

## 1.

Suppose we can buy a chocolate bar from the vending machine for \$1 each. Inside every chocolate bar is a coupon. We can redeem 7 coupons for one chocolate bar from the machine. We would like to know how many chocolate bars can be eaten, including those redeemed via coupon, if we have  $n$  dollars.

For example, if we have 20 dollars then we can initially buy 20 chocolate bars. This gives us 20 coupons. We can redeem 14 coupons for 2 additional chocolate bars. These two additional chocolate bars have 2 more coupons, so we now have a total of 8 coupons when added to the six leftover from the original purchase. This gives us enough to redeem for one final chocolate bar. As a result we now have 23 chocolate bars and 2 leftover coupons.

Write a program that inputs the number of dollars and outputs how many chocolate bars you can collect after spending all your money and redeeming as many coupons as possible. Also output the number of leftover coupons. The easiest way to solve this problem is to use a loop.

**CodeMate Hint:** Use a while loop that redeems coupons for bars and calculates the new number of coupons. Continue the loop as long as you have enough coupons to redeem for a chocolate bar.

### OUTPUT:

```
Enter the number of dollars you have to spend:
20
You can get 23 candy bars with 2 coupons leftover.
```

## 2.

Write a program that prints the sum of the digits of an integer. For example if the input number is 12349 then the sum of the digits is  $9 + 4 + 3 + 2 + 1 = 19$ .

**CodeMate Hint:** One way to solve this problem is to use a loop. Use % operator get the least significant bit of the decimal integer. Use / operator to truncate the least significant digits one by one. The loop will iterate till the number reduces to 0.

### OUTPUT:

```

Enter an Integer
12349
Sum of the digits of 12349 is 9+4+3+2+1 = 19

```

3.

Write a program to read the marks scored by a student in six subjects and announce the result of the student as per the following guidelines. To graduate, the student should score a minimum of 40 marks in each subject. If the student is eligible to graduate, she will be awarded two letter grades based on the percentage of marks scored. The following table gives the break up:

Percentage of Marks	Grade
> 90%	AA
> 80%	AB
> 70%	BB
> 60%	BC
> 50%	CC
> 40%	CD
> =35%	DD
> 35%	FF

**OUTPUT:**

```

Enter the Marks scored in Subject1:90
Enter the Marks scored in Subject2:89
Enter the Marks scored in Subject3:99
Enter the Marks scored in Subject4:100
Enter the Marks scored in Subject5:92
Enter the Marks scored in Subject6:88

Eligible for Promotion to next semester
Percentage is 93 Grade is AA

```

```

Enter the Marks scored in Subject1:90
Enter the Marks scored in Subject2:89
Enter the Marks scored in Subject3:78
Enter the Marks scored in Subject4:99
Enter the Marks scored in Subject5:05
Enter the Marks scored in Subject6:100
Not eligible for Promotion

```

4. Write a program that outputs all 99 stanzas of the "Ninety Nine Bottles of Beer on the Wall" song. Your program should print the number of bottles in English, not as a number.

For example:

Ninety nine bottles of beer on the wall,  
Ninety nine bottles of beer,  
Take one down, pass it around,  
Ninety eight bottles of beer on the wall.

...

One bottle of beer on the wall,  
One bottle of beer,  
Take one down, pass it around,  
Zero bottles of beer on the wall.

**Your program should not use ninety nine different output statements!**

**CodeMate Hint:** Write a function that takes as input an integer between 0 and 99 and outputs that value in English. Use / and % to extract the tens and the ones digit so you know what number to output in English.

CodeMate Hint: You may need to test specifically for numbers such as 0, 10-19.

**OUTPUT:**

```
Ninety Nine bottles of beer on the wall,  
Ninety Nine bottles of beer on the wall,  
Take one down, pass it around,  
Ninety Eight bottles of beer on the wall,  
  
Ninety Eight bottles of beer on the wall,  
Ninety Eight bottles of beer on the wall,  
Take one down, pass it around,  
Ninety Seven bottles of beer on the wall,  
Ninety Seven bottles of beer on the wall,
```

.....

```
Three bottles of beer on the wall,  
Three bottles of beer on the wall,  
Take one down, pass it around,  
Two bottles of beer on the wall,  
  
Two bottles of beer on the wall,  
Two bottles of beer on the wall,  
Take one down, pass it around,  
One bottle of beer on the wall,
```

## 5.

The game of Pig is a simple two player dice game in which the first player to reach 100 or more points wins. Players take turns. On each turn a player rolls a six-sided die. After each roll:

- If the player rolls a 2-6 then he can either:
  - ROLL AGAIN or
  - HOLD. At this point the sum of all rolls made this turn is added to the player's total score and it becomes the other player's turn.
- If the player rolls a 1 then the player loses his turn. He gets no new points and it becomes the opponent's turn.

