cout, endl and “\n”

1. Write a program that displays "This statement is false".

2. Write a program that displays the following lines.

When earth is sundered, and skies choked black,
And sleepers serve the seven curses,
To the hearth there comes a stranger,
Journeyed far 'neath moon and star.

3. Write a program that displays the following:

```
##   ## #######
##   ##      ##
##   ##      ##
####### #######
     ## ##
     ## ##
     ## #######
```

4. Write a program that displays the following robot.

```
   =====
  | o o |
   =====
    | |
   =====
==[ Who ]==
| [ am   ] |
^ [  I? ] ^
   =====
   \/ \/
   || ||
   [] []
-----------
```

Arithmetic operators and sqrt(), pow(), abs() functions

1. Write a program that takes 3 inputs from the user: 1 integer, 1 floating point number and 1 character. Then display what the user entered using cout.
2. Write a program that displays the sum, difference, product, quotient (division), and remainder of two integers provided by the user.
3. Write a program that calculates the number of seconds, minutes and hours in 1 year.
4. Write a program that computes the result using the formula
$$result = a * (b - c) + 1.37$$
where the values of *a, b*, *c* are provided by the user.
5. Write a program that swaps the values of two variables.
6. Write a program that swaps the values of two variables without using a 3rd variable.
7. Write a program that reads the units in meter and converts it to feet. (1 inch = 2.54 cm).
8. Write a program that displays a temperature in Celsius and in Kelvin after reading the temperature in Fahrenheit from the user. Use the conversion formulae $5(F - 32) = 9C$ and $C = K - 273.15$.
9. Write a program that will take the radius of a circle from the user. Then it will compute and print the area and perimeter of the circle.
10. Write a program that will take the length and width of a rectangle from the user. Then it will compute and print the area and perimeter of the rectangle.
11. Write a program that will take the height and base of triangle from the user. Then it will compute and print the area of the triangle.
12. Write a program that converts the days in to month and years. Here the user supplies the number of days. For example, if the user enters 813 days, the program should print: 2 years 2 months 23 days. (don’t worry about leap year)
13. Write a program that inputs number of cents (from 0 to 99) and outputs the minimal number of pennies (1 cent), nickels (5 cents), dimes (10 cents) and

quarters (25 cents) with the same value. For example, 94 cents is the same as 3 quarters, 1 dime, 1 nickel, and 4 pennies.

14. Write a program that inputs a number of hours and outputs the equivalent number of weeks, days, and hours. For example, an input of 4000 would output 23 weeks, 5 days and 16 hours.
15. Write a program that prompts the user for the current year and the user's current age. It then calculates and prints the user’s birth year.
16. Write a program that takes a 6-digit positive integer (i.e., in the range 100,000-999,999) and then prints the reversed number. For example: if the user enters 289405, the program should print 504982.
17. Write a program to compute and print the value of *y* in the following equation.

$$y = \frac{-5+(6 \times 9)}{2+(5 \div 3)}$$

18. Write a program to compute and print the value of *x* in the following equation. (Hint: use pow() and sqrt() functions and don’t forget to include <cmath>)

$$x = \frac{-5+\sqrt{5^2-16}}{4}$$

19. Write a program to compute and print the value of *z* in the following equation.

$$z = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \frac{1}{720}$$

20. Write a program that will take b and c from the user and compute the value of *a* in the following equation. (Hint: use abs() function)

$$a = \sqrt{|b|} + c^2$$

21. Write a program that will take a, b and c from the user and compute the value of *x*.

$$x = \frac{-b+\sqrt{|b^2-4\mathrm{ac}|}}{2\mathrm{a}}$$

22. Write a program to compute and print the value of *s* in the following equation.

$$s = \frac{1^2}{\sqrt{2}} + \frac{2^3}{\sqrt{3}} + \frac{3^4}{\sqrt{4}} + \frac{4^5}{\sqrt{5}} + \frac{5^6}{\sqrt{6}}$$

if and switch statements

1. Write a program to find whether a number is odd or even.
2. Write a program that takes an integer and checks if the number is divisible by 2, 3, 5, 7 and 10.
3. Write a program that will print your grade when you enter your exam score. The program should use the following grading system:

Score range	Grade
[85, 100]	A
[80, 85)	A-
[75, 80)	B+
[70, 75)	B
[65, 70)	B-
[60, 65)	C+
[55, 60)	C
[50, 55)	C-
[45, 50)	D+
[40, 45)	D
[0, 40)	F

4. Write a program that takes the user's age as input and then finds out whether the user is younger than a teenager, a teenager or older than a teenager.
5. Write a program that takes integer inputs from 0 to 10 and displays the number in words. Example: if the user enters 5, the program should print Five.
6. Write a program that takes 3 numbers from the user and finds out the largest and the smallest number among the three numbers.
7. Write a program to check if a year is leap year or not. Use the following algorithm:

```
if (year is not divisible by 4) then (it is a not a leap year)
else if (year is not divisible by 100) then (it is a leap year)
else if (year is not divisible by 400) then (it is a not a leap year)
else (it is a leap year)
```

8. Write a program to compute the roots of a quadratic equation i.e. $ax^2 + \mathrm{bx} + \mathrm{c} = 0$. Where the formula for finding the roots are:

   $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

   Here the value of a, b and c are given by the user. The program needs to find the determinant by calculating $d = (b^2 - 4ac)$.

