

# Taylor High School Computer Science Packet

## I. General Notes

1. Do the problems in any order you like. They do not have to be done in order from 1 to 12.
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.

Good luck!

# Welcome to the Taylor High School Halloween Programming Contest

Check in times and set up begin at arrival.

Welcome and Introduction will be at 9:00.

Programming begins at 9:15 and runs until 12:15.

Pizzas will be available for lunch and should be delivered to your table by 11:30.

The pizzas will cost \$13 and you will have a choice of pepperoni or cheese.

Attached to this package is a pizza order form, please give the completed form to the concession host before 9:00.

If you wish to appeal any of the problem decisions at the end of the contest, please do so as quickly as possible after the contest.

Awards will be presented after the appeals are completed and the results are tallied.

We will be using the PC2 server for contest problem submissions. Accessing the server will be through your web browser. A Taylor host will assist each team with the login information. The server may be accessed using the URL:

HTTP://192.168.1.45:8080.

All problems in this contest that need input data will read the input from the standard system input.

In Java this can be done using:      `Scanner input = new Scanner(System.in);`

In python this can done using:      `data = input("")`

## Registration and Dry Run

All teams will need to register by writing a program that lists each team member's name. This will provide us with your names as well as ensuring your team can connect and submit programs to the PC2 server.

This registration/dryrun problem will need to read one line of text from the standard system input. Then the program will print the line of text followed by the following lines:

- Team Number:
- Advanced or Novice
- Member 1 (please include first and last names)
- Member 2
- Member 2

An exact match is not required as long as all the information is provided. When the program is complete, connect to the server through your browser and log in using your team's information. On the screen click the Submit Problem button on the top left of the screen. Then select the problem DryRun, choose your language (Java or Python), and submit the problem. If the submission is successful, you will get a Yes response from the judges otherwise it will be an indication of a problem.

Please ask for assistance if needed.

# Appeal Form

(Please submit as soon as possible after the contest ends)

Team Number: 21

Problem Name: Test Parity

Reason for appeal: Wrong answer, but correct for test input.

# Appeal Form

(Please submit as soon as possible after the contest ends)

Team Number:

Problem Name:

Reason for appeal:

Number	Name
Problem 1	Almost Prime
Problem 2	KoolAid
Problem 3	Foreign Shoes
Problem 4	Taxation
Problem 5	Crossword Clues
Problem 6	Right Stuff
Problem 7	Periodic Permutations
Problem 8	Deer Tags
Problem 9	Vowely Words
Problem 10	Tiles
Problem 11	Bubble Pop 1
Problem 12	Bubble Pop 2
Problem 13	Part Of Speech
Problem 14	Touchy
Problem 15	Almost
Problem 16	Integral
Problem 17	Major MalFunction
Problem 18	Taxing
Problem 19	Speed
Problem 20	Resize
Problem 21	Puddle Bop
Problem 22	Baseball Bin Berry Berry Good To Me
Problem 23	Box J
Problem 24	Day To Day
Problem 25	Flex J Box
Problem 26	Gender Race
Problem 27	Snapshot Day
Problem 28	Taxi
Problem 29	Test Parity
Problem 30	Weird Change

## 1. Almost Prime

Program Name: AlmostPrime.java    Input File: almostprime.dat

Prime numbers have only two factors: one and itself. There are some numbers that have only three factors: one, another number and itself. For example,

the factors of 4 are 1, 2, 4 (1<sup>st</sup> number is 4)

the factors of 9 are 1, 3, 9 (2<sup>nd</sup> number is 9)

the factors of 25 are 1, 5, 25 (3<sup>rd</sup> number is 25)

Find the first 40 numbers that are “almost prime!”

**Input:** There will be an unknown number of lines in the data file. Each line contains a single integer, N. The largest N will be 40.

**Output:** Print out the Nth number that is “almost prime.”

### **Example Input File**

1

2

3

7

### **Output to screen:**

4

9

25

289

## Java 17 Review

Java 17 is the latest version of Java, released in September 2022.

Java 17 includes several new features and improvements, such as the Stream API, which makes it easier to work with data streams.

Java 17 also includes the Java Platform Modularity System (JPMS), which allows for better modularity and easier maintenance of large codebases.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

Java 17 is a major update to the Java platform, and it is recommended for anyone who wants to take advantage of the latest features and improvements.

Java 17 is available for download from the Oracle website.

## 2. KoolAid

Program Name: KoolAid.java

Input File: koolaid.dat

In the old days (when I was a kid), KoolAid packets made only enough for a 2 quart pitcher, and you had to add 1 cup sugar. Now they come in packets that you add to water bottles (in liters). You can make a 0.5 liter (small water bottle) or a 1.0 liter (medium water bottle) and a 2.0 liter (large water bottle) also. 2 people can share a 1.0 liter or 4 people could share a 2.0 liter bottle, but 5 people would need a 2.0 L and a 0.5 L bottle mixed. Given an amount of people, find the MINIMUM amount of KoolAid to make with NO left-over KoolAid. For example, you could make a 2.0 liter for 1 person, but a 0.5 liter bottle would be less wasteful.

Each person will drink 0.5 liters of Kool-Aid. Make enough for N people using large (2.0 L), medium (1.0 L) and small (0.5 L) bottles. Make only enough for N people with no wasted Kool-Aid.

**Input:** The first line consists of the number of data elements in the file, followed by that number of lines. Each subsequent line contains one integer, N, the number of people.

**Output:** Print the number and type of water bottle, starting with “large”, then “medium”, then “small”.

### **Example Input File**

```
6
1
2
3
4
20
99
```

### **Output to screen:**

```
1 small
1 medium
1 medium 1 small
1 large
5 large
24 large 1 medium 1 small
```

1. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

2. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

3. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

4. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

5. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

6. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

7. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

8. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

9. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

10. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

### 3. Foreign Shoes

Program Name: ForeignShoes.java

Input File: foreignshoes.dat

Shoe sizes in the US and the UK are close, but you want your shoes to fit well! If you are in the UK, find the US shoe size and vice-versa. Here are the conversions:

Women: UK 8 == US 10 (add 2 for US)

Men: UK 9 == US 10 (add 1 for US)

**Input:** The first line consists of the number of data sets in the file. Each data set will consist of 2 strings and an integer: the gender, the country, and the size (separated by a single space).

