

---

# ASSIGNMENT # 2

---

## QUESTION # 1

Write a program that uses `for` statements to print the following patterns separately, one below the other. Use `for` loops to generate the patterns. All asterisks (\*) should be printed by a single statement of the form `cout << '*'`; (this causes the asterisks to print side by side).

[Hint: The last two patterns require that each line begin with an appropriate number of blanks. Extra credit: Combine your code from the four separate problems into a single program that prints all four patterns side by side by making clever use of nested `for` loops.

(a)	(b)	(c)	(d)
*	*****	*****	*
**	*****	*****	**
***	*****	*****	***
****	*****	*****	****
*****	*****	*****	*****
*****	*****	*****	*****
*****	*****	*****	*****
*****	****	****	*****
*****	***	***	*****
*****	**	**	*****
*****	*	*	*****

(e)

```

      *
    ***
  *****
*****
*****
*****
  *****
    ***
      *
```

---

# ASSIGNMENT # 2

---

## QUESTION # 2

Write if statements to do the following:

- If character variable `taxCode` is 'T', increase price by adding the `taxRate` percentage of price to it.
- If integer variable `opCode` has the value 1, read in double values for X and Y and calculate and print their sum.
- If integer variable `currentNumber` is odd, change its value so that it is now 3 times `currentNumber` plus 1, otherwise change its value so that it is now half of `currentNumber` (rounded down when `currentNumber` is odd).
- Assign `true` to the boolean variable `leapYear` if the integer variable `year` is a leap year. (A leap year is a multiple of 4, and if it is a multiple of 100, it must also be a multiple of 400.)
- Assign a value to double variable `cost` depending on the value of integer variable `distance` as follows:

Distance	Cost
-----	-----
0 through 100	5.00
More than 100 but not more than 500	8.00
More than 500 but less than 1,000	10.00
1,000 or more	12.00

## QUESTION # 3

Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number.

For example,  $153 = (1 * 1 * 1) + (5 * 5 * 5) + (3 * 3 * 3)$

## QUESTION # 4

Raising a number to a power p is the same as multiplying n by itself p times. Write a function called `power` that takes two arguments, a double value for n and an int value for p and return the result as double value. Use default argument of 2 for p, so that if this argument is omitted the number will be squared. Write the main function that gets value from the user to test power function.