LOOP

1. Write a program to accept a number from the user and check whether it is prime or not. **Prime number** is a number that is greater than 1 and divided by 1 or itself. In other words, prime numbers can't be divided by other numbers than itself or 1.For example 2, 3, 5, 7, 11, 13, 17, 19, 23… are the prime numbers.

2. Write a program to find the factorial of 5. The **Factorial** function (symbol: !) means to multiply a series of descending natural numbers. Examples- 5! =5*4*3*2*1

3. Write a program to accept a number and display the **Sum of its digits**. Input: 213 Output: 6

4. Write a program to accept a number from user and check whether its **Palindrome or not**. A palindrome number is a number that is same after reverse. For example 121, 34543, 343, 131, 48984 are the palindrome numbers.

5. Write a program to accept a number from user and check whether its **Armstrong or not**. Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370,371 and 407 are the Armstrong numbers. 153 = (1*1*1)+(5*5*5)+(3*3*3) = 1 + 125 + 27 = 153

6. Write a program to display natural numbers from 1 to 10.

7. Write a program to enter a number and check whether the number is **Neon or not**. A neon number is a number where the sum of digits of square of the number is equal to the number. For example if the input number is 9, its square is 9*9 = 81 and sum of the digits is 9. i.e. 9 is a neon number.

8. Write a program to enter a number and check whether the number is **Twisted prime/ Emirp number or not.** If a number is prime, and its reverse number is also prime, then it is called Twisted prime number or emirp number. Example : 37 is prime number , and its reverse order is 73 and 73 is also prime.

9. Write a program to calculate and print the sum of odd numbers and sum of even numbers.

10. Write a program to find the sum of any ten natural numbers.

11. Write a program to accept two numbers and check whether they are **Twin prime or not.** Twin Prime numbers are a pair of numbers which are both prime and their difference is 2. (3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73)

12. Write a program to print a numbers from **Fibonacci series.** In Fibonacci series, next number is the sum of previous two numbers. 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.

13. Write a program to print a numbers from **Tribonacci series.** In Tribonacci series, next number is the sum of previous three numbers.0, 0, 1, 1, 2, 4, 7, 13, 24, 44

M:9239412412
TO KNOW MORE: www.facebook.com/pbainst & javapba.blogspot.com

LOOP

14. Write a program to accept two numbers and find the **Highest Common Factor** (HCF/GCD) and **Lowest common Multiple** (LCM) of numbers. [LCM= Product of two numbers / HCF]
Greatest Common Divisor- The highest number that divides exactly into two or more numbers.
It is the "greatest" thing for simplifying fractions!.

15. Write a program to display all the prime numbers from 1 to 100.

16. Write a program to accept a number and check number is **Perfect or not**. A number is called perfect if it is equal to the sum of its factor other than the number itself. Example: 6= 1 + 2 + 3

17. Write a program to print the Alphabets (A, B, C, D ….. Z)

18. Write a program to accept a number and find length of number. Example- 1234 Length- 4.

19. Write a program to print the **Even Series** & **Odd series.**

20. Write a program to **reverse or mirror image** the entered number. Input:12345 Output: 54321

21. Write a program to print the a numbers from **Square Numbers Series.**1, 4, 9, 16, 25, 36, 49

22. Write a program to print numbers from **Cube Numbers Series**.1, 8, 27, 64.

23. Write a program to accept a number and display the new number after **removing all zeros.**
Input-54001007 Output-5417

24. Write a program to accept a number and check whether the number is **Duck number or not.** A number is said to be Duck if the digit zero is (0) present in it. The program displays the message accordingly. Input: 5063 Output: Duck number. Input: 7453 Output: Not a Duck Number.

25. Write a program to accept a number and check number is **Prime palindrome or not.** First check the number is prime or not, if prime then check the palindrome or not. Example: 101, 131, 151, 181.

26. Write a program to display all the **Armstrong numbers from 1 to 1000.** Armstrong number is a number that is equal to the sum of cubes of its digits. For example 0, 1, 153, 370,371 and 407 are the Armstrong numbers. 153 = (1*1*1) + (5*5*5) + (3*3*3) = 1 + 125 + 27 = 153.

