

The Great Computer Challenge

Programming: Any Language

Level IV

2006

High School Any Language (Non-business) Competition

Special Rules:

The judges of this competition may not be familiar with your computer or your operating system or your programming language or your programming language development environment. If for any of these reasons we cannot run your programs, you will not receive credit for them. Therefore, be sure to leave directions for us that explain:

1. How to start your computer.
2. How to login (including password) if necessary.
3. How to start up your development environment.
4. How to find your programs and load them into the development environment.
5. How to run your programs within the development environment.

You get the most points for a program that works. No special bonus is given for user friendliness. You may receive partial credit for a program that partially works or for code that shows an understanding of the problem and progress toward a solution.

1. Mini Sudoku

This problem is a simplified version of the popular sudoku game. In this version, the object is to create three rows of nine numbers each according to some rules. Each number is in the range 1 to 9. These rules govern the placement of the numbers:

1. Each of the nine digits must be used in each row. Therefore no digit is repeated in any row.
2. Each of the nine digits must be used in the first three columns (columns 1-3).
3. Each of the nine digits must be used in the second three columns (columns 4-6).
4. Each of the nine digits must be used in the third three columns (columns 7-9).

Write a program that produces a legitimate mini-sudoku layout, following the rules above, with the following addition. Your program will accept one row of input that contains 9 digits. These digits describe what must appear on the first row of the output layout.

1. If the digit is in the range 1-9, then that digit must be used in that position in the first row.
2. If the digit is a 0, any digit may be used in that position in the first row, subject to the rules given above.

Example:

input:

001002003

output:

471592863

923846157

586317429

2. Waiting for Xmas

Write a program that accepts a given month, day, and year in integer format and produces as output the number of days from the given day until the next Christmas day, Dec 25. Examples:

Input: 12 23 2005 Output: 2

Input: 12 25 2004 Output: 365

Input: 12 25 1999 Output: 366 (year 2000 is a leap year)

3. Number Personalities

Write a program that accepts an integer and produces as output:

1. whether the number entered is positive or negative or is zero;
2. if the number is not zero, whether it is even or odd;
3. if it is positive whether it is a prime number or not and if it is a prime, what the next highest prime number is;
4. if the number is not a prime, then the value of the immediate lower valued prime number.

Examples:

Input: 17

Output: positive, odd, prime, 19

Input: -17

Output: negative, odd

4. Independent Bubbles

Imagine a collection of spherical bubbles that exist in 3-d space. Each bubble may intersect zero or more of the other bubbles (if one bubble lies entirely within another bubble they are considered to intersect each other). Your program should accept a series of inputs that list the radius and the x, y, z coordinates of the centers of the spheres. An input giving a radius value of 0 will signal the end of the input sequence. Your code should compute and display a table that shows which spheres intersect and which spheres are isolated in space.

Example:

Input:

```
3 0 0 0
1 10 10 10
2 1 1 1
0 0 0 0
```

Output

```
0 1 2
0 - n y
1 n - n
2 y n -
```

5. Duckpin Bowling

Duckpin Bowling is a game played in 10 frames. At the beginning of the frame, 10 duckpins are arranged in the shape of a triangle at one end of a long narrow area called the alley. The bowler stands at the other end and rolls up to three balls down the alley with the object of knocking down all the pins. If she knocks them all down with the first ball it is called a strike. If she requires two balls before they are all knocked over, it is called a spare. If she scores neither a strike nor a spare it is called a wash.

Scoring:

1. If the frame is a STRIKE, the score is 10 plus the number of pins knocked over by the next ball (which would be in the next frame) plus the number of pins knocked over by the ball after that (which could be in the next frame again, or the frame after that).
2. If the frame is a SPARE, the score is 10 plus the number of pins knocked over by the next ball.
3. If the frame is a WASH, the score is the total number of pins knocked down by the three balls.
4. If the 10th frame is a STRIKE, the bowler bowls two additional balls to get the bonus for the strike. If it is a SPARE, she gets one additional ball to get the bonus for the spare.

Scoring representation in program input:

1. A strike is represented by an 'X'.
2. A spare is represented by a number between 0 and 9 (the pins knocked over by the first ball) followed by a forward slash '/'. For example '7/' means the bowler knocked over 7 with the first ball and 3 with the second.
3. A wash is represented by three digits between 0 and 9. For example '321' means the bowler got a total score of six for that frame.

Problem:

Your program should read one line of input that gives the scoring for 10 frames (plus an additional one or two balls if necessary) and tell the total score for that game.

Example:

Input:

321 X 9/ 901 7/ 8/ X X 342 X 9/

Output:

162

Explanation: The scores for the 10 frames are summed up as shown:

$(3+2+1)+(10+9+1)+(9+1+9)+$
 $(9+0+1)+(7+3+8)+(8+2+10)+$
 $(10+10+3)+(10+3+4)+(3+4+2)+$
 $(10+9+1)$