1. Write a C program to print all odd numbers from 1 to n (n is user input) using for loop.

```
#include <stdio.h>

void main()
{
   int i, n;

   printf("Print odd numbers till: ");
   scanf("%d", &n);    //Read the upper limit
   printf("All odd numbers from 1 to %d are: \n", n);

for(i=1;  i<=n;  i++)
  {
    if(i%2!=0)       // Check if the number is odd
    {
       printf("%d\n", i);
    }
  }
}
```

2. Write a C program to read an integer from user and count the total number of digits in it.

```
#include <stdio.h>

void main()
{
    lint num, count = 0;

    printf("Enter any integer: ");
    scanf("%d", &num);

    while(num != 0)
    {
        count++;
        num /= 10;
    }

    printf("Total digits:
%d",count);
}
```

3. Write a C program to read an integer from user and count the total number of <u>nonzero</u> digits in it.

```
#include <stdio.h>

void main()
{
    lint num, count = 0;

    printf("Enter any integer: ");
    scanf("%d", &num);

    while(num != 0)
    {
        //current digit is num%10
        if(num%10 != 0)
           count++;
        num /= 10;
    }

    printf("Total nonzero digits:
%d", count);
}
```

Try yourself 2: Write a C program to read an integer and compute the sum of digits in it. Sample Input/Output:
Enter any integer: **452**
Sum of digits in 452: 11

4. Write a C program to find all the factors of a number.

```
#include <stdio.h>

void main()
{
   int i, num;

   printf("Enter any number to find its factor: ");
   scanf("%d", &num);

   printf("All factors of %d are: \n", num);

   for(i=1; i<=num/2; i++)
   //highest possible factor of num is: num/2
   {
    // If num is exactly divisible by i, then i is a factor of num
      if(num%i==0)
      {
         printf("%d\n",i);
      }
   }
}
```

Try yourself 3: Write a C program to print all the odd factors of a given number.

5. Write a C program that can be used to find whether a number is a prime number or not

```
#include <stdio.h>

void main()
{
    int i, n, isPrime = 1; //isPrime is used as a "flag/indicator". Initially we assume
                           //that n is prime and we set isPrime=1 to indicate this. If we
                       //later find that n is not really a prime, then we will set isPrime=0
    printf("Enter any number to check if it is prime: ");
    scanf("%d", &n);
    for(i=2; i<=n/2; i++)      //highest possible factor of n is: n/2; so continue as long as i <= n/2
    {
         // If n has a factor other than 1 and itself then it is not prime
        if(n%i==0) //if i is a factor of n (i.e., if n is divisible by i), where i
                   //varies from 2 to n/2, then n is not prime
        {
            isPrime = 0; //here we set isPrime = 0 to indicate that n is not a prime
            break;//go to the first statement after this for loop (break out of loop)
        }
    }

    if(isPrime == 0)//If isPrime==0 then n is divisible by a value of i; so n is not prime
    {
        printf("\n%d is not a prime number", n);
    }
    else // If isPrime==1 then n is NOT divisible by ANY value of i; so n is a prime no.
    {
        printf("\n%d is a prime number", n);
    }
}
```

Exercise Problems:

1. **Write a C program to read an integer from user and output its last and first digit. Hint: Like practice 2 & 3, reduce the number by dividing it (by 10) again and again (in a loop) until you reach the first digit.**
2. **Write a C program to check whether an input number is a perfect number or not. A perfect number is a positive integer which is equal to the sum of its proper positive factors. For e.g. 6 is a perfect number; because proper factors of 6 are 1, 2, 3 and 1+2+3 = 6. Also, 28 is a perfect number since sum of its factors = 1+2+4+7+14 = 28.**

Assignment Problems:

1. **Write a C program to enter any number from user and find the reverse of a given number using loop. Sample input/output (bold ones are user inputs):**
   Enter a number: **2345**
   Reverse of 2345 is: 5432
2. **Write a C program to read a number from user and check whether given number is a 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.**
3. **Write a C program to read any integer from user and compute the reverse of given number. Also output whether the reverse number is prime or not. Sample Input/Output:**
   Enter any integer: **4521**
   Reverse number is: 1254. It is not a prime number.
4. **Write a C program to compute the sum of digits of an input number and check if this sum is a prime or not. Sample Input/Output:**
   Enter any integer: **2821**
   Sum of its digits = 13. It is a prime number.