LOOP

27. Write a program to input a number and check whether it is a **Harshad Number or Niven Number or not.** In recreational mathematics, a Harshad number (or Niven number) is an integer (in base 10) that is divisible by the sum of its digits. The number 18 is a Harshad number, because the sum of the digits 1 and 8 is 9 (1 + 8 = 9), and 18 is divisible by 9 (since 18 % 9 = 0)
The number 1729 is a Harshad number, because the sum of the digits 1 ,7, 2 and 9 is 19 (1 + 7 + 2 + 9 = 19), and 1729 is divisible by 19 (1729 = 19 * 91). **Niven Numbers:** 18, 20, 21, 24, 27, 30, 36, 40, 42, 45, 48, 50, 54, 60, 63, 70, 72, 80, 81, 84, 90, 100, 102, 108, 110, 111, 112, 114, 117, 120, 126, 132, 133, 135.

28. Write a program to accept a number and **generate next 10 numbers**. For example: Input:5 Output- 6,7,8,9,10,11,12,13,14,15.

29. Write a program to **Generate Multiplication Table.** Input-5

Output:
```
5   *   1   =   5
5   *   2   =   10
5   *   3   =   15
5   *   4   =   20
5   *   5   =   25
5   *   6   =   30
5   *   7   =   35
5   *   8   =   40
5   *   9   =   45
5   *   10  =   50
```

30. Write a program to find **Factors** of a given number e.g. factors of 12 are 1, 2, 3, 4, 6, 12

31. Write a program print this series: **1, 12, 123, 1234, 12345.**

32. Write a Program to input a number and check whether it is a **SPECIAL NUMBER / Krishna Murthy number / Peterson number** or not.
Krishna Murthy Number / Special number: It is a number which is equal to the sum of the factorials of all its digits. For example: 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145

33. Write a program to input a number and check whether it is a **(Another Rule) SPECIAL NUMBER or not.** For example - 135...then 1+3+5=9 and 1*3*5*9=135

M:9239412412
TO KNOW MORE: www.facebook.com/pbainst & javapba.blogspot.com

LOOP

34. Write a Program to input a number and Print **Product of Digits** of a Number. For Example: Input:223. Output:12.

35. Write a program to input a number and check whether it is **Magic number or Not**.
A number is said to be a Magic number if the sum of its digits are calculated till a single digit is obtained by recursively adding the sum of its digits. If the single digit comes to be 1 then the number is a magic number.
For Example: 1) 55=5+5=10,1+0=1 2) 289=2+8+9=19, 1+9=10, 1+0=1

36. Write a Program to find the **Prime Factors** of a Number.
Prime factors of a number are those factors which are prime in nature and by which the number itself is completely divisible (1 will not be taken as prime number). For example- 24 Output- 2,3

37. Write a program print **Factors of N and their Count.** Output: Factors of 12 are 1 2 3 4 6 12
No. of factors = 6.

38. Write a program Print **Factors of N and their Sum.** For Example- 6.
Factor of 6= 1,2,3,6. (1+2+3+6=12). Output- 12

39. Write a program to enter any number and find the **sum of first and last digit** of the number.
Example: Input number: 1234 Output sum of first and last digit: 5

40. Write a Program to input a number and check whether it is a **Pronic Number or Heteromecic Number or not.**
Pronic Number : A pronic number, oblong number, rectangular number or heteromecic number, is a number which is the product of two consecutive integers, that is, n (n + 1). The first few pronic numbers are:0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 … etc.