   If d is zero the equation has a single root and the value of the root is $(-b/2a)$.

   If the d is positive then there are two real roots and the roots are $(-b + \sqrt{d}\,)/2a$ and $(-b - \sqrt{d}\,)/2a$.

   If the determinant is negative the roots are complex conjugate. The values are $-b/2a\ + (\sqrt{|d|}/2a)i$ and $-b/2a\ - (\sqrt{|d|}/2a)i$.

   Make sure you use floating point numbers. Check your answers using your calculator.

9. Write a program to find out if a character is lower case letter or upper case letter or digit or symbol. (Hint: Learn about ASCII table)
10. Imagine you are writing a program for children. If a child enters the letter ‘a’, your program should display “A is for Apple”. If a child enters ‘z’, it should print “Z is for Zebra”.

Loops

Level I

1. Write a program that will keep taking input (integers) until the user enters -1. (Use while or do-while loop)
2. Write a program that will keep taking input (character) until the user enters a dot ‘.’. (Use while or do-while loop)
3. Write a program that will keep taking input (floats) until the user enters 3.1416. (Use while or do-while loop)
4. Write a program that will print the following series of integers.

$$0, 1, 2, 3, \ldots\ 99$$

5. Write a program that will print the following series of integers.

$$42, 41, 40, 39, \ldots\ 17$$

6. Write a program that will print the following series of floating point numbers.

$$2.25, 2.50, 2.75, 3.00, \ldots\ 4.00$$

7. Write a program that will print the following series of characters.

$$a, b, c, \ldots, z$$

8. Write a program that will print the following series of integers.

$$1, 3, 5, 7, \ldots 73$$

9. Write a program that will print the following series of integers.

$$-42, -40, -38, \ldots - 8$$

10. Take two integers from the user and print all the integers between them.
11. Take two integers from the user and print all the odd numbers between them. You cannot use the % operator.
12. Take two integers from the user and print all the numbers between them that are divisible by 7. You cannot use the % operator.

Level II

1. Take 10 floating point numbers from the user and then print the largest and smallest numbers among the 10 numbers.
2. Take an integer from the user and check if the number is prime or not.
3. Write a program that will count the number of digits in a given integer.
4. Compute the sum, $S = 1 + 2 + 3 + \cdots + 100$
5. Compute the sum, $S = 1 + 3 + 5 + \cdots + 99$
6. Compute the sum, $S = 5 + 10 + 15 + \cdots + 100$
7. Compute the sum, $S = 1 - 2 + 3 - 4 \ldots + 99 - 100$
8. Compute the sum, $S = 1 + 2 + 4 + 8 + \cdots + 1024$
9. Compute the sum, $S = 1 + 2 + 4 + 7 + 11 + \cdots + 56$
10. Write a program that will take 10 numbers from the user and compute the average value.
11. Write a program that will take a floating point number x and an integer y. Then compute the value of x^y without using the **pow**(x, y) function.
12. Write a program that will take an integer and compute its factorial.
13. Write a program that will print the digits of an integer backwards. Example: if the user enters 452643, the program should print 346254.
14. Write a program that will check if a number is palindrome (symmetric). For example, 12521 is a palindrome because if you reverse the number, it remains the same.
15. Take an integer from the user and compute the sum of its digits. For example, if the user enters 4231, the output should be 10.
16. Write a program that will identify a perfect number. In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. For example, 6 is a perfect number because the sum of its positive divisors 1+2+3 is equal to the number itself.

17. Write a program that will identify an Armstrong number. An Armstrong number is an integer such that the sum of the cubes of its digits is equal to the number itself. For example, 371 is an Armstrong number since $3^3 + 7^3 + 1^3 = 371$.

18. Write a program that will find the GCD (Greatest Common Divisor) of two numbers. For example the GCD of 18 and 24 is 6.

19. Write a program that will find the LCM (Least Common Multiple) of two numbers. For example the LCM of 12 and 15 is 60.