**Output:** Print out the gender, the country and the converted size (separated by a single space).

#### **Example Input File**

3

Women UK 7

Men US 10

Men UK 10

#### **Output to screen:**

Women US 9

Men UK 9

Men US 11

1. In the following code, what will be printed to the screen?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

## 4. Taxation

Program Name: Taxation.java

Input File: taxation.dat

When you check out at the grocery store, the receipt shows whether or not the item was taxable. Most grocery items are not taxed, but some supplies and “junkfood” items are taxed. If it is taxable, they add the tax only to those items. Given a receipt (indicating if it is taxable), calculate the total cost on the bill. Assume a tax rate of 8.25% on the taxable items, and do not round until the last sum.

**Input:** The first line consists of the number of items purchased. Each subsequent line contains the cost of the item, with a “T” at the beginning (followed by a space) if it is taxable.

**Output:** Show “The total is \$\_\_\_\_.”

### **Example Input File**

```
10
2.99
T 3.99
25.20
T 25.00
19.00
T 19.99
5.00
6.00
T 7.00
T 8.00
```

### **Output to screen:**

```
The total is $127.45
```

1. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to two decimal places.

2. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to three decimal places.

3. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to one decimal place.

4. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to four decimal places.

5. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to five decimal places.

6. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to six decimal places.

7. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to seven decimal places.

8. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to eight decimal places.

9. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to nine decimal places.

10. Write a C++ program that asks the user for a number between 1 and 100. If the user enters a value less than 1 or greater than 100, then the program should output an error message and ask the user to enter a new value. The program should then output the square root of the user's input. The output should be displayed to ten decimal places.

## 5. Crossword Clues

Program Name: CrosswordClues.java

Input File: crosswordclues.dat

You are writing crossword puzzles and you have started a new puzzle with some of the words filled in. Now, you are trying to find words that complete the partial letters going down or across. Given a partial word, see if it matches a word in your word bank. There may be more than one match, so print out all possible matches. Blank letters in the puzzle will be represented by an asterisk (\*).

For example, you have a word with 3 letters starting with ‘m’. Here are some matches for a three-letter word starting with m:

m\*\*: “mom”, “man”, “moo”, “met”, “mat”, “men”.....

In this program, the longest word in the word bank will be 7 letters.

**Input:** The first 10 lines consist of 10 words each (the 100 words in the word bank—the judge data file will contain the same words). The rest of the file contains an undetermined number of partial words, each on one line.

**Output:** For each partial word, print out all the words that match (in the order they appear in the word bank) on one line. If there are no matching words, print out “NO MATCH.”

### Example Input file

```
as at aft and ant apps amble ample apple applet  
be bean beat bend bump bunt bust butte battle before  
ebb end eon east ends ever eves either esters eastern  
fen fro fun fend fern from font fuzz fonts front  
gin gun gut gins guns gust guts gusty gutsy goner  
hat hit hot hut hats hits host huts horses hotter  
pen pin pun port post pots punt porter potter punter  
sap sip sop saps sips sops soot sort scoot skirt  
tap tip top taps tips tops toot tort trips troop  
zap zen zip zit zaps zips zits zebra zebras zipper  
a*  
z*  
z**  
a**le  
f*n*  
***t
```

### Output to screen:

```
as at  
NO MATCH  
zap zen zip zit  
amble ample apple  
fend font  
beat bunt bust east font gust host port post punt soot sort toot tort
```



## 6. Right Stuff

Program Name: RightStuff.java

Input File: rightstuff.dat

In science class, you want to get the right answer. Scientists measure accuracy (how close you are to the accepted value) and precision (how close all your data values are together). You want to get the “right stuff.”

In this program, take a data set and calculate the accuracy and precision. For the accuracy, compare the average to the true answer. The percent error needs to be less than or equal to 5% (plus or minus). For the precision, compare the range (high – low) with the average. The range must be less than or equal to 10% of the average of all values.

**Input:** The first line consists of the number of data sets in the file. Each subsequent line will contain a series of N double type variables. The first number is the accepted value, and the other (N-1) values make up the experimental data. N will be at least 5 and at most 50.

**Output:** Print out “Accurate”, “Precise”, “Both”, or “Neither.”

### **Example Input file**

```
3
2.75 2.68 2.70 2.71 2.75 2.75 2.76
3.14 2.14 4.14 2.14 4.14 2.14 4.14 2.14 4.14
9.99 6.99 7.01 7.11 6.98 7.00 7.05 7.09
```

### **Output to screen:**

```
Both
Accurate
Precise
```

1. **What is the output of the following code?**

```
for i = 1 to 5 do
    print(i)
```

2. **What is the output of the following code?**

```
for i = 1 to 5 do
    print(i)
    for j = 1 to 3 do
        print(j)
```

3. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    print(i)
```

4. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    print(i)
```

5. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    print(i)
```

6. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    for m = 1 to 5 do
        print(m)
    print(i)
```

7. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    for m = 1 to 5 do
        print(m)
    for n = 1 to 6 do
        print(n)
    print(i)
```

8. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    for m = 1 to 5 do
        print(m)
    for n = 1 to 6 do
        print(n)
    for o = 1 to 7 do
        print(o)
    print(i)
```

9. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    for m = 1 to 5 do
        print(m)
    for n = 1 to 6 do
        print(n)
    for o = 1 to 7 do
        print(o)
    for p = 1 to 8 do
        print(p)
    print(i)
```

10. **What is the output of the following code?**

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 4 do
        print(l)
    for m = 1 to 5 do
        print(m)
    for n = 1 to 6 do
        print(n)
    for o = 1 to 7 do
        print(o)
    for p = 1 to 8 do
        print(p)
    for q = 1 to 9 do
        print(q)
    print(i)
```

## 7. Periodic Permutations

Program Name: Periodic2.java

Input File: periodic2.dat

You can make the word “chocolate” from the symbols of elements on the periodic table:  
“C”, “H”, “O”, “C”, “O”, “La”, “Te”  
using the symbols Carbon, Hydrogen, Oxygen, Lanthanum and Tellurium. You can also use Cobalt for the “Co” before “La”.

I also saw a UIL Science team t-shirt using the word “champions” from Carbon, Hydrogen, Americium, Phosphorus, Iodine, Oxygen, Nitrogen and Sulfur:  
“C”, “H”, “Am”, “P”, “I”, “O”, “N”, “S”

The word “banana” or “bananas” can be made with “Ba”, “Na”, “Na” and “S”, but not with “N” because there is no “A” or “An” on the periodic table. Trying to use Nitrogen “N” would appear to fail when checking the rest of the word, but using Sodium “Na” would work for the rest. So you need to try any combination of 1 and 2 letter element symbols

In this program, determine if a lowercase word can be made with any combination of 1 or 2 letter element symbols.