41. Write a Program input a number and check whether it is a **Disarium Number or not.**
A number will be called DISARIUM if sum of its digits powered with their respective position is equal to the original number. For example 135 is a DISARIUM.
(Workings $1^1 + 3^2 + 5^3$= 135, some other DISARIUM are 89, 175, 518 etc)

42. Write a program print the series:
1 11 111 1111 11111

LOOP

43. Write a Program to input a number and check whether it is a **Fascinating Number or not.**
Fascinating Numbers: Some numbers of 3 digits or more exhibit a very interesting property. The property is such that, when the number is multiplied by 2 and 3, and both these products are concatenated with the original number, all digits from 1 to 9 are present exactly once, regardless of the number of zeroes.
Consider the number 192,
192 x 1 = 192
192 x 2 = 384
192 x 3 = 576
Concatenating the results: 192384576. [It could be observed that **'192384576'** consists of all digits from 1 to 9 exactly once. Hence, it could be concluded that 192 is a Fascinating Number.]
Some examples of fascinating Numbers are: 192, 219, 273, 327, 1902, 1920, 2019 etc.

44. Write a program to display all the **Automorphic numbers from 1 to 1000**.
An automorphic number is a number whose square ends in the same digits as the number itself.
For example: $6^2 = 36, 5^2 = 25, 25^2 = 625$

45. Write a Program to input a number and check whether it is a **Flappy Number or not**.
Flappy number- A **five digit** number is called Flappy if product of its last two digits is 32 and sum of all five digits is 36. E.g. 79848. (7+9+8+4+8)=36.(4*8)=32

46. Write a program to accept a number and display the result in its **Binary equivalent**.
Input :15 Output: 1111

47. Write a program to accept a binary number (base 2) and convert it into its **Decimal equivalent** (base 10). Input:1111 Output: 15

48. Write a program to enter a number containing **three digits or more**. Arrange the digits of the entered number in **ascending order** and display the result.
Input: Enter the number 4972 Output: 2,4,7,9

49. Write a program to accept a number and check whether the number is **present** in the **Fibonacci series or not**. The program displays the message accordingly.
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Input number :55
Output: 55 is present in Fibonacci series.

LOOP

50. Write a program to accept a number and check whether the number is **present** in the **Tribonacci series or not**. The program displays the message accordingly.
0, 0, 1, 1, 2, 4, 7, 13, 24, 44. Input number :43
Output: 43 is not present in Tribonacci series.

51. Write a program find the **smallest digit** of integer that is input.
Input: 9524 Output: Smallest digit is 2.

52. Write a program find the **greatest digit** of integer that is input.
Input :9524 Output: Greatest digit is 9.

53. Write a program covert **Binary to Octal**. Input: 1110 Output: 16

54. Write a program covert **Octal to Binary.** Input: 16 Output: 1110

55. Write a program covert **Binary to Hex**. Input: 1110 Output: E

56. Write a program covert **Hex to Binary.** Input: E Output:1110

57. Write a Program input a number and check whether it is a **Kaprekar number or not**.
A positive whole number 'n' that has 'd' number of digits is squared and split into two pieces, a right-hand piece that has 'd' digits and a left-hand piece that has remaining 'd' or 'd-1' digits. If the sum of the two pieces is equal to the number, then 'n' is a Kaprekar number. The first few Kaprekar numbers are: 9, 45, 297….
Example 1: 9
$\mathbf{9^2 = 81}$, right-hand piece of 81 = 1 and left hand piece of 81 = 8.
Sum = 1 + 8 = 9, i.e. equal to the number. Hence, 9 is a Kaprekar number.
Example 2: 45
$\mathbf{45^2 = 2025}$, right-hand piece of 2025 = 25 and left hand piece of 2025 = 20
Sum = 25 + 20 = 45, i.e. equal to the number. Hence, 45 is a Kaprekar number.
Example 3: 297
$\mathbf{297^2 = 88209}$, right-hand piece of 88209 = 209 and left hand piece of 88209 = 88
Sum = 209 + 88 = 297, i.e. equal to the number. Hence, 297 is a Kaprekar number.