20. Write a program that will take an integer from the user and print the following shapes. Example: If the user enters 5, the output should be the following.

```
a)  *              b)  * * * * *   c)         *    d) * * * * *
    * *                * * * *              * *         * * * *
    * * *              * * *              * * *           * * *
    * * * *            * *              * * * *             * *
    * * * * *          *              * * * * *               *
```

21. Create a pyramid and diamond by combining the basics triangles above.

22. Generate the Fibonacci series up to the nth term. The user will provide the value of *n*. The first 10 terms of the series are 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

23. Write a program that checks if a number is a term in the Fibonacci series.

Level III

1. Write a program that will print all the prime numbers between two integers entered by the user.
2. If the user enters two numbers, print all the palindrome numbers between those two numbers.
3. If the user enters two numbers, print all the perfect numbers between those two numbers.
4. Find the sum of the squares of the first 100 odd numbers.
5. Compute the sum, $S = 1^1 + 2^2 + 3^3 + \cdots + 10^{10}$
6. Compute the sum, $S = 1^0 + 2^{-1} + 3^{-2} + \cdots + 10^{-9}$
7. Compute the sum, $S = 1! + 2! + 3! + \cdots + 10!$
8. Compute the value of e^x. The user will enter the value of *x* and *n*. The value of *n* cannot be greater than 10. *x* can be a fraction. Use the following formula.

$$e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

9. Find all the numbers between 10 and 1000 where each number's summation of their digits is a prime number. For example 344 should be printed because 3+4+4=11 which is a prime number.
10. Take 25 integers from the user and count the number of odd, even, positive and negative numbers the user enters. Print the results. Remember 0 is neither positive nor negative but it's an even number.
11. Print the following shapes given the size. If the user enters 5 the shapes should look like the following.

```
a)  1            b)  5 4 3 2 1      c)  1            d)  Five  Four Three Two One
    1 2              1 2 3 4            1 0              Five  Four Three Two
    1 2 3             3 2 1             1 0 1            Five Four Three
    1 2 3 4          1 2                1 0 1 0          Five Four
    1 2 3 4 5         1                 1 0 1 0 1        Five
```

Array

1 Dimensional

1. Fill an integer array of size 100 with random numbers. Find the maximum, minimum and average value from the array.
2. Take two integers from the user. Fill an integer array with random values between the numbers given by the user. Print the array. (also solve this problem with characters instead of integers)
3. Ask the user to enter the size of an array. Create an integer array of the given size and fill it with random numbers. Print the array then take a target number from the user. Search the target value in the array. If the value is found, print the indices (index) where the value was found. Also print how many times the value was found in the array.
4. Generate 100 random integers and store them in an array. Sort the array in descending order. Print both the unsorted values and the sorted values. (also solve this problem with characters instead of integers)
5. Write a program that counts the number of times each number occurs in an array.
6. Create an integer array of size 10. Fill the array by taking input from the user. Then check if the array is sorted or not. If sorted then, print how it is sorted (ascending or descending). If it is not sorted, print that it is not sorted.
7. Find the 1st maximum, 2nd maximum, 1st minimum, 2nd minimum and the median value from a floating point array of size 11.
8. Populate an integer array with random values. Separate the even values and store them in another array and the odd numbers in another array. Print all three arrays.
9. Populate an integer array with random values. Separate the even and odd values and store them in one array. You cannot create separate arrays for odd and even numbers. Print the two arrays.
10. Populate a floating point array with 1000 random floating point values. Perform binary search on the array by taking a target from the user. You only need to tell the user whether the target number was found or not found in the array.

11. Take a number from the user. Create an array of that size. Fill the array with the Fibonacci series. Print the array.
12. Find how many prime numbers are there in an array.
13. Convert a decimal number to binary number.

2 Dimensional

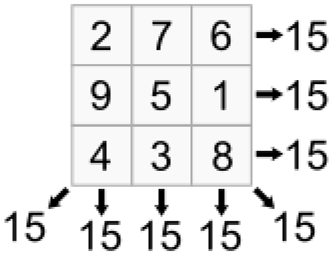
1. Create a 5 by 3 array. Take two integers from the user. Fill the array with random floating point numbers between the numbers entered by the user. Then print the array.
2. Create two 4 by 3 integer matrices called A and B. Add/subtract A and B and store the resultant matrix in C. Print A, B and C.
3. Ask the user to enter row and column sizes. Create a matrix A. Store the transpose of A in another matrix B.
4. Create a 4 by 3 integer matrices called A and a 3 by 2 integer matrix called B. Multiply A and B and store the resultant matrix in C. Print A, B and C.
5. Count the number of prime numbers in a 2D integer array.
6. Compute the average value of each of the columns of a 15 x 10 floating point array.
7. Convert a 5x7 2D array to a 1D array.
8. Convert a 1D array of size 10 to a 2D array (make sure no data is lost).
9. Compute the sum of the diagonal elements of an N x N array.
10. Create a 5 x 5 matrix. Compute the sum of the values below the diagonal and above the diagonal. Print the sums separately.