**Input:** The first 4 lines will each contain 25 atomic symbols. The 5<sup>th</sup> line contains an integer N, representing the number of words to check. The next N lines each contains a single word (no spaces or special characters) made of lowercase letters that is no longer than 30 characters. The first four lines of data will be the same in the judge data file.

**Output:** Print “yes” or “no” for each word indicating if it can be made from the element symbols.

### Example Input file

```
H He Li Be B C N O F Ne Na Mg Al Si P S Cl Ar K Ca Sc Ti V Cr Mn  
Fe Co Ni Cu Zn Ga Ge As Se Br Kr Rb Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn  
Sb Te I Xe Cs Ba La Ce Pr Nd Pm Sm Eu Gd Tb Dy Ho Er Tm Yb Lu Hf Ta W Re  
Os Ir Pt Au Hg Tl Pb Bi Po At Rn Fr Ra Ac Th Pa U Np Pu Am Cm Bk Cf Es Fm  
9  
chocolate  
rose  
bananas  
champions  
north  
south  
uil  
ladygaga  
xenophobe
```

(Continued on next page...)

**(Problem 7 contin.)**

**Output to screen:**

yes  
no  
yes  
yes  
no  
yes  
no  
yes  
yes

## 8. Deer Tags

Program Name: DeerTags.java

Input File: deertags.dat

You work for a wildlife biologist who is tracking deer with radio transmitter tags. The deer can be tracked over time to see if or how long they can survive. If a radio tag is moving, it is assumed the deer is still alive. If the radio tag is stationary for a day, the body is searched for. If a tagged carcass is found, the cause of death could be from natural causes or by a predator. Given the radio data, write a program tabulating the statistics for the deer population observed.

Each time a deer is observed, the radio id and a status report will be logged. If a live deer is found, it can be tranquilized and tagged. If a dead deer with a tag is found, the type of death will be logged.

In this program, each line of data will contain a 4 character string followed by a status report. The optional status reports are as follows:

- OK
- NEW
- DEAD – COYOTE
- DEAD – MOUNTAIN LION
- DEAD – BEAR
- DEAD – NATURAL CAUSES

Given a database of radio tag information, print a report with statistics of how many deer are still alive or dead (ranked by cause of death from highest to lowest). If there is a missing category in a data set, show “0%” after the category.

**Input:** There are an unknown number of data sets in the file. Each data set consists of a radio id (4 character string) followed by a status report (from the above list).

**Output:** Print out statistics (nearest percent) for the number of deer still alive. Then rank the cause of death from highest to lowest, showing the percent of each type. If there is a missing category, show “0 %”.

**(Continued on next page...)**

**(Problem 8 contin.)**

**Example Input file**

```
A001 NEW
A002 NEW
A001 OK
A003 NEW
A003 DEAD - NATURAL CAUSES
A004 NEW
A002 OK
A001 OK
A005 NEW
A006 NEW
A007 NEW
A008 NEW
A009 NEW
A010 NEW
A011 NEW
A012 NEW
A013 NEW
A014 NEW
A015 NEW
A014 OK
A014 OK
A014 OK
A002 DEAD - NATURAL CAUSES
A015 DEAD - COYOTE
A014 DEAD - BEAR
A013 DEAD - BEAR
A012 DEAD - BEAR
A011 DEAD - MOUNTAIN LION
A010 DEAD - MOUNTAIN LION
A009 DEAD - MOUNTAIN LION
A008 DEAD - MOUNTAIN LION
A007 DEAD - MOUNTAIN LION
```

**Output to screen:**

```
ALIVE 27%
MOUNTAIN LION 33%
BEAR 20%
NATURAL CAUSES 13%
COYOTE 7%
```

## 9. Vowely Words

Program Name: Vowely.java      Input File: vowel.dat

Some words have a lot of vowels (aeiou) compared to consonants. Given a word, determine if it is “vowely” or not. A word is “vowely” if at least half of its letters are vowels.

**Input:** The first line will indicate the number of words. Each line will contain one word (no spaces).

**Output:** Print out either “YES” or “NO”.

### **Example Input file**

```
6
banana
mystic
baaxuwaashee
regular
complex
regulate
```

### **Output to screen:**

```
YES
NO
YES
NO
NO
YES
```



## 10. Tiles

Program Name: Tiles.java

Input File: tiles.dat

You work for a company that lays floor tile. You run the website and want to put a “tile calculator” on the site. The customer will enter the length and width (in feet) of the rectangular area to be tiled, and you will tell them how many tiles (1 square foot each) they will need. When the tile layers talk to you, they tell you that there are some edges they have to cut -- so you will need to allow for 10% extra tiles for cutting.

If you calculate a fractional number of tiles, then make a whole number by using the next highest integer. For example, if you have a 14 by 16 foot room, the area is 224. Using 10% more, it would be 246.4 tiles: use 247 tiles.

**Input:** The first line will indicate the number of data sets. Each data set will consist of two integers, the length and width.

**Output:** Print out the number of tiles (an integer).

### **Example Input file**

```
3
14 16
10 20
11 21
```

### **Output to screen:**

```
247
220
255
```

After you have completed the first few steps, you will notice that the code is becoming longer and more complex. This is because each step adds another layer of complexity to the program. For example, if you want to add a new feature to the program, you will need to change the existing code to accommodate the new feature. This can be time-consuming and frustrating, especially if you are not experienced in programming.

One way to make the process easier is to break down the program into smaller, more manageable pieces. This can help you focus on one task at a time and avoid getting overwhelmed by the complexity of the entire program.

Another way to make the process easier is to use a debugger. A debugger is a tool that helps you identify and fix errors in your code. It can help you find bugs and errors that you might otherwise miss.

Finally, it is important to remember that programming is a skill that takes time and practice to master. Don't be afraid to make mistakes or to ask for help. With persistence and dedication, you can become a proficient programmer.