58. Write a program **find the sum of series**.
1 + 4 + 9 + 61 + 52 + 63 + 94 + ……. N

LOOP

59. **The International Standard Book Number (ISBN)** is unique numeric book identifier which is printed on every book. The ISBN is based upon a 10 digit code. The ISBN is legal if : 1xdigit_1 + 2xdigit_2 + 3xdigit_3 + 4xdigit_4+ 5xdigit_5 + 6xdigit_6 + 7xdigit_7 + 8xdigit_8 +9xdigit_9 + 10xdigit_10 is divisible by 11.
Example: For an ISBN **1401601499**
Sum = 1x1+2x4+3x0+4x1+5x6+6x0+7x1+8x4+9x9+10x9 = **253** which is **divisible by 11.**
Write a program to : 1) Input the ISBN code as a **10 digit Integer.**
2) If the ISBN is not a 10 digit integer, output the message, "Illegal ISBN" and terminate the program
3) If the number is 10 digits, extract the digits of number and compute the sum as explained above. If the sum is divisible by 11, output message, "Legal ISBN". If the sum is not by 11, output the message, "Illegal ISBN".
More ISBN- 0201530821 (Valid), **035680324** (Invalid), **0231428031** (Invalid)

60. Write a program to enter a number containing three digits or more. Arrange the digits of entered number in **descending order** and display the result.
Input : Enter a number : 5296 Output: 9,6,5,2

61. Write a program to check whether two numbers are **Amicable or not.**
Amicable Number: If two numbers are such that the sum of the perfect divisors of one number is equal to the other number and the sum of the perfect divisors of the other number is equal to the first number, then the numbers are called Amicable Numbers. **Example: 220 and 284.**
The divisors of 220 are: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110
The divisors of 284 are:1, 2, 4, 71, 142
Now, adding the divisors of 220: 1+2+4+5+10+11+20+22+44+55+110 = 284, which is equal to number b=284
Now, adding the divisors of 284: 1+2+4+71+142 = 220, which is equal to number a=220
Hence, the numbers 220 and 284 are amicable numbers.

62. Write a program to accept 10 different numbers and display the **greatest and smallest** of the numbers.

63. Write a program to check whether the number is **Happy Number or not.**
Happy Number: A happy number is a number in which the eventual sum of the square of the digits of the number is equal to 1. Example:28 $=2^2 + 8^2 = 4 + 64 = 68$
$68 = 6^2 + 8^2 = 36 + 64 = 100$
$100 = 1^2 + 0^2 + 0^2 = 1 + 0 + 0 = 1$ Hence, **28** is a happy number.
Example:12=$1^2 + 2^2 = 1 + 4 = 5$ Hence, **12** is not a happy number.

LOOP

64. Write a program to print the 10 numbers of the **Pell series**. **Pell series** is series of numbers which starts with 1 and then 2 and goes on. The next term is sum of double of precedent number and the number before the precedent number. Example: 5' is the sum of double of '2' and '1' , i.e. , (1 + (2x2)). This way the series goes on. **1 2 5 12 29 70 169 408 985 2378**

65. Write a program to print the 10 numbers of **Lucas series.** Lucas numbers are similar to Fibonacci numbers. Lucas numbers are also defined as the sum of its two immediate previous terms. But here the first two terms are 2 and 1 where as in Fibonacci numbers the first two terms are 0 and 1 respectively. **2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123**

66. Write a program find the **average of digits**. Input: 5241 Output: 3

67. Write a program **remove all even digits** from it. Example: Input:1234 Output:13

68. Write a program **remove all odd digits** from it. Example: Input:1234 Output:24

69. Write a program to input an integer and check whether it is **perfect, abundant or deficient number**. If sum of the factors excluding if self is equal to that number it is perfect, if greater than that number it is abundant and if less than that number it is deficient number.

70. Write a program to input a number and check whether all **digits in it are same or not**. For Example: 555 Output All digit same

71. Write a program to input 5 integers and check whether all the entered **numbers are same or not.**
Input: 4 5 6 8 9 Output: Not same numbers
Input: 6 6 6 6 6 Output: Same numbers.

72. Write a program to find the **sum of all 3 digits odd numbers** which are **multiples of 5.**

73. Write a program to input 20 numbers and find the **average of those numbers which are even.**

74. Write a program to input a number and check whether it is a **Unique Number or not.** A Unique number is a positive integer (without leading zeros) with no duplicate digits. For example 7, 135, 214 are all unique numbers whereas 33, 3121, 300 are not.

