DWITE Online Computer Programming Contest
November 2004

Problem 4

For Loops

For loops are frequently used in computer programming when the programmer knows the exact number of times a piece of code needs to be repeated.

For loops come in many different flavours, depending on the programming language used, but the outcome is identical. For example, the following for loops all find the sum of the first ten positive integers.

C / C++ / Java / Visual C++	**Delphi / Pascal**

`sum = 0;`
`for(int i=1; i<=10; i++) {`
` sum = sum + i;`
`}`
	`sum := 0;`
	`for i:=0 to 10 do`
	` sum := sum + i;`

PHP	**Turing**

`$sum = 0`
`for ($i = 1; $i <= 10; $i ++)`
`{`
` $sum = $sum + $i;`
`}`
	`sum = 0`
	`for i:1..10`
	` sum = sum + i`
	`end for`

Visual Basic	

`sum = 0`
`For i = 1 To 10`
` sum = sum + i`
`Next i`	

This particular problem deals with the Visual Basic for loop. You are to write a program that determines the outcome value of a particular for loop.

The input file (**DATA41.txt** for the first submission and **DATA42.txt** for the second submission) will contain five sets of data. Each set will contain four lines:

- the first line will be:
  `sum = value` where ***value*** is the initialized value for the variable ***sum***, an integer, -100<= ***value*** <= 100
- the second line will be the for loop heading in the form:
  `For i = start To finish`
  where ***start*** is the starting value of the loop counter, an integer, 0 <= ***start*** <= 10,
  finish is the ending value of the loop counter, an integer, ***start*** <= ***finish*** <= 100
- the third line will contain the formula that determines the value for the variable sum
  `sum = formula`
  where ***formula*** may contain spaces, integers, between -100 and 100, multilication (*), integer division (\), addition (+), subtraction (-), or the variables ***sum*** or ***i.*** The third line will never contain more than 255 characters.
- the fourth line will contain the statement: `Next i`

The output file (**OUT41.txt** for the first submission and **OUT42.txt** for the second submission) will contain the final value of ***sum*** after the for loop is completed. BEDMAS applies, but there will be no brackets (except around negative integers) or exponents, so watch out for order of operation. -32000 <= final value <= 32000

Sample Input (Only three sets given)

```
sum = 0
For i = 1 To 10
 sum = sum + i
Next i
sum = 0
For i = 1 To 10
 sum = sum + (-5)
Next i
sum = 2
For i = 2 To 4
 sum = sum * 2 + i \ 3 - i * 24 + sum \ 4
Next i
```

Sample Output

```
55
-50
-477
```