## 1:Write a C program that will print all integers between 1 and n. You must use while loop.

```
l = int(input())    # l = 3
n = int(input())    # n = 5
if l<n:      #if 3<5:
  while l<=n:      # while 3<=5
    print(l,end=' ')               #output: 3
    l+=1
  # while l>n:
  #   break
elif l>n:
  while l>=n:
    print(n,end=' ')
    n+=1
```

```
5
3
3 4 5
```

## 2:Write a C program that will print the first n odd numbers. You must use while loop

```
n = int(input())
i = 1
j = 1
while i<=n:
  print(j,end=' ')
  i+=1
  j+=2
```

```
5
1 3 5 7 9
```

## 3:Write a C program that will calculate the sum of the following series up to its nth term:

## 1.3+2.6+4.9+8.12+....

```
#n = int(input())
n =5
i = 1
j = 1
k = 3
total = 0
while i<=n:
  total = total +(j)*(k)
  j*=2
  k+=3
  i+=1
print(total)
```

```
387
```

```
n = 5
idx = 0
total = 00
while idx<n:    # 4<5
    a = 2**idx  # a = 2**4. a= 1 2 4 8 16
    b = 3*idx+1 # b = 3*5   b= 3 6 9 12 15
    count = a*b # count = 1.3+2.6+4.9+8.12+16.15
    total = total + count
    idx = idx + 1
    print(a,b)
    print(total)
print(total)
```

```
1 1
1
2 4
9
4 7
37
8 10
117
16 13
325
325
```

## 4: Write a C program that will take as input an integer n, and find out if it is prime or not.

```
number = int(input('Enter any number: '))              # number = 6
f = 0                                                  # f = 0
if number == 1 or number ==0:                          # False because number is not equal to 1 or 0
  f = 1                                                # not happening
for i in range(2,number):                              # if the any divisor before the number return a 0 remainder it is not a prime
  if number%i == 0:
    f=1
if f==1:
  print('Number is not prime')
else:
  print('number is prime')
```

```
Enter any number: 25
Number is not prime
```

```
number = int(input('Enter any number: '))
f = 0
i = 2
if number == 1 or number == 0:                          # False because number is not equal to 1 or 0
  f = 1
while i<number:
  if number%i == 0:
    f =1
  i+=1



if f==1:
  print('Number is not prime')
else:
  print('number is prime')
```

```
Enter any number: 29
number is prime
```

## 5:Write a C program that will take as input two integers, and calculate their GCD.

```
a = int(input())
b = int(input())
GCD_a = []
GCD_b = []
GCD = []
for i in range(1,a+1):
  if a%i == 0:
    GCD_a.append(i)
for i in range(1,b+1):
  if b%i == 0:
    GCD_b.append(i)
print(GCD_a)
print(GCD_b)
for i in GCD_a:
  for j in GCD_b:
    if  i == j:
      GCD.append(j)
print(max(GCD))
```

```
11
13
[1, 11]
[1, 13]
1
```

```
a = int(input())
b = int(input())
GCD_a = []
GCD_b = []
GCD = []
i = 1
while i<=a:
  if a%i == 0:
    GCD_a.append(i)
  i+=1
print(GCD_a)
i = 1
while i<=b:
  if b%i == 0:
    GCD_b.append(i)
  i+=1
i = 0
while i < len(GCD_a):
    j = 0
    while j < len(GCD_b):
        if GCD_a[i] == GCD_b[j]:
            GCD.append(GCD_b[j])
        j += 1
    i += 1


print(max(GCD))
```

```
11
13
[1, 11]
1
```

## 6:Write a C program that will take as input two integers, and calculate their LCM.

```
a = int(input())
b = int(input())
a1 = a
b1 = b
LCM_a = []
LCM_b = []
LCM = []
for i in range(1,11):
  a1+=a
  LCM_a.append(a1)
for i in range(1,11):
    b1+=b
    LCM_b.append(b1)
print(LCM_a)
print(LCM_b)
for i in LCM_a:
  for j in LCM_b:
    if  i == j:
      LCM.append(j)
print(min(LCM))
```

```
a = int(input())
b = int(input())
a1 = a
b1 = b
LCM_a = []
LCM_b = []
LCM = []
i = 1
while i<= 10:
  LCM_a.append(a1)
  a1+=a
  i+=1
i = 1
while i<= 10:
  LCM_b.append(b1)
  b1+=b
  i+=1
i = 0
while i < len(LCM_a):
  j = 0
  while j < len(LCM_b):
    if LCM_a[i] == LCM_b[j]:
      LCM.append(LCM_b[j])
    j+=1
  i+=1
print(min(LCM))
```

```
24
60
120
```

## 7:Write a C program to calculate the sum of the digits of an input integer.

```
a = list(input())
i = 0
total = 0
while i< len(a):
    total = total + int(a[i])
    i+=1
print(total)
```

```
65237
23
```

```
a=['1']
print(a[0])
```

```
1
```

## 8:Write a C program to find out the reverse of an input integer.

```
a = list(input())
a.reverse()
i = 0
total = ''
while i< len(a):
  total = total + a[i]
  i+=1
print(int(total))
```

```
65237
73256
```

## 9:Write a C program to find out if an input integer is palindrome or not.

```
a = input()
c = int(a)
b = list(a)
b.reverse()
i = 0
total1 = '0'
while i < len(b):
  total1 = total1 + b[i]
  i+=1
if int(total1) == c:
    print('palindrome')
else:
    print('Not palindrome')
```