75. Write a program to input a number and check whether it is a **Trimorphic or not.** A number n is called trimorphic if n^3 ends in n. For Example $49^3 = 117649$

LOOP

76. Write a program to accept a number and check and display whether it is a **spy number or not.** (A number is spy if the sum its digits equals the product of the digits.) Example: consider the number 1124. Sum of digits = 1+1+2+4=8 Product of the digits = 1*1*2*4=8

77. Write a program to input an integer and check whether all are **prime digits or not.**

78. Write a program to print sum of negative numbers, sum of positive even numbers and sum of positive odd numbers from a list of numbers (N) entered by the user. The list terminates when the user enters a zero.

79. Write a program to calculate the sum of n natural numbers.

80. Write a program to input a number and check whether it is a **Smith number or not.** Smith number is such a number the sum of whose digits equals the sum of digits of its prime factors. [Smith Number is a composite number in which the sum of its digits is equal to the sum of the digits of all its prime factors.] **For Example:** 378 is a Smith number as the sum of the digits of 378 are 3+7+8=18. The prime factors of 378 are: 2, 3, 3, 3, 7 (Sum= 2+3+3+3+7=18)
Similarly 22 is a Smith Number as the sum of digits are 2+2=4. The prime factors of 22 are 2 and 11 (Sum=2+(1+1) =4). **Another example** includes 27,58,85,94,121,166,202,265 etc.

81. Write a program to input a number and check whether it is an **Evil Number or not.** Evil Number- An Evil number is a positive whole number which has even number of 1's in its binary equivalent.
INPUT: 15 BINARY EQUIVALENTS: 1111 NO. OF 1's: 4 OUTPUT: EVIL NUMBER
INPUT: 26 BINARY EQUIVALENTS: 11010 NO. OF 1's: 3 OUTPUT: NOT AN EVIL NUMBER

82. Write a program to input a number and check whether it is a **Keith Number or not.** A Keith Number is an integer N with 'd' digits with the following property: If a Fibonacci-like sequence (in which each term in the sequence is the sum of the 'd' previous terms) is formed, with the first 'd' terms being the decimal digits of the number N, then N itself occurs as a term in the sequence.
For example, 197 is a Keith number since it generates the sequence 1, 9, 7, 17, 33, 57, 107, 197, ……..Some Keith numbers are: 14 ,19, 28 , 47 , 61, 75, 197, 742, 1104, 1537……

83. Write a program to display all the **Magic numbers from 1 to 1000.**

84. Write a program to display all the **Kaprekar numbers from 100 to 1000.**

85. Write a program to display all the **Happy numbers from 1 to 1000.**

LOOP

86. Write a program to print **only even numbers from Fibonacci series** [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.]. Output: 2, 8, 34 …

87. Write a program to print **only odd numbers from Fibonacci series** [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.].

88. Write a program to print **only prime numbers from Fibonacci series** [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89.]. Output: 2, 3, 5, 13…..

89. Write a program to print **only prime numbers from Tribonacci series [**0, 0, 1, 1, 2, 4, 7, 13, 24]

90. Write a program to display **all 5 digit palindromic prime numbers**.

91. Write a program to accept a number and **find out the sum of the squares of the digits of the number.** Input:458 Output: 105

92. Write a program to accept a number and **find the sum of its prime digits**. Input:623 Output:5

93. Write a program to **Circular Prime or not.** A Circular Prime is a prime number that remains prime under cyclic shifts of its digits. When the leftmost digit is removed and replaced at the end of the remaining string of digits, the generated number is still prime. The process is repeated until the original number is reached again. Examples: 131, 197, 1193 etc.
INPUT: 1193
OUTPUT:
1193
1931
9311
3119
1193 IS A CIRCULAR PRIME

94. Write a program print **Floyd's triangle.** The Floyd's triangle (named after Robert Floyd) algorithm is a right-angled triangular array of natural numbers. It is defined by filling the rows of the triangle with consecutive numbers, starting with the number one in the top left corner.
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

LOOP

95. Write a program to print the **possible consecutive number combinations**.
A positive natural number, (for e.g. 27), can be represented as follows:
2+3+4+5+6+7
8+9+10
13+14
Where every row represents a combination of consecutive natural numbers, which add up to 27.
Write a program which inputs a positive natural number N and prints the possible consecutive number combinations, which when added give N.

