```
private bool PrimeCheck(int num) {
    return Array.BinarySearch(listofPrimes, num) >= 0;
}
 int answer = 0;
int[] listofPrimes = PrimeAppend(PrimeCheckLimit);
int count = 0;
int i = 4;

while (count < 11) {
        int fwdTrunk = listofPrimes[i];
        int revTrunk = 0;
        int number = 1;
        bool TrunkCheck = true;
        while (fwdTrunk > 0 && TrunkCheck) {
            revTrunk += number * (fwdTrunk % 10);
            TrunkCheck = PrimeCheck(revTrunk) && PrimeCheck(fwdTrunk);
            fwdTrunk /= 10;
            number *= 10;
    }

    if (TrunkCheck) {
        count++;
        answer += listofPrimes[i];
    }
    i++;
}
```