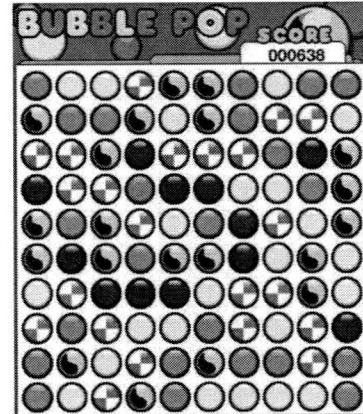
## 11. Bubble Pop 1

Program Name: BubblePop1.java

Input File: bubblepop1.dat

I have a couple of games on my Kindle, and Bubble Pop is one of my favorites. When you select contiguous bubbles (3 or more connected vertically or horizontally), they pop and the bubbles above them fall down in their place. You get a higher score for popping bubbles that are larger. You keep popping bubbles until the largest contiguous bubble size is 2. There are 5 different “colors” of bubbles.

You are writing the code for this game! It is a complex program, so you are writing only the first part. Given an array index of the matrix, determine if the bubble can burst and the size of the bubble if it can. For this program, you will not be asked to burst the same bubble more than once (so you can alter the matrix and it will not affect the results).



**Input:** The first 10 lines consist of the game board, each line containing 10 capital letters (A-E). The next line contains the number of data sets. Each data set contains two integers, the row and column of the bubble you are trying to pop.

**Output:** Print out either  
“YES X” where X represents how large the bubble is, or  
“NO” if there are not 3 contiguous characters.

### Example Input file

```
ABBCDDABAA
DAADBDACCB
CCDECCCCAED
ECCAEEBBAD
DADCBAECBD
DEDADDEBDB
BCEEEBCCDB
CACBBACBCB
ADBCADAACB
ABCDABBBBB
5
0 0
2 0
0 9
6 4
9 9
```

(Continued on next page...)

**(Problem 11 contin.)**

**Output to screen:**

NO  
YES 4  
NO  
YES 3  
YES 9

## 12. Bubble Pop 2

Program Name: BubblePop2.java

Input File: bubblepop2.dat

This is a second program about the Bubble Pop game. In the real game, there is more complexity. When a bubble is burst, the cells from above fall down into the void created, and a random bubble is added to the top. For this program, always put the letter X at the top. Therefore, do not check to see if the letter X is contiguous to burst the bubble. Given a series of moves, determine if the game continues or if the game is over (when there is a maximum of 2 connected cells on the game board). If the game is over, assume the last move will cause this.

Note, that when one bubble is popped, the game board is altered. This may cause a new bubble to be created that can be popped. This is how you gain points in the game: keep popping new bubbles!

For example, when the ‘C’ at row 2 column 4 is selected, the three ‘C’s are deleted and the above cells fall down, leaving 3 ‘X’s on the top row. Then, the ‘D’ at row 1 column 3 could be a bubble with size 4:

ABBCDDABAA	ABBCXXXBAA	ABBXXXXBAA
DAADBDACCB	DAADDACC <del>B</del>	DAACXXACCB
CCDE <u>CCC</u> AED	CCDEBDAE <del>D</del>	CCDEBXAAE <del>D</del>
ECCAEEBBAD	ECCAEEBBAD	ECCAEEBBAD
DADCBAECBD	---> DADCBAECBD	---> DADCBAECBD
DEDADDEBDB	DEDADDEBDB	DEDADDEBDB
BCEEEBCCDB	BCEEEBCCDB	BCEEEBCCDB
CACBBACBCB	CACBBACBCB	CACBBACBCB
ADBCADAACB	ADBCADAACB	ADBCADAACB
ABCDABBBBB	ABCDABBBBB	ABCDABBBBB

In this program, every selection will pop a bubble. The last selection could cause the game to end. If the maximum bubble size across the entire board is at most 2, then the game is lost.

(I have included an applet version of this game, so ask the contest director for the code AFTER the contest. It's not finished—a work in progress—but still kinda cool. :)

**Input:** The first 10 lines consist of the game board, each line containing 10 capital letters. The next line contains the number of data sets. Each data set contains two integers, the row and column of the bubble you are trying to pop.

**Output:** If the game ends, print out “GAME OVER”. Then print out a blank line and the resulting game board.

**(Continued on next page...)**

**(Problem 12 contin.)**

**Example Input file**

```
ABBCDDABAA  
DAADBDACCB  
CCDECCCAED  
ECCAEEBBAD  
DADCBAECBD  
DEDADDEBDB  
BCEEEBCCDB  
CACBBACBCB  
ADBCADAACB  
ABCDABBBBB  
5  
2 4  
2 0  
6 4  
9 9  
9 9
```

**Output to screen:**

```
XXXXXXXXXX  
AXXCXXXBXX  
DBBDDDACXX  
EAAEBDAAAX  
DADAEEBBCX  
DEDCBAECEX  
BCDADDEBAX  
CACBBBCCBX  
ADBCAACBCA  
ABCDADAACB
```

## 13. Part Of Speech

Program Name: PartOfSpeech.java Input File: partofspeech.dat

The word “die” has several meanings. Here are several:

- die (v) – to stop living – He will die if he does not get a transplant.
- die (v) – to stop working – The car can die if you get water in the fuel tank.
- die (n) – a polyhedron used in games of chance – We roll a die in our game.
- die (n) – a stamping or pressing tool – A die was placed in the machine to create our template.

There are many parts of speech in the English language. There are nouns, verbs, adjectives, adverbs, pronouns, conjunctions, definite and indefinite articles, and probably many more I would know if I were an English teacher.

In this program, you will determine if a word is a noun or verb based on the following criteria:

- If a word is preceded by the articles [a, an, the, this], it is a noun
- If a word is preceded by a helping verb [have, has, had, do, did, does.....], then it is a verb.

**Input:** The first line consists of the articles to determine if the part of speech is a noun. The next two lines contains of a list of helping verbs to determine if the part of speech is a verb. The next line consists of the number of data sets. Each data set consists of a word to search for and the number of sentences in that data set (N). There will be N lines with a sentence on each line. The first 3 lines of input will be identical to the judge data set!

**Output:** Print out the word for each data set followed by N lines showing either “NOUN” or “VERB” for each sentence in the data set. Separate data sets with a blank line.

### **Example Input File**

```
a an the this
have has had do did does would could should can may might must
will shall am are is was were be been being
2
die 4
He will die if he does not get a transplant.
The car can die if you get water in the fuel tank.
We will roll a die to move the car in Monopoly.
A die was placed in the machine to create our template.
face 2
I can face my fears.
Please don't hit me in the face!
```

**(Continued on next page...)**

**(Problem 1 contin.)**

**Output to screen:**

VERB

VERB

NOUN

NOUN

VERB

NOUN

## 14. Touchy

Program Name: Touchy.java

Input File: touchy.dat

I have a touch lamp that sits beside my bed. If it is off and you touch it, it turns on dim. If you touch it again, it turns on medium power. If you touch it again, it turns on high power. One more touch and it goes off. So if it is on low power, it takes 3 touches to turn it off. Since it is usually on low power before bedtime, I usually touch it 3 times to turn it off. However, if it's on medium and I touch it 3 times, it turns back on low!

