Basics:

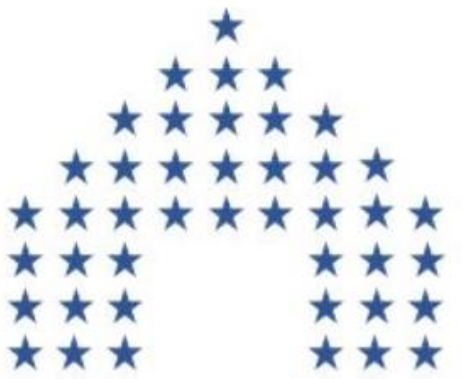
1. Write a program that take two integers from the user and print the results of this equation as float:
   Result = ((num1 + num2) * 3) /2– 10.
2. Write a program that print the biggest number between two integer numbers and also print which if those numbers are equal, not equal.
3. Write a program that take an integer from user and print if this number is odd or even.
4. Write a program that take an integer from user and print if this number is prime or not.
5. Write a program that take an integer from user and print if this is a power of 2 or not, e.g:

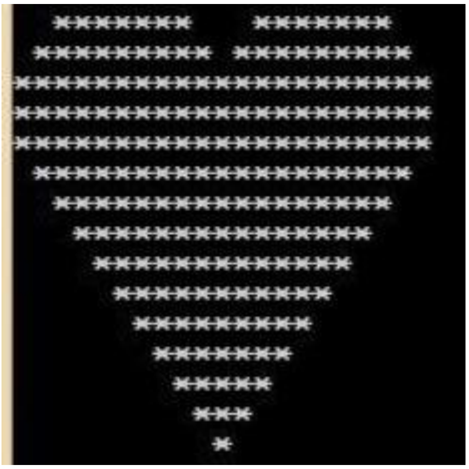
   16, 8, 4, 32, 64, 128 → yes

   25, 9, 49 → no
6. Write a program that take an integer and computes the factorial.
7. Write a program that reads a positive integer and checks if it can have a second square or not, e.g:

   25, 16, 64, 9, 49 → perfect square

   8, 5, 32 → not perfect square
8. Write a program that reads a student grade percentage and prints "Excellent" if his grade is greater than or equal 85, "Very Good" for 75 or greater; "Good" for 65, "Pass" for 50, "Fail" for less than 50.
9. Write a program that print all prime number from 1 to 100.
10. Write a program to display the following stars patterns.



- 

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

-

Bitwise:

1. Read value of the 4th bit in 8 bit binary number given by user.
2. Set value of the 5th bit (make it one) in 8 bit binary number given by user.
3. Clear value of the 2nd bit (make it Zero) in 8 bit binary number given by user.
4. Toggle value of the 6th bit (make it 0 if it is 1 and 1 if it is 0) in 8 bit binary number given by user.
5. Set last 2 bits of an 8 bit number given by user.
6. If you have 1 byte variable, write a code to swap bits #2 with #6 (swap mean put bit #2 in the location of bit #6 and bit #6 in location of bit #2)
7. Write a program that reads a positive integer and reverse all bits.

   x = **0x12345678** -> x will be **0x78563412**

8. Reverse all bit in an 8 bit binary number given by user e.g:

   00101011 → 11010100

   10100100 → 00100101

   11110110 → 01101111

9. Write a program that reads a positive integer and calculate the number of ones & zeros in the number
10. Write a program that reads a positive integer and calculate the number of consecutive zeros between two ones e.g: 0b1100011010011100000 1 the result will be 5
11. Write a code to multiply an input 1byte from user by 14 without using multiplication operator: e.g: user input is 2 → output will be 28
12. Write a code to check if number is odd or even using bitwise operators only.

Functions:

1. Write four ways to swap the value of two integer numbers.
2. Write a C Function that reads two integers and checks if the first is multiple of the second.
3. Write a C Function that checks if an integer is even or odd.
4. Write a C Function that checks if an integer is prime or not.
5. Write a C Function that return the addition or subtraction or multiplication or division for two numbers. The function should take the required operation and two numbers as arguments. It also should check that the input operation is one of those operation that mentioned before and if not it should return error. The function should be implemented
   using switch case.

String:

1. Write a C Function that converts any letter from lowercase to uppercase.
2. Write a C program to find length of string using pointers.
3. Write a C program to copy one string to another using pointers.

4. Write a C program to concatenate two strings using pointers.
5. Write a C program to compare two strings using pointers.
6. Write a C program to find reverse of a string using pointers.
7. C function to check if a string is mirror or not: char check_mirror(char* arry)

Arrays & pointers:

1. Write a code to detect if the processor is big endian or little endian
2. Write a C Function that take an array and its size, then sorting the element ascending.
3. Write a C Function that take an array and its size, then sorting the element descending.
4. Write a C Function that take an array and its size, return the max number.
5. Write a C Function that take an array and its size, return the min number.
6. Write a C Function that take an array and its size, return the average of its elements.
7. Write a C Function that take an array, an int and its size, return 0 if this integer is not found and the index of the int in the array if it is found.
8. Write a C program to find second largest element in an array.
9. Write a C Function that take an array and its size, then reverse all the element of the array
    - **void reverse(char* array, char size); e.g:** Input 1 5 6 3 4 5 → output 5 4 3 6 5 1
10. Write a C Function that take an array, its size and int* size of the new array, then remove the repeated value and return an array with no repeated values.
11. Write a C Function that take an array and its size, and return the most occurrence element in the array e.g: x[] = {1,2,3,8,6,5,3,9,3,4,3,7,2} the OUTPUT will be 3
12. Write a function which count the max number of consecutive elements in an array of 12 element:

    x[10] = {1, 1, 1, 5, 5, 5, 3, 3, 5, 5, 5, 5} and the user enter 5 then the output will be 4

    char consecutive(char *arr, char value);

Just write the below statements:

1. Pointer to int.
2. Pointer to pointer to int.
3. Pointer to array of 10 element of int.
4. Pointer to function which take 2 int and return void.
5. Array of 20 pointer to int.
6. Array of 10 pointer to function.
7. Pointer to constant int.
8. Pointer to constant pointer.

Write only the prototype of the function:

1. Function that take 2 dimensional array and return the address of 1 dimensional array.
2. Function that take two variables and return the biggest one.
3. Call back function that take pointer to long and pointer to function (the function return pointer to integer and take char).

Struct:

1. C Program to Store Information of Students Using Structure
2. C Program to Calculate Difference Between Two Time Periods
3. https://www.hackerrank.com/challenges/dynamic-array/problem