PROGRAM 2:

QUESTION:

Create a class TwistedPrime having following specifications. Class Name: TwistedPrime

Data Methods:

int n – to store the number

Member Methods:

TwistedPrime(int nn) – to initialize n with nn

int prime – to check weather the number is prime or not

int TwistedPrimeCheck – to check weather the number is Twisted Prime or not

void display() - to display the print statement

Implement the main method to create the object of the class and call the above methods properly.

ALGORITHIM:

1. Begin.
2. Create a class named `TwistedPrime`.
3. Declare an instance variable `n` to store the input number.
4. Create a parameterized constructor `TwistedPrime(int nn)` to initialize the instance variable `n` with the input value `nn`.

Algorithm for the `Prime(int n)` Method:

1. Begin.
2. If `n` is less than or equal to 1, return 0.
3. Iterate from `i` = 2 to the square root of `n`: 1. If `n` is divisible by `i`, return 0.
4. Return 1.
5. End.

Algorithm for the `TwistedPrimeCheck(int n)` Method:

1. Begin.
2. If the result of `Prime(n)` is 0, return 0.
3. Initialize variables `reversed` and `temp` to 0 and `n`, respectively.
4. Iterate while `temp` is not 0:
   1. Extract the last digit of `temp` as `digit`.
   2. Build the `reversed` number by multiplying it by 10 and adding `digit`.
   3. Update `temp` by dividing it by 10.
5. Return the result of `Prime(reversed)`.
6. End.

Algorithm for the `display()` Method:

1. Begin.
2. Call the `TwistedPrimeCheck(n)` method to determine if the given number `n` is a twisted prime number.
3. If the result is 1, print "The given number is a twisted prime number."
4. Otherwise, print "The given number is not a twisted prime number."
5. End.

Algorithm for the `main` Method:

1. Begin.
2. Create a `Scanner` object `sc` to read input from the user.
3. Display a prompt asking the user to enter a number.
4. Read the entered number using the `nextInt()` method of the `Scanner` object and store it in the variable `num`.
5. Create an instance of the `TwistedPrime` class using the input number `num`.
6. Call the `display()` method on the created instance to determine and display if the number is a twisted prime or not.
7. End.

VARIABLE DESCRIPTION TABLE

|  |  |  |
| --- | --- | --- |
| Variable Name | Variable Type | Variable Description |
|  |  |  |
| n | int (instance) | An instance variable of the |
|  |  | class `TwistedPrime` to store |
|  |  | the input number. |
|  |  |  |
| nn | int (parameter) | Parameter for the constructor, |
|  |  | representing the input |
|  |  | number. |
|  |  |  |
| i | int | Loop variable used for |
|  |  | iteration in the `Prime(int n)` |
|  |  | method. |
|  |  |  |
| sum | int | Variable to calculate the sum |
|  |  | of proper divisors of `n` in the |
|  |  | `Prime(int n)` method. |
|  |  |  |
| sc | Scanner | A `Scanner` object used to |
|  |  | read input from the user in the |
|  |  | `main` method. |
|  |  |  |
| number | TwistedPrime (class) | An instance of the |
|  |  | `TwistedPrime` class created |
|  |  | using the input number |
|  |  | `num`, used to perform |
|  |  | operations and interact with |
|  |  | the class methods. |
|  |  |  |