Write a program that determines the final power setting of a lamp after it is touched a certain number of times! The lamp can have one of 4 power settings:

0123 (0=off, 1=low, 2=medium, 3=high)

**Input:** The first line consists of the number of data elements in the file, followed by that number of lines. Each subsequent line contains two integers: the initial power setting (0-3) and the number of times the lamp is touched.

**Output:** Print the final power setting for each data set (0-3).

### **Example Input File**

```
6
0 1
1 2
2 3
0 20
3 22
3 7
```

### **Output to screen:**

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

QUESTION 1  
A computer program has a variable named `count` which contains the value 10. It then uses a `for` loop to add 1 to `count` 10 times. After the loop has completed, what will be the value of `count`?  
  
A) 10  
B) 11  
C) 20  
D) 21  
  
ANSWER  
The answer is C) 20.  
  
Explanation:  
The variable `count` starts at 10. The loop adds 1 to `count` 10 times. This means that after the first iteration, `count` is 11. After the second iteration, `count` is 12. This continues until after the tenth iteration, where `count` is 20. At this point, the loop has completed its iterations, and the final value of `count` is 20.

## 15. Almost

Program Name: Almost.java

Input File: almost.dat

My daughter in kindergarten likes to play tic-tac-toe with me (at restaurants on the kid menu, or at home on paper, or in chalk on the sidewalk...). Her strategy is not high level yet, but she can always win if she has two letters in a row with one missing space. Write a program to make sure she always wins!

**Input:** The first line consists of the number of data sets in the file. Each data set will consist of 3 lines of 3 characters each (space, capital X, or capital O).

**Output:** For each data set, print out the row and column (1, 2, or 3). Every data set will have only one way for the X to win.

### **Example Input File**

```
3
XOO
 X
 O
OXO
 X
 O

OXO
 X
```

### **Output to screen:**

```
3 3
3 2
1 1
```

1. In the following code, what will be printed to the screen?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

## 16. Integral

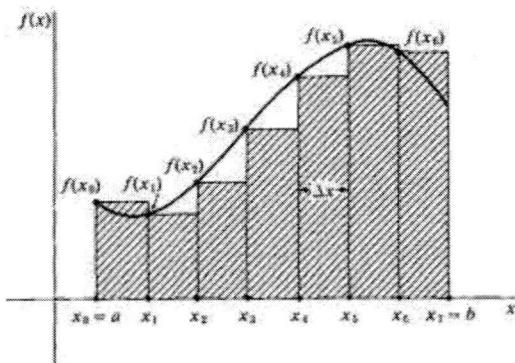
Program Name: Integral.java

Input File: integral.dat

In calculus, the integral is the area under a curve (also the inverse of a derivative). In this program you will find a numerical value for the value of:

$$F(x) = \int_{x_1}^{x_2} f(x) dx$$

This is the area under the curve  $f(x)$  from  $x_1$  to  $x_2$  and is called a definite integral.



In a sense, you find the area of a bunch of rectangles and add them together! (For the calculus student, this is a left Riemann sum).

You will have an initial x, a final x, and how large dx will be. For this program, you will use the following function:  $f(x) = 3x^2 + 2x + 1$ .

**Input:** The first line consists of the number of data sets in the file. Each subsequent line will have 3 double variables: the initial x, a final x, and dx (the width of the rectangles).

**Output:** For each data set, print out the integral to six decimal places.

### Example Input File

```
3
0 2.5 0.1
0 10 1.0
0 10 0.05
```

### Output to screen:

```
23.200000
955.000000
1102.012500
```



## 17. Major MalFunction

Program Name: MalFunction.java

Input File: malfunction.dat

You are going to evaluate a quadratic function in the form  $f(x) = ax^2+bx+c$ , where a, b, and c are integers and x is a rational number (double variable).

**Input:** The first line consists of the number of data sets in the file. Each subsequent line will have 3 integer variables and 1 double variable: the coefficients of the quadratic polynomial and the independent variable, x.

**Output:** Print out the function value for each data set rounded to the nearest thousandth.

### **Example Input file**

```
3
1 1 1 1
1 1 1 0.5
5 4 3 2
```

### **Output to screen:**

```
3.000
1.750
31.000
```

Java has a built-in class called `Math` which contains many useful methods for performing mathematical calculations. These methods include:

- `sqrt()` – returns the square root of a number.
- `ceil()` – rounds up a number to the nearest integer.
- `floor()` – rounds down a number to the nearest integer.
- `random()` – generates a random number between 0.0 and 1.0.
- `max()` – returns the maximum value of two numbers.
- `min()` – returns the minimum value of two numbers.
- `abs()` – returns the absolute value of a number.
- `pow()` – returns the power of a number.
- `sin()`, `cos()`, `tan()` – returns the sine, cosine, and tangent of an angle respectively.
- `log()` – returns the natural logarithm of a number.
- `exp()` – returns the exponential value of a number.
- `sqrt()` – returns the square root of a number.

The `Math` class also contains constants for the values of  $\pi$  and  $e$ .

Java also has a built-in class called `Math` which contains many useful methods for performing mathematical calculations. These methods include:

The `Math` class also contains constants for the values of  $\pi$  and  $e$ .

## 18. Taxing

Program Name: Taxing.java

Input File: taxing.dat

Amazon.com purchases used to have no sales tax in Texas. In addition to its free shipping, this was a great incentive to buy from them. Then it built warehouses in Texas, so sales taxes now apply to purchases for Texas residents. Use a sales tax rate of 8.25% and calculate the total bill for your order!

**Input:** The first line will indicate the number of data sets. Each data set will consist of a double variable (the purchase amount) on a separate line.

**Output:** Print out the total bill (with tax) with dollar sign in front and rounded to the nearest cent.

### **Example Input file**

```
3
100.00
59.99
10.01
```

### **Output to screen:**

```
$108.25
$64.94
$10.84
```

1. **What is the difference between a linked list and an array?**

A linked list is a sequence of nodes connected by pointers. Each node contains data and a pointer to the next node in the sequence. An array, on the other hand, is a collection of elements indexed by their position. Each element is stored at a specific memory location and can be accessed directly by its index.