SAMPLE DATA

INPUT: N = 9 **OUTPUT:** 4 + 5 2 + 3+ 4	**INPUT:** N = 15 **OUTPUT:** 7 +8 1 +2+ 3+ 4+ 5 4 +5+ 6	**INPUT:** N = 21 **OUTPUT:** 10+ 11 1+ 2+ 3+ 4+ 5+ 6 6+ 7+ 8

96. Write a Program to print all the **Twin Prime numbers within a given range.**
Note: Twin Prime numbers are a pair of numbers which are both prime and their difference is 2.
Example: Twin Prime numbers in the range 1 to 100 are:
(3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73)
Output: Enter the lower range: 1
Enter the upper range: 200
The Twin Prime Numbers within the given range are:
(3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73) (101,103) (107,109) (137,139) (149,151) (179,181) (191,193) (197,199)

97. Write a program to find the **frequency of the digits** in a number.
Input: 15522412
Output: Frequency of 1 is=2, Frequency of 2 is=3, Frequency of 4 is=1, Frequency of 5 is=2

98. Write a program to input a natural number less than 1000 and **display it in words.**
[Note we have solved the program for numbers in the range [1-9999]

Input: 29 Output: TWENTY NINE	Input: 17001 Output: OUT OF RANGE
Input: 119 Output: ONE HUNDRED AND NINETEEN	Input: 500 Output: FIVE HUNDRED

LOOP

98. Write a program to input a number. Display the product of successors of even digits of the number entered by user. **Input:** 2745
Output: The even digits are 2 and 4
The product of successor of even digits is 3*5=15

99. A "sum-product" number is an integer that is equal to the sum of its digits times the product of its digits. Write a program to input a number and check whether it is a "Sum-Product" number or not. Display an appropriate message accordingly. Example: 1+2+3=1*2*3
Input: 123. **Output:** It is a Sum-Product number.

100. The **International Mobile Station Equipment Identity** or **IMEI** is a number, usually unique, to identify mobile phones, as well as some satellite phones. It is usually found printed inside the battery compartment of the phone.
The IMEI number is used by a GSM network to identify valid devices and therefore can be used for stopping a stolen phone from accessing that network.
The IMEI (15 decimal digits: 14 digits plus a check digit) includes information on the origin, model, and serial number of the device.
The IMEI is validated in three steps:

1. Starting from the right, double every other digit (e.g., 7 becomes 14).
2. Sum the digits (e.g., 14 → 1 + 4).
3. Check if the sum is divisible by 10.

For Example:
If input is IMEI = **490154203237518**

IMEI	4	9	0	1	5	4	2	0	3	2	3	7	5	1	8
Double every other digits	4	18	0	2	5	8	2	0	3	4	3	14	5	2	8
Sum digits	4 + (1 + 8) + 0 + 2 + 5 + 8 + 2 + 0 + 3 + 4 + 3 + (1 + 4) + 5 + 2 + 8 = 60														

Since, 60 is divisible by 10, hence the given IMEI number is Valid.
Design a program to accept a fifteen digit number from the user and check whether it is a valid IMEI number or not. For an invalid input, display an appropriate message.
To Check IMEI NO dial *#06# from your mobile.

LOOP

101. Write a program to input all two digit numbers between 10 and 99 (both inclusive).The program displays only those numbers whose the unit digit is twice the tens digit. Sample output 12, 24, 36

102. Write a program to accept two numbers and check whether they are **Sexy prime or not.** In mathematics, Sexy Primes are prime numbers that differ from each other by six. For example, the numbers 5 and 11 are both sexy primes, because they differ by 6. If p + 2 or p + 4 (where p is the lower prime) is also prime.
Given a range of the form [L, R].The task is to print all the sexy prime pairs in the range.

