

# **This packet sample is provided courtesy of A+ Computer Science.**

**This is a partial packet with 4 problems.  
A complete packet contains 12 problems.**

**[www.apluscompsci.com](http://www.apluscompsci.com)**

# 2010 A+ Computer Science Sample Packet

## Computer Science Competition Hands-On Programming Set

### I. General Notes

1. Do the problems in any order you like. You do not have to solve in 1 to 12 order.
2. All problems have a value of 60 points.
3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.
4. Your program should not print extraneous output. Follow the form exactly as given in the problem.
5. A penalty of 5 points will be assessed each time that an incorrect solution is submitted. This penalty will only be assessed if a solution is ultimately judged as correct.

### II. Point Values and Names of Problems

Number	Name	Point Value
Problem 1	A Greeting in the Dark	60
Problem 2	Objection	60
Problem 3	Submissions	60
Problem 4	A Prime Opportunity	60
Problem 5	Power Squared	60
Problem 6	Roman Homework	60
Problem 7	The Ending Sounds Like the Beginning	60
Problem 8	Possibility of Donuts	60
Problem 9	Homework Time	60
Problem 10	Balloono	60
Problem 11	Wikiracing!	60
Problem 12	Superstitious Thirteen	60
		<b>Total 720</b>

**Problem #1**  
**60 Points**

**A Greeting in the Dark**

**Program Name:** greeting.java

**Input File :** none

“Hello, world!” is one of the most common programs created by programmers. A new rendition has been created, one ought to try recreating it.

**Input**

There is no input for this problem.

**Output**

Print out “Oh hai world!” in the format shown below  
(The distance between the words is 6 spaces)

**Example Input File**

none

**Example Output To Screen**

OO	H	H		H	H	AAAA	III	W	W	OO	RRRR	L	DDDD	!!						
O	O	H	H		H	H	A	A	I	W	W	O	O	R	R	L	D	D	!!	
O		O	HHHH		HHHH	AAAA	I	W	W	W	O		O	RRRR	L		D	D	!!	
O	O	H	H		H	H	A	A	I	W	W	W	O	O	R	R	L		D	D
OO		H	H		H	H	A	A	III	WWWWW		OO		R	R	LLLL	DDDD		!!	

## Problem #2

### 60 Points

### Objection

**Program Name:** objection.java

**Input File:** objection.in

Once again, a programming team is arguing the correctness of their programming solution. In order to bring rain down on the judges, they will cry “Objection!” at various points until the judges finally cry out “HOLD IT!” at their futile attempts. The judges can only take so many objections. However, if the Objections they can stand is less than or equal to 0, it means that the team was actually right! Thus a cry of “GUILTY!”, will be cried out since the judges falsely declared the problem incorrect.

#### Input

The first number will tell you the number of times the program will run. Afterwards, each number will tell you how many times the team was able to cry “Objection” before a “HOLD IT!” was declared, or if the team screamed “GUILTY!” when they discovered their problem was correct. The number will always be an integer.

#### Output

The first integer will tell you the number of trials. From then on, the number will tell the number of “Objection!”’s that will be shouted before “HOLD IT!” is finally shouted. If the number is less than or equal to 0, a “GUILTY!” is shouted instead. A blank line will separate all trials.

#### Example Input File

```
3
1
5
-3
```

#### Example Output To Screen

```
Objection!
HOLD IT!
```

```
Objection!
Objection!
Objection!
Objection!
Objection!
HOLD IT!
```

```
GUILTY!
```

**Problem #3**  
**60 Points**

**Submissions**

**Program Name:** submit.java  
**Input File:** submit.in

You are an administrator for MLIA. Your goal is to write a program to check for the correct formatting of submitted entries. A correctly formatted entry should begin with the word "Today", and the last word should be "MLIA"

**Input**

First will be the number indicating the number of trials. After that, each line will contain a separate submission that needs to be verified.

**Output**

If the submission starts with "Today" and ends with "MLIA" (case insensitive) the program should print out "VALID ENTRY"

Otherwise, if the submission fails to meet that requirement, the program should print out "INCORRECT FORMATTING, TRY ANOTHER SUBMISSION"

**Example Input File**

```
5
ToDAY, I went to school.  mlia
Hehehe today mlia this shouldn't work
Today, I went to a programming contest.  Hehe.  MLIA
TODAYMLIA
T0day is a brand new day! MLIA
```

**Example Output To Screen**

```
VALID ENTRY
INCORRECT FORMATTING, TRY ANOTHER SUBMISSION
VALID ENTRY
VALID ENTRY
INCORRECT FORMATTING, TRY ANOTHER SUBMISSION
```

**Problem #4**  
**60 Points**

**A Prime Opportunity**

**Program Name:** prime.java

**Input File:** prime.in

Lay Z. Bee, like many students before him has decided his math homework is taking too long to do by hand, and would rather devise a program to do it for him. His homework is to do the prime factorization of a given integer.

**Input**

The first number will be the number of problems Lay Z. Bee must do. After that, each number is an integer for which he must find all the prime factors for.

**Output**

Print out all the prime factors for each number on its own line. If a prime is factored twice out of a number, it will be printed twice. (Example: 4 is 2 2, 16 is 2 2 2 2). The factors should be in printed in ascending order.

**Example Input File**

```
5
7
84
121
1000
96
```

**Example Output To Screen**

```
7
2 2 3 7
11 11
2 2 2 5 5 5
2 2 2 2 2 3
```

# **This packet sample is provided courtesy of A+ Computer Science.**

**This is a partial packet with 4 problems.  
A complete packet contains 12 problems.**

**[www.apluscompsci.com](http://www.apluscompsci.com)**