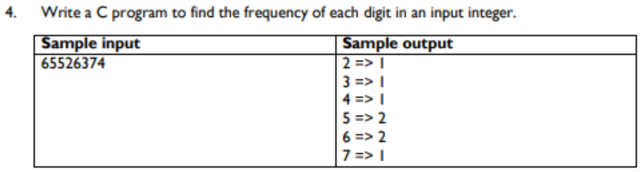
```
345
Not palindrome
```

```
a = list(input())

print(a)
print(b)
```

```
123
['1', '2', '3']
<list_reverseiterator object at 0x7be72ec8cfd0>
```

## 10:



4. Write a C program to find the frequency of each digit in an input integer.

| Sample input | Sample output |
|---|---|
| 65526374 | 2 => 1<br>3 => 1<br>4 => 1<br>5 => 2<br>6 => 2<br>7 => 1 |

```
a = input()
c = int(a)
b = list(a)
i = 0
while i< len(b):
  print(f'{b[i]} => {b.count(b[i])}')
  i+=1
```

```
65526374
6 => 2
5 => 2
5 => 2
2 => 1
6 => 2
3 => 1
7 => 1
```

```
4 => 1
```

```
a = input()
b = list(a)

for digit in b:
    print(f'{digit} => {b.count(digit)}')
```

```
54556
5 => 3
4 => 1
5 => 3
5 => 3
6 => 1
```

This way

```
num = int(input("Enter an integer: "))

# Initialize variables to store frequency of each digit
freq0, freq1, freq2, freq3, freq4, freq5, freq6, freq7, freq8, freq9 = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

# Count the frequency of each digit
while num != 0:
    digit = num % 10  # Extract the last digit
    if digit == 0:
        freq0 += 1
    elif digit == 1:
        freq1 += 1
    elif digit == 2:
        freq2 += 1
    elif digit == 3:
        freq3 += 1
    elif digit == 4:
        freq4 += 1
    elif digit == 5:
        freq5 += 1
    elif digit == 6:
        freq6 += 1
    elif digit == 7:
        freq7 += 1
    elif digit == 8:
        freq8 += 1
    elif digit == 9:
        freq9 += 1
    num //= 10       # Remove the last digit

# Print the frequency of each digit
print("Frequency of each digit:")
if freq0 != 0:
    print(f"0 => {freq0}")
if freq1 != 0:
    print(f"1 => {freq1}")
if freq2 != 0:
    print(f"2 => {freq2}")
if freq3 != 0:
    print(f"3 => {freq3}")
if freq4 != 0:
    print(f"4 => {freq4}")
if freq5 != 0:
    print(f"5 => {freq5}")
if freq6 != 0:
    print(f"6 => {freq6}")
if freq7 != 0:
    print(f"7 => {freq7}")
if freq8 != 0:
    print(f"8 => {freq8}")
if freq9 != 0:
    print(f"9 => {freq9}")
```

```
Enter an integer: 66579775
Frequency of each digit:
5 => 2
6 => 2
```

```
7 => 3
9 => 1
```

Wrong

```
# num = int(input("Enter an integer: "))
# freq = [0] * 10

# # Count the frequency of each digit
# while num != 0:
#     digit = num % 10  # Extract the last digit
#     freq[digit] += 1   # Increment the frequency of the digit
#     num //= 10        # Remove the last digit

# # Print the frequency of each digit
# print("Frequency of each digit:")
# for i in range(10):
#     if freq[i] != 0:
#         print(f"{i} => {freq[i]}")
```

```
Enter an integer: 5567456
Frequency of each digit:
4 => 1
5 => 3
6 => 2
7 => 1
```

## 11:Write a C program to swap the first and the last digits of an input integer

```
a = input()

first = a[0]
last = a[-1]

switched = last + a[1:-1] + first
print(switched)
```

```
65237
75236
```

## 12:Write a C program to find out if an input integer is an Armstrong number or not. An Armstrong number is one for which the sum of each digit to the power of the number of digits is the same as the number itself.

```
a = input()
b = list(a)
total1 = 0
total2 = '0'
for i in b:
  total2 = total2+i
for i in b:
  total1= total1 + int(i)**len(b)
if total1 == int(total2):
  print('Armstrong')
else:
  print('Not Armstrong')
```

```
1634
Armstrong
```

```
a = input()
b = list(a)
total1 = 0
total2 = '0'
i = 0
while i < len(b):
  total1 = total1 + int(b[i])**len(b)
  i+=1
i = 0
while i < len(b):
  total2 = total2 + b[i]
  i+=1
if total1 == int(total2):
  print('Armstrong')
else:
  print('Not Armstrong')
```

```
503
Not Armstrong
```

## 13:Write a C program to convert an input integer to binary

floor point number...

```
number = int(input())
binary = ''
while 0<number:
  binary = str(number % 2) + binary
  number = number // 2
print(binary)
```

```
123
1111011
```

Problem

floating point number...

```
number = float(input())
binary = ''
i = 0
j = 5
number1 = number
while i<=j:
  binary = binary + str(int(number1 * 2))
  number1 = number1*2
  i+=1
print(binary)
```

```
.123
000137
```

Decimal to binary

```
number = float(input())
# number_floor = int(number)
# binary1 = ''
# while 0< number_floor:
#   binary1 = str(int(number_floor % 2)) + binary1
#   number_floor = number_floor // 2
# print(binary1)
```

## 14:Write a C program to convert an input integer to octal.

```
number = int(input())
octal = ''
while 0<number:
  octal = str(number % 8) + octal
  number = number // 8
print(octal)
```

```
123
173
000137153162125251
```

## 15:Write a C program to convert a binary integer to decimal.

```
number = int(input())
number1 = list(str(number))
number1.reverse()
print(number1)
decimal = 0
i=0
while i<len(number1):
    decimal = decimal + int(number1[i])*2**i
    i+=1
print(decimal)
```

```
1111011
['1', '1', '0', '1', '1', '1', '1']
123
```

## 16:Write a C program to convert an octal integer to decimal.