Input : L = 6, R = 59 **Output :** (7, 13) (11, 17) (13, 19) (17, 23) (23, 29) (31, 37) (37, 43) (41, 47) (47, 53) (53, 59)	**Input :** L = 1, R = 19 Output : (5, 11) (7, 13) (11, 17) (13, 19)

103. Write program to calculate power of number without using pre define function or math functions.

104. A **Composite Magic number** is a positive integer which is composite as well as a magic number.
Composite number: A composite number is a number that has more than two factors. For example: 10, factors are: 1, 2, 5, 10
Magic number: A magic number is a number in which the eventual sum of the digits is equal to 1. For example: 28=2+8=10=1+0=1
Accept two positive integer's m and n, where m is less than n as user input. Display the number of Composite Magic integers that are in the range between m and n (both inclusive) and output them along with the frequency, in the format specified below.

Input: m=10 n=100 **Output:** THE COMPOSITE INTEGERS ARE: 10, 28, 46, 55, 64, 82, 91, 100 FREQUENCY OF COMPOSITE MAGIC INTEGERS IS: 8	**Input:** m=1200 n=1300 **Output:** THE COMPOSITE INTEGERS ARE: 1207, 1216, 1225, 1234, 1243, 1252, 1261, 1270, 1288 FREQUENCY OF COMPOSITE MAGIC INTEGERS IS: 9	**Input:** m=120 n=99 **Output:** INVALID INPUT

105. Write a program to check whether a given number is an **ugly number** or not. Ugly numbers are positive numbers whose only prime factors are 2, 3 or 5. The sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 shows the first 10 ugly numbers. Note: 1 is typically treated as an ugly number. Example: 18. Factors of 18 = 2 * 3 * 3. Prime factors: 2, 3. 18 is ugly number. But number =13. factors of 13: 1, 13. Prime factors: 13. So 13 is not an ugly number.

LOOP

106. Write a program to check whether a given number is **Abundant or not**. A number n is said to be Abundant Number if sum of all the proper divisors of the number denoted by sum(n) is greater than the value of the number n. And the difference between these two values is called the abundance. Example Divisor of 12: 1, 3, 4, 6, 2 Sum of divisor (1 + 3 + 4 + 6 + 2) =16 if 16>12 then 12 is abundant no.

107. Write a program in C to check whether a number is **Lychrel number** or not. Lychrel Number is a natural number that cannot form a palindrome through the iterative process of repeatedly reversing its digits and adding the resulting numbers.
Input: 56 Output: 56 is lychrel: false Explanation: 56 becomes palindromic after one iteration: 56 + 65 = 121.
Input:196 Output: 196 is lychrel: true Explanation: 196 becomes palindromic after 19 iterations:
196 + 691 = 887
887 + 788 = 1675
1675 + 5761 = 7436
7436 + 6347 = 13783
13783 + 38731 = 52514
....
16403234045 + 54043230461
70446464506 + 60546464407

108. Write a program to find the Abundant numbers (integers) between 1 to 1000.

109. Write a program to generate and show the first 15 **Narcissistic decimal numbers**. Narcissistic Number is a number that is the sum of its own digits each raised to the power of the number of digits.
Input: 153 Explanation: 1^3+5^3+3^3=153
Input: 1634 Explanation: 1^4+6^4+3^4+4^4=1634

110. Write a program to display the first 10 **Catalan numbers.** In combinatorial mathematics, the Catalan numbers form a sequence of natural numbers that occur in various counting problems, often involving recursively-defined objects. They are named after the Belgian mathematician Eugène Charles Catalan.

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!\,n!} = \prod_{n=2}^{n} \frac{n+k}{k} \quad \text{for } n \geq 0.$$

The first 10 catlan numbers are :

1 1 2 5 14 42 132 429 1430 4862