CONDITIONAL AND LOOPING CONSTRUCTS

PROGRAMS

```
# check if the number is positive or negative or zero using if elif
num = float(input(“enter a number”))
if num > 0:
   print("Positive number")
elif num == 0:
   print("Zero")
else:
   print("Negative number")
```

```
# check if the number is positive or negative or zero using if else /nested if
num = float(input("Enter a number: "))
if num >= 0:
   if num == 0:
      print("Zero")
   else:
      print("Positive number")
else:
   print("Negative number")
```

```
#to check if the input number is odd or even
num = int(input("Enter a number: "))
if (num % 2) == 0:
   print("{0} is Even".format(num))
else:
   print("{0} is Odd".format(num))

#to check if the input year is a leap year or not
year = int(input("Enter a year: "))

if (year % 4) == 0:
   if (year % 100) == 0:
      if (year % 400) == 0:
         print("{0} is a leap year".format(year))
      else:
         print("{0} is not a leap year".format(year))
   else:
      print("{0} is a leap year".format(year))
else:
   print("{0} is not a leap year".format(year))
```

#to convert a numerical grade to a letter grade, ‘A’, ‘B’, ‘C’, #‘D’ or ‘F’, where the cutoffs for ‘A’, ‘B’, ‘C’, and ‘D’ are 90, #80, 70, and 60 respectively using if else ladder

```
score=int(input(“enter score”))
if score >= 90:
    letter = 'A'
  else:   # grade must be B, C, D or F
    if score >= 80:
      letter = 'B'
    else:  # grade must be C, D or F
      if score >= 70:
        letter = 'C'
      else:    # grade must D or F
        if score >= 60:
          letter = 'D'
        else:
          letter = 'F'
print(“grade is”,letter)
```

#to convert a numerical grade to a letter grade, ‘A’, ‘B’, ‘C’, #‘D’ or ‘F’, where the cutoffs for ‘A’, ‘B’, ‘C’, and ‘D’ are 90, #80, 70, and 60 respectively using if elif

```
score=int(input(“enter score”))
if score >= 90:
        letter = 'A'
    elif score >= 80:
        letter = 'B'
    elif score >= 70:
        letter = 'C'
    elif score >= 60:
        letter = 'D'
    else:
        letter = 'F'
print(“grade is”,letter)
```

#find the largest number among the three input numbers using logical operator

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))
if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3
print("The largest number is",largest)
```

#find the largest number among the three input numbers withou logical operator/nested if

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))
if(num1>num2):
        if(num1>num3):
                largest=num1
        else:
                largest=num3
else:
        if(num2>num3):
                largest=num2
        else:
                largest=num3
print("The largest number is",largest)
```

#Take in the Marks of 5 Subjects and Display the division

```
sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
sub4=int(input("Enter marks of the fourth subject: "))
sub5=int(input("Enter marks of the fifth subject: "))
avg=(sub1+sub2+sub3+sub4+sub4)/5
if(avg>=60):
    print(“first division”)
else:
    if(avg>=45):
        print(“second division”)
    else:
        if(avg>=33):
            print(“third division”)
        else:
            print(“fail”)
```

```
#to find the factorial of a number
n=int(input("Enter number:"))
fact=1
while(n>0):
    fact=fact*n
    n=n-1
print("Factorial of the number is: ")
print(fact)
```

```
#to find the sum of digits in a number
n=int(input("Enter a number:"))
tot=0
while(n>0):
    dig=n%10
    tot=tot+dig
    n=n//10
print("The total sum of digits is:",tot)
```

```
#to count the number of digits in a number
n=int(input("Enter number:"))
count=0
while(n>0):
    count=count+1
    n=n//10
print("The number of digits in the number are:",count)

#to check whether a given number is a palindrome
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if(temp==rev):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

```
#to compute prime factors of an integer
n=int(input("Enter an integer:"))
print("Factors are:")
i=1
while(i<=n):
   k=0
   if(n%i==0):
      j=1
      while(j<=i):
         if(i%j==0):
            k=k+1
         j=j+1
      if(k==2):
         print(i)
   i=i+1
```

```
#to check if a number is a strong number

sum1=0
num=int(input("Enter a number:"))
temp=num
while(num):
   i=1
   f=1
   r=num%10
   while(i<=r):
      f=f*i
      i=i+1
   sum1=sum1+f
   num=num//10
if(sum1==temp):
   print("The number is a strong number")
else:
   print("The number is not a strong number")
```

```
#to find the LCM of two numbers
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
if(a>b):
   min1=a
else:
   min1=b
while(1):
   if(min1%a==0 and min1%b==0):
      print("LCM is:",min1)
      break
   min1=min1+1
```

