1. Write a C program to read a number from user and check whether given number is a prime palindrome or not. A number is a palindrome if the number is the same as its reverse for e.g. 23432 is a palindrome but 2345 is not. A number is prime palindrome if it is both a prime number and a palindrome. For e.g., 23432 is not a prime palindrome, but 131 is a prime palindrome.
2. Write a C program to find Armstrong Numbers within *m* and *n* where *m, n* are user inputs. A number n is called an Armstrong number if the sum of its digits’ to the power k is equal to n, where k is the number of digits in n. For e.g. 407 is an Armstrong number because 407 contains 3 digits and 43+03+73 = 407. 1634 is also an Armstrong number because 1634 contains 4 digits and 14+64+34+44 = 1634.  
   Sample input/output :  
   Enter starting and ending range: 1 2000  
   Armstrong numbers between 1 and 2000 are:  1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634,
3. Calculate the sum of the following series, where and are provided as user inputs.
4. Write a program in C to print the n-th prime number. For e.g. 10th prime is 29.
5. Write a C program to convert a decimal number into a binary number.
6. Write a program in C to display the n-th Fibonacci number. Note that i-th Fibonacci number is the sum of the previous two Fibonacci numbers i.e. it is the sum of (i-1)-th and (i-2)-th Fibonacci numbers. Assume that the 1st two Fibonacci numbers are 0 and 1, respectively. So Fibonacci series is: 0, 1, 2, 3, 5, 8, 13, .....  
   Sample input/output :  
   Enter n: **7**

7-th Fibonacci number is: 13

1. Write separate C programs to print the following patterns for n lines (n is input) using nested loop:

|  |  |  |
| --- | --- | --- |
| 1  234  56789  0123456  789012345 | 0         01        010        0101       01010 | \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* |