2. **What is the difference between a stack and a queue?**

A stack is a Last-In-First-Out (LIFO) data structure. It follows the principle of "Last In, First Out". Elements are added at the top and removed from the top. A queue, on the other hand, is a First-In-First-Out (FIFO) data structure. It follows the principle of "First In, First Out". Elements are added at the rear and removed from the front.

3. **What is the difference between a linked list and a stack?**

A linked list is a sequence of nodes connected by pointers. Each node contains data and a pointer to the next node in the sequence. A stack is a Last-In-First-Out (LIFO) data structure. It follows the principle of "Last In, First Out". Elements are added at the top and removed from the top.

4. **What is the difference between a stack and a queue?**

A stack is a Last-In-First-Out (LIFO) data structure. It follows the principle of "Last In, First Out". Elements are added at the top and removed from the top. A queue, on the other hand, is a First-In-First-Out (FIFO) data structure. It follows the principle of "First In, First Out". Elements are added at the rear and removed from the front.

## 19. Speed

Program Name: Speed.java

Input File: speed.dat

In physics class, you learn that the speed of sound is really slow compared to the speed of light. A good way to estimate distances is to measure how long it takes a sound wave to travel a certain distance. A good rule of thumb is to count the number of seconds when you see lightning or fireworks or other things that you can both see and hear. Sound can travel a mile in roughly 5 seconds. Given a time in seconds, estimate the distance in miles!

(To be more precise for physics class, the speed of sound at normal temperature and pressure is 343.0 m/s. In 5.000 seconds sound can travel 1715 m, which is 1.066 miles.)

**Input:** The first line will indicate the number of data sets. Each data set will consist of an integer, the time in seconds.

**Output:** Print out the distance in miles (rounded to the nearest tenth).

**Example Input file**

3  
1  
5  
12

**Output to screen:**

0.2  
1.0 1.1  
2.4 2.4e

1. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    print(i)
```

2. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    print(i)
    for j = 1 to 3 do
        print(j)
```

3. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    print(i)
```

4. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    print(i)
```

5. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    print(i)
```

6. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    for m = 1 to 1 do
        print(m)
    print(i)
```

7. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    for m = 1 to 1 do
        print(m)
    for n = 1 to 1 do
        print(n)
    print(i)
```

8. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    for m = 1 to 1 do
        print(m)
    for n = 1 to 1 do
        print(n)
    for o = 1 to 1 do
        print(o)
    print(i)
```

9. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    for m = 1 to 1 do
        print(m)
    for n = 1 to 1 do
        print(n)
    for o = 1 to 1 do
        print(o)
    for p = 1 to 1 do
        print(p)
    print(i)
```

10. In the following code, what will be printed? Explain your answer.

```
for i = 1 to 5 do
    for j = 1 to 3 do
        print(j)
    for k = 1 to 2 do
        print(k)
    for l = 1 to 1 do
        print(l)
    for m = 1 to 1 do
        print(m)
    for n = 1 to 1 do
        print(n)
    for o = 1 to 1 do
        print(o)
    for p = 1 to 1 do
        print(p)
    for q = 1 to 1 do
        print(q)
    print(i)
```

## 20. Resize

Program Name: Resize.java

Input File: resize.dat

When you resize a scanned/digital picture (larger or smaller) you don't always get the same ratio as the next photo paper size. For example, when you take a 5"x7" scan and try to resize to a 3"x5" the ratios are a little off. To the nearest hundredth, you can get either a 3.00" by 4.20" or a 3.57" by 5.00" using the 5:7 ratio of the original. Given a picture scan and a desired output, show both possible sizes so you know which way to crop!

**Input:** The first line consists of the number of data elements in the file, followed a picture size and the desired scaled picture size, each on one line. The input sizes will be integers.

**Output:** Show the two resized possibilities, to the nearest hundredth of an inch.

### **Example Input file**

```
3
5 by 7 to 3 by 5
8 by 10 to 5 by 7
8 by 10 to 4 by 6
```

### **Output to screen:**

```
3.00 by 4.20 or 3.57 by 5.00
5.00 by 6.25 or 5.60 by 7.00
4.00 by 5.00 or 4.80 by 6.00
```



## 21. Puddle Bop

Program Name: Puddle.java

Input File: puddle.dat

You are playing a game where you are trying to jump over puddles of water. The game is set up as a matrix (rows and columns). You can only jump on small islands of dry ground. The water cells are labeled as “W” and the dry ground is labeled “G”. There are rocks, “R”, and sandy areas, “S”. Once you jump over the water, is it frozen, so it turns to “F.”

- If you jump onto a patch of dry ground (any size of contiguous G cells vertically or horizontally), then output “OK”.
- If you jump onto water (any size of contiguous W cells), then that puddle turns into F cells and output “PUDDLE”.
- If you jump onto rocks or sand (R or S), then output “OOPS”.

Write the code for this game! Assume that the game either continues (G or W) or will end (R or S) on the last move.

**Input:** The first line will indicate the size of the game board (R and C both less than 20). The next R lines contain the matrix consisting of capital letters. The next line will contain the number of moves in the game, N (no more than 20). The next N lines will contain two integers, the row and column (array indices) of the move separated by a space.

**Output:** For each move, output the appropriate statement (“OK”, or “PUDDLE”, or “OOPS”). Then print out a blank line and the resulting game board.

### Example Input file

