

2004 State Convention Computer Programming Test Programming Section

This section of the computer programming test will consist of ten problems. Each of these problems will be solved by writing a computer program to process a specified input and respond with the correct output for any case provided. The competition will last an hour and a half, operating as follows:

- When you have completed a program, raise your hand to signal a judge.
- Inform the judge which program you have ready for testing.
- The judge will read test input that you must enter into your program.
- If the output is correct for all input read by the judge and displayed in the proper format, you will receive 10 points.
- Otherwise, the judge will inform you that either your output was incorrect or your output was in the incorrect format and then initial in the penalty box for that question. You may submit the program once and only once more for testing.
- If the output is correct on the second attempt you will receive 6 points.

The problem will specify the input and output conditions for each program. The problem will also provide an example to help clarify any confusion as to what is being asked. Additional information may be inherent in the sample data, but the sample data **may not** test all possibilities. You may be asked to list the program for the judge, so be prepared to justify what you have written. It is suggested that you save your programs in that event that an alternate solution is approved. Programmers may assume that the judge will provide valid data, so error checking may be kept to a minimum.

- 1) No calculators are permitted.
- 2) Each school is limited to one team of three members and one computer.
- 3) Each school must bring its own computer, extension cord, and power strip.
- 4) Only one member may be using a keyboard at any given time.
- 5) With the exception of standard libraries and the AP classes, no school may use code written before the competition.
- 6) Programs may be written in C, C++, Java, Visual Basic, BASIC, or Pascal.
- 7) Input format may vary at your discretion, but each value must be entered in the same order provided by the judge.
- 8) Output format must match the format of the sample output provided.
- 9) All output must be accurate to at least the ten thousandth's place.
- 10) The runtime of any program may not exceed five minutes.
- 11) No programs will be judged more than twice.
- 12) During the final minutes, judges will score only one program at a time.
- 13) Teams must stop **immediately** when time is called. A team may have **one** program in memory to be judged.
- 14) In **all** cases, disputes will be settled by the judges at the competition.
- 15) To receive credit, the answer sheet must be filled out completely. Do not forget to enter your team's FAMAT number and the name of the school and team members.
- 16) All solutions have been written and checked in C++.

1. Write a program to convert a positive decimal integer to an integer of any arbitrary base. The base of the output will be restricted to the range 2 through 35, inclusive. Use the characters 'A' through 'Z' to represent digits greater than 9.

Example:

```
Integer (base 10): 75
Base: 19
3I
```

2. Write a program to convert a positive integer of any arbitrary base to a decimal integer. Again, the base of the input will be restricted to the range 2 through 35, inclusive and the characters 'A' through 'Z' will be used to represent digits greater than 9.

Example:

```
Integer: 3I
Base: 19
75
```

3. Write a program to compute the prime factorization of a positive integer. Note that unreasonably large integers will not be given as input as the upper limit will be $2^{16} - 1$. The output can either list all the factors or list repeated factors with exponents; however, the factors must be listed in *ascending* order. The example lists both options.

Example:

```
Input: 45
3*3*5          or          3^2*5
```

4. Write a program to make calculations on two vectors of the form $U = A\mathbf{i} + B\mathbf{j} + C\mathbf{k}$ and $V = D\mathbf{i} + E\mathbf{j} + F\mathbf{k}$, where $A, B, C, D, E, F \in \mathfrak{R}$ and $\mathbf{i} = (1, 0, 0)$, $\mathbf{j} = (0, 1, 0)$, $\mathbf{k} = (0, 0, 1)$. The input will be the coefficients describing the vectors. The output should be $(U \bullet V) \cdot [(U - V) \times (V)]$, where \bullet denotes the inner (or dot) product and \times denotes the cross product.

Example:

```
A=1
B=1
C=1
D=0
E=1
F=1
(0, -2, 2)
```

5. Write a program to make calculations on complex numbers in the following form $X = A+Bi$ and $Y = C+Di$, where $i = \sqrt{-1}$ and $A, B, C, D \in \mathfrak{R}$. The input will be the coefficients describing the complex numbers and the output should be $|X - Y| \cdot (X^2 + Y^2)$.

Example:

```
A=1
B=1
C=0
D=1
-1+2i
```

6. Write a program to parse a string into tokens. A token is defined as a continuous string of alphanumeric characters or a string surrounded by quotation marks. Any other characters should be disregarded, but can still be used to separate tokens.

Example:

```
Input: "fruit flies",bananas,moldy bread
Token 1: fruit flies
Token 2: bananas
Token 3: moldy
Token 4: bread
```

7. Write a program to compute the derivative of a polynomial of x in the form $f(x) = a_1 \cdot x^{b_1} + a_2 \cdot x^{b_2} + \dots + a_n \cdot x^{b_n}$, where $a_i, b_i \in \mathfrak{R}$ for all i between 1 and n . The inputs will be n , a_i and b_i for each i between 1 and n , and a value for x (called x_k). The output should be $\frac{d}{dx} f(x) | x = x_k$.

Example:

```
n=2
a1=1
b1=1
a2=2
b2=2
xk=1
5
```

8. Write a program to compute statistical information about a set of numbers. The input will be the number of elements in the set followed by each element. The output should be the probability that the first element would be selected at random from the set, the arithmetic mean of the set, and the median of the set (in that order).

Example:

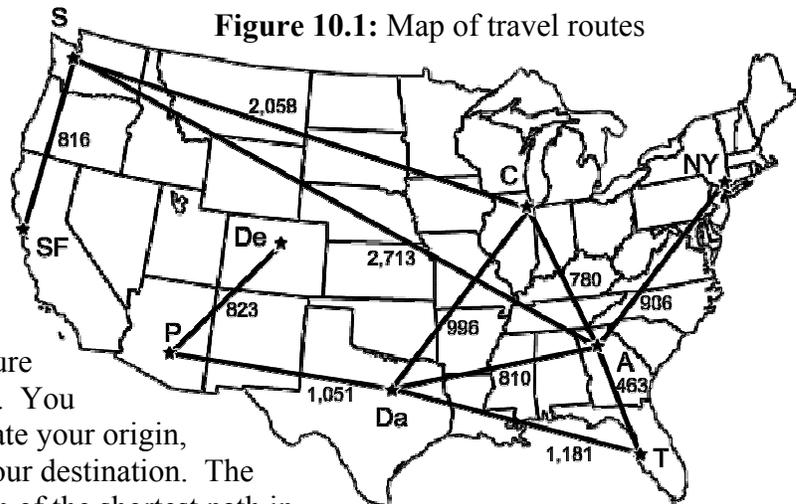
```
Number of elements: 3
1
2
3
0.33333, 2, 2
```

9. Write a program to compute information about a right cylinder. The input will be the radius and height of the cylinder. The output should be the total surface area and the difference between the volume of the cylinder and the volume of a right rectangular prism inscribed into the cylinder, in that order. Use $\pi=3.14159$ in your program.

Example:

```
r=1
h=1
12.56636, 1.14159
```

10. You like to take road trips all over the country. You are planning one this summer between two of the cities listed on the map and you only want travel through the cities listed. You have to write a program to determine the shortest possible drive, based on the information in figure 10.1 (all distances are in miles). You will be given a string to designate your origin, followed by one to designate your destination. The output should be only the length of the shortest path in miles. Note that cities are indicated by a star and a character label. Any intersections without a star should not be considered as stopping points in your path.



Example:

```
Origin: A
Destination: T
463 mi
```