PROGRAM 3

QUESTION:

Create a class PalPrime having the following specifications:

Class Name: PalPrime

Data Method:

int n – to store the number.

Member Methods:

PalPrime(int nn) – to initialize n with nn.

int prime(int n) – to check weather the number is prime or not. int reverseNumber(int n) – to return the reverse of the given number.

void check() - to check if the reversed number and original number are same and display the print statement.

Write a main method to create the object of the class and call all the above methods properly.

ALGORITHM:

1. Start
2. Define a class named `PalPrime`.
3. Declare an integer variable `n` as a class member.
4. Define a constructor `PalPrime(int nn)`:

- Initialize the class member `n` with the value of `nn`.

Algorithm for `Prime` method:

1. Start
2. Input: An integer `n`.
3. If `n` is less than or equal to 1, return 0 (indicating not prime).
4. Loop `i` from 2 to the square root of `n`:
   * If `n` is divisible by `i`, return 0 (indicating not prime).
5. Return 1 (indicating prime).
6. End

Algorithm for `reverseNumber` method:

1. Start
2. Input: An integer `n`.
3. Initialize an integer variable `reverse` as 0.
4. While `n` is not equal to 0:
   * Calculate the last digit of `n` using `n % 10` and store it in `digit`.
   * Update `reverse` as `reverse \* 10 + digit`.
   * Update `n` as `n / 10`.
5. Return the `reverse` value.
6. End

Algorithm for `check()` method:

1. Start `check()` method.
2. If the result of `Prime(n)` is 0, indicating that `n` is not a prime number:
   * Print "Given number is not prime number."
3. Calculate the reverse of the number using the `reverseNumber(n)` method and store it in the variable `reverse`.
4. If the original number `n` is equal to the reverse:
   * Print "Given number is a pal prime number."
5. Else:
   * Print "Given number is not a pal prime number."
6. End `check()` method.

Algorithm for `main` method:

1. Start `main` method.
2. Create a `Scanner` object named `sc` to read input.
3. Print "Enter a number."
4. Read an integer input from the user and store it in the variable `num`.
5. Create an instance of the `PalPrime` class named `number` by calling the constructor with the input `num`.
6. Call the `check()` method on the `number` instance to perform the prime and palindrome checks.
7. End `main` method.

VARIABLE DESCRIPTION TABLE

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
|  |  |  |
| n | int | Stores the input |
|  |  | number for processing. |
|  |  |  |
| nn | int | Input parameter for the |
|  |  | constructor to initialize |
|  |  | `n`. |
|  |  |  |
| i | int | Loop variable used for |
|  |  | prime number |
|  |  | checking. |
|  |  |  |
| reverse | int | Stores the reversed |
|  |  | number during the |
|  |  | reverse operation. |
|  |  |  |
| digit | int | Stores the last digit of |
|  |  | the number during |
|  |  | reverse process. |
|  |  |  |
| sc | Scanner | Reads input from the |
|  |  | user. |
|  |  |  |
| num | int | Stores the input number |
|  |  | read from the user. |
|  |  |  |
| number | PalPrime | Instance of the |
|  |  | `PalPrime` class to |
|  |  | perform checks. |
|  |  |  |