First Inaugural UAB

High School Programming Contest

May 7, 2005

Each problem in this packet contains a brief description, followed by two example executions of a successful implementation.

You will submit an answer by copying the source code of your solution onto a floppy disk, which you must then pass on to one of the volunteers in your assigned room. Your answer will be graded and a result form will be returned to you.

The example input and output shown for each question should be regarded only as a suggestion. We will test your program on the example provided, as well as two other sample input test cases

If your program fails, the result sheet will state the counterexample test case that caused your program to be judged incorrect. Each incorrect answer will be assessed a 20 minute penalty. As noted in the rules, the overall time will be considered as a secondary tie breaker (with total number of problems solved the initial ranking metric).

Please pay close attention to the directions in each problem description. In some cases, a clue may be given related to the expected limitations of your solution (e.g., Problem 1 only asks you to compute the factorial for the first 7 positive integers, but you may not use simple print statements to do this).

Problem 1:
Factorial Calculator

If n is an integer greater than 0, n factorial (n!) is the product: n* (n-1) * (n-2) * (n-3)... * 1. By convention, 0! = 1. You must write a program that allows a user to enter an integer between 1 and 7. Your program must then compute the factorial of the number entered by the user.

Your solution MUST actually perform a computation (i.e., you may not simply print “5040” to the screen as a literal value if the input is 7).

Example 1:

```
Enter a number: 4
4! = 24
```

Example 2:

```
Enter a number: 7
7! = 5040
```

Problem 2:
Searching 101

For this problem, you must allow the user to enter exactly 5 numbers as initial input. The program must then ask for an additional number. The output of the program will indicate whether the last number is contained in the first 5 numbers. Your output should resemble that provided in the example below. All of the numbers that will be entered are integers.

Example 1:

```
Enter a number: 1
Enter a number: 2
Enter a number: 5
Enter a number: 10
Enter a number: 15
Enter the number to be searched: 10
The number 10 appears in the first 5 numbers.
```

Example 2:

```
Enter a number: 10
Enter a number: 20
Enter a number: 30
Enter a number: 40
Enter a number: 50
Enter the number to be searched: 60
The number 60 does not appear in the first 5 numbers.
```

Problem 3:
Binary to Decimal Converter

A binary number is a sequence of bits (binary digits – 0’s and 1’s) of the form B_nB_{n-1}……B_1 B_0, where each B_i is a bit. The decimal equivalent is calculated by $B_n * 2^n + B_{n-1} * 2^{n-1} + ... + B_1 * 2 + B_0$. Write a program to input a binary number and output the decimal equivalent. The sample input will not have more than 8 individual bits (i.e., the largest value to be entered is 11111111).

Example 1:

```
Enter binary number: 101
In decimal: 5
```

Example 2:

```
Enter binary number: 11111
In decimal: 31
```

Problem 4:
Push Ups with Blaze

At UAB football games, Blaze does push ups after each Blazer score. After the first Blazer touchdown (and point after), Blaze does 7 push ups. After the second touchdown and point after, the score is now 14 and Blaze does 14 push ups.

Write a program that calculates how many total push ups Blaze does during the whole game. Assume that only 7 point touchdowns (including the point after) occur. Prompt for the final score and print out how many push ups Blaze has done.

Example 1:

```
Enter final score: 21
Push ups: 42
```

Example 2:

```
Enter final score: 7
Push ups: 7
```

Problem 5:
Valid Identifier Checker

An identifier in Java, C++ and most other programming languages must begin with a letter and then may be followed by any number of letters or digits. It is possible that underscores (_) will also appear, but only in the middle and never two consecutively. Write a program to read a string and output whether it is a valid or invalid identifier. Each string will be 10 characters or less in size.

Example 1:

```
Enter id: UAB_HSPC
Answer: UAB_HSPC is a valid identifier
```

Example 2:

```
Enter id: a_b_c__2
Answer: a_b_c__2 is not a valid identifier
```

Problem 6:
Shortest Path in Alabama

Write a program to find the shortest routing and distance between two Alabama cities using the following distance table. You are not allowed to use any other manually computed distances in your program.

Alabaster-Birmingham	24 miles
Alabaster-Montgomery	71 miles
Birmingham-Huntsville	103 miles
Birmingham-Tuscaloosa	59 miles
Demopolis-Mobile	141 miles
Demopolis-Montgomery	101 miles
Demopolis-Tuscaloosa	65 miles
Mobile-Montgomery	169 miles
Montgomery-Tuscaloosa	134 miles

Example 1:

```
Enter city #1: Demopolis
Enter city #2: Birmingham
Shortest routing and distance:
Demopolis-Tuscaloosa-Birmingham, 124 miles
```

Example 2:

```
Enter city #1: Mobile
Enter city #2: Huntsville
Shortest routing and distance:
Mobile-Montgomery-Alabaster-Birmingham-Huntsville, 367 miles
```