)
- B. On any given turn only the objects from one row may be removed. There is no restriction on which row or on how many objects you may remove.
- C. You cannot skip a move or remove zero objects or remove more objects than there are remaining in the row.
- D. Opponents take turns removing objects until there are none left.

Create a program to play the game of Nim.

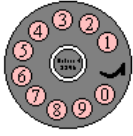
- It should begin by requesting the board set-up (the number of rows and the number of stones in each row) and the win option to be used.
 - At all times, the screen should display the win option and whose whose turn it is (Player A or Player B).
 - After each play, the game display should be updated accordingly (removing stones from the right, keeping them left-justified as shown).
 - And when someone wins, a message box should announce who won.
-

Dialer

(30 points)

All files go in - **k:\hcc\dialer**

Project file name: **dialer.vbp**



too tired from hunting dinosaurs all day.

Before the days of touchtone dialing, all telephones had rotary dials--a round dial with 10 holes in it. The holes were arranged in a circular pattern around the edge of the dial. Each hole had a number, 0 to 9, showing through. The hole with number 1 was located on the right, then moving counter-clockwise, the hole with a 2, then with a 3, and so on. The hole with 0 followed the hole with 9. To dial a number, the caller placed a finger into the hole with the desired number and spun the dial manually in a clock-wise direction until the finger was stopped by the small cradle located just underneath the hole with number 1. The process was slow, but most people didn't have too much trouble learning to use a rotary dial telephone, as long as they weren't

In this problem, you are to write a Visual Basic program that simulates the rotary dial of a rotary dial telephone, but you must do it in the following manner:

You must include a class module named Dialer. Each Dialer object will represent a single circular hole on a rotary dial telephone.

The Dialer class must maintain the following information: the container (use a variant type, although the container may be a form or a picture box), the coordinates of the center (in twips), the radius, the number that is displayed in the center, the forecolor and the fillcolor.

The Dialer class must contain the following methods (subroutines):

- an initializing subroutine that takes appropriate parameters so that, when called, will be able to set all maintained information.
- a subroutine named **Show**, with no parameters, that displays the Dialer object (note that the initializing subroutine does NOT display the object).
- A subroutine named **Shade**, also with no parameters, that displays the Dialer object, but always with a white forecolor and a black fillcolor.
- Additional subroutines may be added optionally.

The main form must set up 10 Dialer objects and display them in such as manner as to represent a rotary dial on a rotary dial telephone. Other graphics (or shape objects)--to complete the appearance of a rotary dial telephone--are optional. When the user types a phone number into a text box and then clicks a command button, each digit of the phone number is "dialed" as follows: beginning with the dialed number and progressing down to 1, the Dialer object containing that numbers turns white on black, then back to its original color, thus simulating the old fashioned act of dialing a rotary phone.