1	4	5	2	0
2	3	5	9	1
4	6	2	6	1
1	2	2	5	2
0	1	4	2	1

Sum of left diagonal = 12
Sum of right diagonal = 13

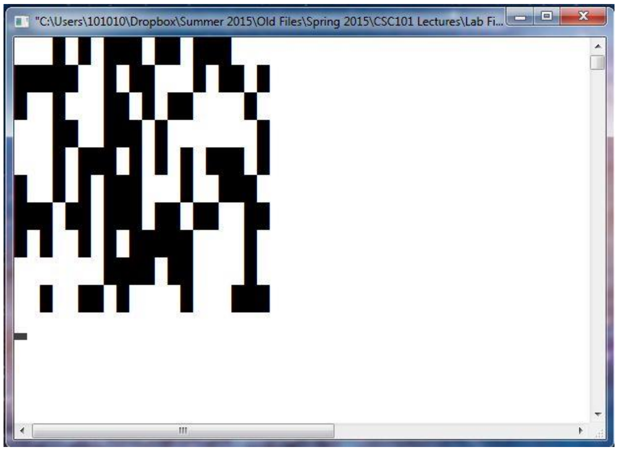
*Challenge Problems are for those who actually enjoys programming (no graphics is required)**

11. **Challenge problem:** Create a magic square of the size specified by the user and store the values in a matrix. Magic square is a square where the sum of every row and column and diagonal is the same. For example, a 3 x 3 magic square looks like:



2	7	6	→15
9	5	1	→15
4	3	8	→15

12. **Challenge problem:** Create a chessboard pattern using extended ASCII codes.
13. **Challenge problem:** Generate a random pattern like below that changes every second.



"C:\Users\101010\Dropbox\Summer 2015\Old Files\Spring 2015\CSC101 Lectures\Lab Fi...

14. **Challenge problem:** Write a program where the user can move a character within a 25 x 25 character matrix using the arrow keys.
15. **Challenge problem:** Write a Tic-Tac-Toe game where the computer never loses.
16. **Challenge problem:** Write a Sokoban game.
17. **Challenge problem:** Write a game where the user has to solve a maze. Add levels with increasing difficulty.
18. **Challenge problem:** Write a game of tetris.

String

1. Take a string from the user and check if it's a palindrome.
2. Count the number of vowels, digits and words in a string.
3. Take a string from the user then replace all the vowels in that string with *.
4. Take the name of the user and store it in a single string, then print the last name followed by a comma, then print the initial letter of the first name followed by a dot. For example, if the user enters "Muhammad Shamsu", your program should print "Shamsu, M."
5. Take 10 names from the user, store them in an array. Then sort the names in ascending order based on the **first** name. Also sort the array based on their **last** names.
6. Take a string from the user, print the reverse string.
7. Take a sentence from the user and print only the first letters of each word.
8. Take a string from the user and count the number of uppercase letter, lowercase letters and digits.
9. Take a sentence and print the words in reverse order (don't reverse the string character by character).
10. Check if an email address is valid or not. A valid email address will have the format: xxxxxxx@xxxxx.xxx. The email address must contain a part before the at (@), another part after at (@) and before the dot (.) and the last part after the dot (.).
11. Write a program to check if a string is a number or not. Example: 12312 is a number but 12312t34 is not a number.
12. Write a program that will ask the user's month (Example: January, February.. December), year (Example: 1990,1992…2010) and date (Example:1,2,..31) of birth. Then the program will print the date in the dd/mm/yy format.
13. Write a program to check if an identifier is valid or not. Identifier is the name of a variable/class/function.

Function

1. Write a function that takes a number and a character and prints a pyramid of that size using the provided character.
2. Write a function that takes a number and returns the factorial.
3. Write a function that takes two or three numbers and returns their HCF. Write another function for finding LCM of two or three numbers.
4. Write a function that checks if a number is prime/perfect/palindrome or not. If the number is prime/perfect/palindrome the function should return true, otherwise false. Use the function in the main function.
5. Write a function that performs linear search when provided with an array and a target value.
6. Write a function that sorts an array of numbers. The function should have an additional input parameter for deciding the order (ascending or descending). First print the unsorted array, then use the function to sort the array in ascending order, print the array, then sort it in descending order, print again.
7. Write a function that returns the minimum, maximum and the average value of a given array.
8. Write a function that computes the sum of the following series. Write your own factorial and power function and use it to compute the sum.

$$e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

9. Write a function that returns *the answer to life the universe and everything*.

10. Write a recursive function that returns the GCD of two numbers.