```
10 10
GWGGGGRRGG
WWGGGGGGSSG
RSGGGGSSSS
SSSWWWGGGG
SSRWGGGGG
RRRWGGGGGG
GGGGRRRGGG
GGGSSSSSG
WWWGGWWWWR
GWWWWWGGS
6
0 0
1 1
2 2
2 5
4 4
9 9
```

**(Continued on next page...)**

**(Problem 12 contin.)**

**Output to screen:**

OK  
PUDDLE  
OK  
OK  
PUDDLE  
OOPS

FGGGGGRRGG  
FFGGGGGSSG  
RSGGGGSSSS  
SSSFFFGGGG  
SSRFFGGGGG  
RRRFGGGGGG  
GGGGRRRGGG  
GGGSSSSSSG  
WWWGGWWWR  
GWWWWWGGS

## 22. Baseball Bin Berry Berry Good To Me

Program Name: baseball.java

Input File: baseball.dat

**General Statement :** For you diehard SNL fans, you will recognize this (slightly politically incorrect) quote from Chico Escuela, the fictional baseball player character introduced by Garrett Morris in the fourth season back in 1978, during a hilarious interview with Jane Curtin (or “Hane”, as he pronounced her name) in the Weekend Update sketch for that episode.

Nevertheless, Chico’s manager needs your help with the scorecard he keeps after every play. You will be given a status of what runners are currently on base, and then the type of hit the batter makes (0 = out, 1 = single, 2 = double, 3 = triple). The current status of the runners on base will be indicated by a 1 or 0, 1 meaning on base, 0 showing the base is empty. An input line of 1 0 1 2 means runners are on 1<sup>st</sup> and 3<sup>rd</sup>, and the batter hits a double, resulting in an output line of 0 1 1 1, meaning the batter reached 2<sup>nd</sup>, runner is still on 3<sup>rd</sup>, and 1 run scored. First base is empty.

**Input:** Several sets of four integers as described above, with one space between each, all on one line.

**Output:** The resulting outputs for the given sets, also as described above, with at least one space between each value, all on one line.

### Example Input File

```
1 1 0 0
1 1 1 2
0 0 1 1
1 1 1 4
```

### Example Output to screen:

```
1 1 0 0
0 1 1 2
1 0 0 1
0 0 0 4
```

QUESTION 1

What is the output of the following program?

```
for i = 1 to 5
    print i
```

ANSWER 1

The output of the program is:

```
1
2
3
4
5
```

QUESTION 2

What is the output of the following program?

```
for i = 1 to 5
    print i
    for j = 1 to 3
        print j
```

ANSWER 2

The output of the program is:

```
1
2
3
4
5
1
2
3
1
2
3
1
2
3
```

QUESTION 3

What is the output of the following program?

```
for i = 1 to 5
    for j = 1 to 3
        print i
    print j
```

ANSWER 3

The output of the program is:

```
1
1
1
1
1
2
2
2
2
2
3
3
3
3
3
4
4
4
4
4
5
5
5
5
5
```

QUESTION 4

What is the output of the following program?

```
for i = 1 to 5
    for j = 1 to 3
        print i
    for k = 1 to 2
        print j
    print k
```

ANSWER 4

The output of the program is:

```
1
1
1
1
1
2
2
2
2
2
3
3
3
3
3
4
4
4
4
4
5
5
5
5
5
```

QUESTION 5

What is the output of the following program?

```
for i = 1 to 5
    for j = 1 to 3
        print i
    for k = 1 to 2
        print j
    for l = 1 to 2
        print k
```

ANSWER 5

The output of the program is:

```
1
1
1
1
1
2
2
2
2
2
3
3
3
3
3
4
4
4
4
4
5
5
5
5
5
```

QUESTION 6

What is the output of the following program?

```
for i = 1 to 5
    for j = 1 to 3
        print i
    for k = 1 to 2
        print j
    for l = 1 to 2
        print k
    for m = 1 to 2
        print l
```

ANSWER 6

The output of the program is:

```
1
1
1
1
1
2
2
2
2
2
3
3
3
3
3
4
4
4
4
4
5
5
5
5
5
```

## 23. Box J

Program Name: BoxJ.java

Input File: none

**General Statement :** Output this 4X5 box of stars with the letter J imbedded as shown.

**Input:** None

**Output to screen:**

```
*****  
*****  
** J **  
*****
```



## 24. Day To Day

Program Name: daytoday.java

Input File: daytoday.dat

**General Statement :** Given two dates, calculate and output the number of days between the two dates, exclusive. For example, there are 19 days between June 2, 2012 and June 22, 2012, namely June 3,4,5...21.

**Input:** An initial N indicating N sets of data to follow, each data set consisting of six integers, each set on one line, consisting of the starting month, day, and year and then the ending month, day, and year. The second date is guaranteed to be after the first date by at least one day, but could be in a later year.

**Output:** The number of days between each pair of given dates.

**Example Input File**

```
3
6 2 2012 6 22 2012
7 4 2011 12 25 2011
12 28 1980 1 1 1981
```

**Example Output to screen:**

```
19
173
3
```

QUESTION 1  
A classmate has asked you to help him learn how to use Java to program a game. He has asked you to explain what Java is, what it does, and how he can use it to program a game. You have agreed to help him, so you have decided to write a short report for him. Your report will include:  
  
1. An explanation of what Java is.  
2. An explanation of what Java does.  
3. An explanation of how he can use Java to program a game.

## 25. Flex J Box

Program Name: flexjbox.java

Input File: flexjbox.dat

**General Statement :** Given four positive integers, output the resulting “J in the box” as shown.

**Input:** An initial integer N, followed by N sets of four integers indicating rows, columns, and J position (row,col) in the box

**Output:** The resulting “J in the box” according to the given data, with at least one blank line of separation between each output. Note: The “J” will always be inside the box, and never on the border of the box.

**Assumptions – Helpful Hints :** Remember, row first, then column

### Example Input File

```
2
4 5 2 2
10 9 3 4
```

### Example Output to screen:

```
*****
*****
**J**
*****
*****
***** 
***** 
***** 
****J*** 
***** 
***** 
***** 
***** 
***** 
*****
```

QUESTION 1  
A linked list is a sequence of nodes. Each node contains a value and a reference to the next node in the sequence. The last node in the sequence has a reference to null.  
The code below defines a class named Node. The class has a constructor that takes a value and a reference to the next node. The class also has a method named print which prints the value of the node.  
  
class Node {  
 int value;  
 Node next;  
 public Node(int value) {  
 this.value = value;  
 this.next = null;  
 }  
 void print() {  
 System.out.println(value);  
 }  
}  
  
The code below creates a linked list with three nodes. The first node has a value of 1, the second node has a value of 2, and the third node has a value of 3. The third node's next reference is set to null.  
  
Node n1 = new Node(1);  
Node n2 = new Node(2);  
Node n3 = new Node(3);  
n1.next = n2;  
n2.next = n3;  
n3.next = null;  
  
What is printed when the code n1.print(); is run?  
  
A) 1  
B) 2  
C) 3  
D) 1, 2, 3

## 26. Gender Race

Program Name: gender\_race.java

Input File: gender\_race.dat

**General Statement :** For the first time in the history of racing, males and females will compete against each other. What a concept! You have been hired to keep track of the winners. Given at most twenty names, finishing time and gender for each racer, determine the best and second best time overall, best and second best among the males, and finally best and second best among the females. There will always be at least two males and two females in the race.

**Input:** Integer N showing how many racers will be in the race, followed by N sets of data containing first name, gender, and finishing time for each racer. There will be no ties, and no more than 20 racers, and only one race.

**Output:** Name, time, and gender for each of the following:

- Overall winner
- Overall second
- Best Male
- Second place male
- Best female
- Second place female

### Example Input File