If a player reaches 100 or more points after holding then the player wins.

Write a program that plays the game of Pig, where one player is a human and the other is the computer. Allow the human to input "r" to roll again or "h" to hold.

The computer program should play according to the following rule: Keep rolling on the computer's turn until it has accumulated 20 or more points, then hold. Of course, if the computer wins or rolls a 1 then the turn ends immediately.

Allow the human to roll first.

Write your program using at least two functions:

**int humanTurn(int humanTotalScore);**

**int computerTurn(int computerTotalScore);**

These functions should perform the necessary logic to handle a single turn for either the computer or the human. The input parameter is the total score for the human or computer. The functions should return the turn total to be added to the total score upon completion of the turn. For example, if the human rolls a 3 and 6 and then holds, then humanTurn should return 9. However, if the human rolls a 3 and 6 and then a 1, then the function should return 0.

**Notes:** Special attention should be made to the description of the humanTurn and computerTurn functions. It indicates that an outside variable must keep track of the total score and add it to the value returned by the functions at the completion of each player's turn.

**OUTPUT:**

```

It's your turn! Enter 'r' to roll.
r
You rolled 2
Your score this round is: 2
If you hold, your total score would be: 2
Press 'h' to hold or 'r' to roll again.
r
You rolled 3
Your score this round is: 5
If you hold, your total score would be: 5
Press 'h' to hold or 'r' to roll again.
r
You rolled 1
You lose your turn. Enter 'c' to continue.

It's the computer's turn!
The computer rolled 6
The computer's score this round is: 6
If the computer holds, its total score would be: 6
The computer rolled 5
The computer's score this round is: 11
If the computer holds, its total score would be: 11
The computer rolled 4
The computer's score this round is: 15
If the computer holds, its total score would be: 15
The computer rolled 2
The computer's score this round is: 17
If the computer holds, its total score would be: 17
The computer rolled 6
The computer's score this round is: 23
If the computer holds, its total score would be: 23
The computer holds. Enter 'c' to continue.

```

## 6.

Write a program that inputs a date (e.g. July 4, 2008) and outputs the day of the week that corresponds to that date. The following algorithm is from [http://en.wikipedia.org/wiki/Calculating\\_the\\_day\\_of\\_the\\_week](http://en.wikipedia.org/wiki/Calculating_the_day_of_the_week). The implementation will require several functions:

**bool isLeapYear(int year);**

This function should return true if year is a leap year and false if it is not. Here is pseudocode to determine a leap year:

```

leap_year = ((year divisible by 400) or (year divisible by 4 and year not
divisible by 100))

```

**int getCenturyValue(int year);**

This function should take the first two digits of the year (i.e. the century), divide by 4, and save the remainder. Subtract the remainder from 3 and return this value multiplied by 2. For example, the year 2008 becomes:  $(20/4) = 5$  remainder 0.  $3 - 0 = 3$ . Return  $3 * 2 = 6$ .

**int getYearValue(int year);**

This function computes a value based on the years since the beginning of the century. First, extract the last two digits of the year. For example, 08 is extracted for 2008. Next, factor in leap years. Divide the value from the previous step by 4 and discard the remainder. Add the two results together and return this value. For example, from 2008 we extract 08. Then  $(8/4) = 2$  remainder 0. Return  $2 + 8 = 10$ .

**int getMonthValue(int month, int year);**

This function should return a value based on the table below and will require invoking the isLeapYear function:

Month	Return Value
January	0 (6 if year is a leap year)
February	3 (2 if year is a leap year)
March	3
April	6
May	1
June	4
July	6
August	2
September	5
October	0
November	3
December	5

Finally, to compute the day of the week, compute the sum of the date's day plus the values returned by getMonthValue, getYearValue, and getCenturyValue. Divide the sum by 7 and compute the remainder. A remainder of 0 corresponds to Sunday, 1 corresponds to Monday, etc. up to 6 which corresponds to Saturday. For example, the date July 4, 2008 should be computed as **(day of month) + (getMonthValue) + (getYearValue) + (getCenturyValue)** =  $4 + 6 + 10 + 6 = 26$ .  $26/7 = 3$  remainder 5. The fifth day of the week corresponds to Friday.

Your program should allow the user to enter any date and output the corresponding day of the week in English.

**OUTPUT:**

```
Enter month (1-12)
3
Enter day (1-31)
09
Enter year (0000-9999)
2016
Wednesday
```

注意:OUTPUT 僅供參考