CYCLE 2.1

1. Program to find the factorial of a number.

Algorithm:

① Enter a number (num)

② fact = 1

③ if num < 0

print negative number. no factorial

④ else num = 0

print fact = 1

⑤ else

for i in range (1, num+1)

fact = fact * i

⑥ print fact

⑦ stop

Program:

```
num = int (input ("Enter a number"))
fact = 1
if num < 0:
    print ("Negative number !! NO factorial..")
elif num == 0:
    print ("Factorial = 1")
else:
    for i in range (1, num+1):
        fact = fact * i
    print ("Factorial (", num, ") = ", fact)
```

Output

Enter a number 4

Factorial (4) = 24

2. Generate Fibonacci series of N terms.

Algorithm:

① Enter the number of elements (N)

② f1 = 1, f2 = 1, count = 0

③ if N <= 0

print enter a positive number

④ else if n1 == 1

print f1

⑤ else

while count < N

print(f1)

f3 = f1 + f2

f1 = f2 , f2 = f3 , count += 1

⑥ print

⑦ stop.

Program:

```
n = int(input("Enter number of terms : "))
f1 = 1
f2 = 1
count = 0
if(n<=0):
    print("please enter a positive number")
elif n==1:
    print("Fibonacci sequence upto",n,":")
    print(f1)
else:
    print("Fibonacci sequence:")
```

```
while  count < n :
        print (f1)
        f3 = f1 + f2
        f1 = f2
        f2 = f3
        count + = 1
```

Output

```
Number of terms : 5
Fibonacci sequence :
1
1
2
3
5
```

3. Find the sum of all items in a list.

Algorithm:

① Start
② Enter size of list
③ lp = [], sum = 0
④ Enter items
⑤ for i in range (0, n)
append item to lp
sum = sum + item.
⑥ print lp
⑦ stop.

Program:

```
lp = []
sum = 0
n = int (input (" Enter the size of list : "))
print (" Enter list items : \n")
for i in range (0, n):
    item = int (input())
    lp. append (item)
    sum = sum + item
print (" Given list = ", lp)
print (" sum of list items = ", sum)
```

Output

Enter size of list : 5

Enter list items

2

3

4

5

6

Given list = [2, 3, 4, 5, 6]

Sum of list items = 20

4. Generate a list of four digit numbers in a given range with all their digits even and number is a perfect square.

Algorithm:

① Start
② n1, n2
③ for i in range (n1, n2)
    for j in range (32, 100)
④ if i = j*j
    str(i)
⑤ if (string[0] %2 = 0 and string[1] %2 = 0 and string[2] %2 = 0 and string[3] %2 = 0)
⑥ print i
⑦ stop.

program:

```
n1 = int(input("Enter first number "))
n2 = int(input("Enter last number "))
print()
print("4 digit, perfect square, even digit numbers = ")
for i in range(n1, n2):
    for j in range(32,100):
        if i == j*j:
            string = str(i)
            if int(string[0])%2 == 0 and
            int(string[1])%2 == 0 and
```

```
        int(string[2]) %2 == 0      and
        int(string[3]) %2 == 0 :

        print(i)
```

Output

Enter first number 1000

Enter last number 9999

4 digit, perfect square, even digit numbers

4624

6084

6400

8464.

5. Display the given pyramid with step number accepted from user Eg: N = 4 1 2 4 3 6 9 4 8 12 16

Algorithm

① Start
② Enter choice (1-11)
③ print pattern
④ stop.

Program.

```
c = int(input("Enter your choice from 1-11 \n"))
if(c==1):
    print(4)
elif(c==2):
    print(4)
    print(4,1)
elif(c==3):
    print(4)
    print(4,1)
    print(4,1,2)
elif(c==4):
    print(4)
    print(4,1)
    print(4,1,2)
    print(4,1,2,4)
elif(c==5):
    print(4)
```

```
        print(4,1)
        print(4,1,2)
        print(4,1,2,4)
        print(4,1,2,4,3)

elif (c==6):
        print(4)
        print(4,1)
        print(4,1,2)
        print(4,1,2,4)
        print(4,1,2,4,3)
        print(4,1,2,4,3,6)

elif (c==7):
        print(4)
        print(4,1)
        print(4,1,2)
        print(4,1,2,4)
        print(4,1,2,4,3)
        print(4,1,2,4,3,6)
        print(4,1,2,4,3,6,9)

elif(c==8):
        print(4)
        print(4,1)
```

```
        Path(4,1,2)
        Path(4,1,2,4)
        Path(4,1,2,4,3)
        Path(4,1,2,4,3,6)
        Path(4,1,2,4,3,6,9,4)
elif(c==9):
        Path(4)
        Path(4,1)
        Path(4,1,2)
        Path(4,1,2,4)
        Path(4,1,2,4,3)
        Path(4,1,2,4,3,6)
        Path(4,1,2,4,3,6,9,4,8)
elif(c==10):
```

output

Enter your choice from 1-11

5

4

4 1

4 1 2

4 1 2 4

4 1 2 4 3

6. count the number of characters (character frequency) in a string.

Algorithm:

① start
② dict = { }
③ enter a string
④ for n in str
keys = dict.keys()
⑤ if n in keys
dict[n] = dict[n] + 1
else
dict[n] = 1
⑥ print dict
⑦ stop

Program:

```
dict = { }
str = input ("Enter a string")
for n in str:
    keys = dict.keys()
    if n in keys:
        dict[n] = dict[n] + 1
    else
        dict[n] = 1
print (dict)
```

Output

Enter a string jinxu shuji

{'j': 2, 'i': 2, 'n': 1, 'u': 1, ' ': 1, 's': 1, 'h': 1, 'a': 1}