```
6
Sue 48 F
Kelly 52 M
Stacey 43 F
Kelly 49 F
Stacey 50 M
John 44 M
```

### Example Output to screen:

```
Stacey 43 F
John 44 M
John 44 M
Stacey 50 M
Stacey 43 F
Sue 48 F
```

QUESTION 10  
A linked list is a sequence of nodes. Each node contains a value and a pointer to the next node in the sequence.

QUESTION 11  
A stack is a data structure that follows the Last In First Out (LIFO) rule.

QUESTION 12  
The `for` loop is used to iterate over a collection of items. It is typically used when you know the number of iterations required.

QUESTION 13  
A function is a block of code that performs a specific task and can be called from other parts of a program.

QUESTION 14  
A variable is a named storage location that holds a value.

QUESTION 15  
A class is a template for creating objects. It defines the properties and behaviors that all objects of that class share.

QUESTION 16  
An array is a collection of elements of the same type stored at contiguous memory locations.

QUESTION 17  
A pointer is a variable that stores the memory address of another variable.

QUESTION 18  
A function is a block of code that performs a specific task and can be called from other parts of a program.

QUESTION 19  
A variable is a named storage location that holds a value.

QUESTION 20  
A class is a template for creating objects. It defines the properties and behaviors that all objects of that class share.

## 27. Snapshot Day

Program Name: Snapshot.java

Input File: snapshot.dat

**General Statement :** Schools track tardies and attendance every day, but once a year take a “snapshot” of the situation. Your job is to take the data gathered and do a brief analysis report.

**Input:** Data gathered for each of six classrooms, one each for classes 1 through 6, consists of total enrolled in the class, number absent that day, and number tardy. There will be no duplicate data sets.

**Output:** The output will be as follows:

1. Total number enrolled in all classes
2. Total number of tardies in all classes
3. The class with the best attendance (highest percentage not absent)
4. The list of classes in order from best to worst attendance for the day according to percentage not absent.

### Example Input File

```
10 1 1
10 0 1
20 3 2
20 6 0
20 1 3
20 4 6
```

### Example Output to screen:

```
100
13
2
2 5 1 3 6 4
```

1. In the following code, what will be printed?  
public class Main {  
 public static void main(String[] args) {  
 System.out.println("Hello World");  
 System.out.println("Hello Again");  
 System.out.println("Hello There");  
 }  
}

## 28. Taxi, Taxi, UIL Taxi!

Program Name: taxi.java

Input File: taxi.dat

**General Statement :** The cost to ride a taxi in UIL land is 50 cents for the first 1/5 mile (or less) and 22 cents for each additional 1/5 mile or part thereof. In addition, the taxi will wait for you while you conduct your business. The cost for waiting is 20 cents per 60 second period or any part thereof.

**Input:** Several sets of data, each set consisting of two non-negative values, each set on one line, separated by a space. The first value is the number of miles traveled, and the next is the number of seconds of waiting time.

**Output:** The cost of each taxi ride, with a +- 1 cent tolerance of error. Dollar format output required, each answer on one line.

**Example Input File**

```
1 0
2 0
1.5 75
3.4 125
```

**Example Output to screen:**

```
$1.38
$2.48
$2.44
$4.62
```

QUESTION 10  
What is the output of the following code?

```
for i = 1 to 5 do
    print(i)
```

The code provided is a simple for loop. It starts at  $i = 1$  and increments  $i$  by 1 until it reaches 5. At each iteration, it prints the value of  $i$ . Therefore, the output will be:

1  
2  
3  
4  
5

## 29. Test Parity

Program Name: testparity.java

Input File: testparity.dat

**General Statement :** In a somewhat misguided effort to “level the testing field”, your teacher has decided to assess a “tax” on all test scores. Tests will be greater than zero but no higher than 120. Using the following tax table, all test scores will be adjusted so it appears that everyone seems to be doing fairly well, hoping the administration will be pleased.

Score Range:

$0 < \text{score} \leq 25$	Action: Double the score
$25 < \text{score} \leq 50$	Increase score by 50%
$50 < \text{score} \leq 75$	Increase score by 25%
$75 < \text{score} \leq 100$	No change
$100 < \text{score} \leq 120$	Decrease score by 10%

Your job is to write a program that will make the appropriate adjustment for each score.

**Input:** Several test scores, all on one line, each separated by one space.

**Output:** The rounded integer adjusted scores for each original score, all on one line, each separated by at least one space.

**Example Input File**

12 50 80 117

**Example Output to screen:**

24 75 80 105

QUESTION 10  
A linked list is a collection of nodes. Each node contains a value and a pointer to the next node in the list. The first node in the list is called the head of the list. The last node in the list is called the tail of the list. If you want to add a new node to the end of a linked list, you would:  
A) Add the new node to the beginning of the list.  
B) Set the value of the new node to the value of the current tail node.  
C) Set the value of the new node to the value of the current head node.  
D) Set the value of the new node to the value of the current tail node's pointer.  
E) Set the new node's pointer to the current tail node's value.

## 30. Weird Change

Program Name: WeirdChange.java

Input File: weirdChange.dat

**General Statement :** In UIL land they do things weird, including their monetary system, where the basic unit is, you guessed it, the “uil”! Each “uil” is worth 3 cents in US currency. The only coins in this system have values of 1, 5, 13, 23, 37 and 47. Your job is to help your UIL coach with his money during the visit to UIL land. The reason he/she needs help is because of his/her obsession with giving exact change as well as the absolute fewest number of coins necessary for the transaction.

**Input:** Several positive integers, each between 1 and 200, representing how many “uils” something costs.

**Output:** The dollar amount equivalent to the given “uil” cost, and six values representing the exact change using the fewest “uil” coins (in descending order of value) needed to make the transaction. Assume your coach always has enough coins to make the purchase.

### Example Input File

```
12
41
34
50
```

### Example Output to screen:

```
$0.36 0 0 0 0 2 2
$1.23 0 0 1 1 1 0
$1.02 0 0 1 0 2 1
$1.50 0 1 0 1 0 0
```