```
#Check Armstrong number (for 3 digits)
sum = 0
temp = num
while temp > 0:
  digit = temp % 10
  sum += digit ** 3
  temp //= 10
# display the result
if num == sum:
  print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

```
#checking whether a number is prime or not
import math
print ("Enter the a number")
number = int(input())
i = 2
prime = True
#if the number is not divisible by any number less than the #square root of the number
#then it is prime
while i <= int(math.sqrt(number)):
    if number%i == 0:
        prime = False
        break
    i = i+1

if number < 2:
    prime = False
if prime:
    print (number,"is a prime number")
else:
    print (number,"is not a prime number")
```

```
#Checking Armstrong Numbers
import math

print ("Enter the a number")
number = int(input())

#to calculate the number of digits in a number
number_of_digits = int(math.log10(number))+1

sum_arm = 0
temp = number
while temp != 0:
   sum_arm = sum_arm + int(math.pow(temp%10,number_of_digits))
   temp = temp//10
if sum_arm == number:
   print ("Yes an Armstrong number")
else:
   print ("No")
```

```
#to print the table of a given number
n=int(input("Enter the number to print the tables for:"))
for i in range(1,11):
    print(n,"x",i,"=",n*i)

#iterate over string
for character in 'hello':
    print(character)

#print no from 10 to 1
for i in range(10,0,-1):
   print(i)
#iterate through the list:
fruits = ['apple', 'banana', 'mango']
for fruit in sharks:
   print(fruit)
#iterating through a dictionary
shark = {'name': 'Sammy', 'animal': 'shark', 'color': 'blue'}
for key in shark:
   print(key + ': ' + sammy_shark[key])
```

```
#factorial of a given no
n = int(input('Enter an integer >= 0: '))
fact = 1
for i in range(2, n + 1):
    fact = fact * i
print(str(n) + ' factorial is ' + str(fact))

#check prine no
number = int(input("Enter any number: "))
# prime number is always greater than 1
if number > 1:
    for i in range(2, number):
        if (number % i) == 0:
            print(number, "is not a prime number")
            break
    else:
        print(number, "is a prime number")
# if the entered number is less than or equal to 1
# then it is not prime number
else:
    print(number, "is not a prime number")
```

```
#Fibonacci Series = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 ...

Number = int(input("\nPlease Enter the Range Number: "))

# Initializing First and Second Values of a Series
First_Value = 0
Second_Value = 1

# Find & Displaying Fibonacci series


for Num in range(0, Number):
        if(Num <= 1):
                Next = Num
        else:
                Next = First_Value + Second_Value
                First_Value = Second_Value
                Second_Value = Next
        print(Next)
```

#printing pyramid patterns

```
*
* *
* * *
* * * *
* * * * *
```

```
n=int(input("Enter a no"))
for i in range(0, n):

    # inner loop to handle number of columns
    # values changing acc. to outer loop
    for j in range(0, i+1):

        # printing stars
        print("* ",end="")
         # ending line after each row
    print("\r")
```

#printing pyramid patterns

```
        *
       * *
      * * *
     * * * *
    * * * * *
```

```
n=int(input("enter a no"))
k = 2*n - 2
    # outer loop to handle number of rows
   for i in range(0, n):
      # inner loop to handle number spaces
      # values changing acc. to requirement
      for j in range(0, k):
         print(end=" ")

      # decrementing k after each loop
      k = k - 2
      # inner loop to handle number of columns
      # values changing acc. to outer loop
      for j in range(0, i+1):

         # printing stars
         print("* ", end="")
          # ending line after each row
      print("\r")
```

#printing pyramid patterns

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
n=int(input("enter a no"))
k = 2*n - 2

   # outer loop to handle number of rows
   for i in range(0, n):

      # inner loop to handle number spaces
      # values changing acc. to requirement
      for j in range(0, k):
         print(end=" ")

      # decrementing k after each loop
      k = k - 1
      # inner loop to handle number of columns
      # values changing acc. to outer loop
      for j in range(0, i+1):
         # printing stars
         print("* ", end="")

      # ending line after each row
      print("\r")
```

#number pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

```
n=int(input("enter a no"))
num = 1

    # outer loop to handle number of rows
    for i in range(0, n):

        # re assigning num
        num = 1

        # inner loop to handle number of columns
            # values changing acc. to outer loop
        for j in range(0, i+1):

                # printing number
            print(num, end=" ")

            # incrementing number at each column
            num = num + 1
             # ending line after each row
        print("\r")
```

#Character Pattern

```
A
B B
C C C
D D D D
E E E E E
```

```
n=int(input("enter a no"))
num = 65

    # outer loop to handle number of rows
    # 5 in this case
    for i in range(0, n):

        # inner loop to handle number of columns
        # values changing acc. to outer loop
        for j in range(0, i+1):
            # explicitely converting to char
            ch = chr(num)
             # printing char value
            print(ch, end=" ")
             # incrementing number
        num = num + 1

        # ending line after each row
        print("\r")
```