



# **ACKNOWLEDGEMENT**

## PRACTICAL EXERCISE-1A

**Q: Write a program to find the factorial**

**PROGRAM: -**

```
school project > practicals > prog1.py > ...
1  number = int(input("Enter a number:"))
2  inp_num = number
3  factorial = 1
4  while number > 0:
5      factorial = factorial*number
6      number -= 1
7
8  print(f"The factorial of {inp_num} is {factorial}")
9
```

**OUTPUT: -**

```
PS F:\Harsh> python -u "f:\Harsh\my-programms\prog1A.py"
Enter a number:23
The factorial of 23 is 25852016738884976640000
```

## PRACTICAL EXERCISE-1B

**Q: Write a program to check if a number is prime or not prime**

**PROGRAM: -**

```
school project > practicals > prog2.py > ...
1  number = int(input("Enter a number: "))
2
3  if number < 2:
4      print(number, "is not a prime number")
5  else:
6      is_prime = True
7
8      for i in range(2, int(number**0.5) + 1):
9          if number % i == 0:
10             is_prime = False
11             break
12
13     if is_prime:
14         print(number, "is a prime number")
15     else:
16         print(number, "is not a prime number")
17
```

**OUTPUT: -**

```
● PS F:\Harsh> python -u "f:\Harsh\my-programms\prog1B.py"
Enter a number: 23
23 is a prime number
```

## PRACTICAL EXERCISE-2

**Q: Write a program to find the sum of a list recursively**

### PROGRAM: -

```
school project > practicals > prog3.py > summation
1  inpt_list = eval(input("Enter a list: "))
2
3  def summation(lst):
4      if len(lst) == 0:
5          return 0
6      else:
7          return lst[0] + summation(lst[1:])
8
9  result = summation(inpt_list)
10 print(f"The sum of the list: {inpt_list} is {result}")
11
```

### OUTPUT: -

```
PS F:\Harsh> python -u "f:\Harsh\my-programms\prog2.py"
Enter a list: [233,222,8678,44]
The sum of the list: [233, 222, 8678, 44] is 9177
```

### PRACTICAL EXERCISE-3

**Q: Write a program to calculate the  $n^{\text{th}}$  term of Fibonacci series.**

#### PROGRAM: -

```
school project > practicals > prog3.py > ...
1  def fib(n):
2      if n < 2:
3          return n
4      return fib(n-1) + fib(n-2)
5
6  def get_fibonacci_term(n):
7      print(f"Calculating the {n}th Fibonacci term...")
8      result = fib(n)
9      print(f"The {n}th Fibonacci term is {result}")
10
11  n = int(input("Enter the nth term to calculate: "))
12  get_fibonacci_term(n)
```

#### OUTPUT: -

```
● PS F:\Harsh> python -u "f:\Harsh\my-programms\prog3.py"
Enter the nth term to calculate: 34
Calculating the 34th Fibonacci term...
● The 34th Fibonacci term is 5702887
```

## PRACTICAL EXERCISE-4

**Q: Write a program to search any word in given string or sentence**

### PROGRAM: -

```
school project > practicals > prog4.py > ...
1  text = "This is a sample text. Here is another line."
2
3  query = input("Enter word to search: ")
4
5  lines = text.split('. ')
6  line_num = 1
7
8  for line in lines:
9      words = line.split()
10
11     word_num = 1
12
13     for word in words:
14         if word == query:
15             print(f"Found {query} at line {line_num}, word {word_num}")
16
17         word_num += 1
18
19     line_num += 1
```

### OUTPUT: -

```
PS F:\Harsh> python -u "f:\Harsh\my-programms\prog4.py"
Enter word to search: a
Found a at line 1, word 3
```

## PRACTICAL EXERCISE-5

**Q: Write a program to read and display file content line by line with each word separated by #**

### PROGRAM: -

```

school project > practicals > prog5.py > ...
1  filename = r'C:\Users\maste\OneDrive\Desktop\phyton programming\school
   project\practicals\data.txt'
2
3  with open(filename) as file:
4      for line in file:
5          words = line.strip().split()
6
7          print('#'.join(words))
8
9

```

### OUTPUT: -

```

PS F:\Harsh> python -u "f:\Harsh\my-programms\prog5.py"
Eu#do#ad#sunt#mollit#ex#ex#quis#et#ut#est#velit.#Fugiat#ad#eu#nulla#voluptate.#Lorem#exercitation#Lo
rem#voluptate#exercitation#proident#aliquip#nisi.

Nulla#laboris#ut#duis#eu#pariatur#tempor.#Sunt#velit#reprehenderit#aliqua#dolor#culpa#proident#adipi
sicing#labore#aliqua#non#enim#elit.#Id#consectetur#ex#non#esse.#Eiusmod#ea#irure#incidunt#eiusmod.
#Ex#qui#amet#dolore#Lorem#exercitation#consectetur#mollit#officia#velit#pariatur#sit#ullamco.

Sit#aliquip#consectetur#velit#aute.#Est#laboris#aliqua#ea#ut#sint.#Adipisicing#adipisicing#cupidatat
#ea#nulla#cillum#enim#ex#est#officia#reprehenderit.#Est#adipisicing#proident#cupidatat#dolor#duis#ad
ipisicing#proident.
PS F:\Harsh>

```



## PRACTICAL EXERCISE-6

**Q: Write a program to read the content of a file and display the total number of consonants, uppercase, vowels and lowercase characters.**

### PROGRAM: -

```
school project > practicals > prog6.py > ...
1
2 consonants_upper = 0
3 vowels = 0
4 lower_chars = 0
5
6 VOWELS = 'aeiou'
7
8 with open(r'C:\Users\maste\OneDrive\Desktop\phyton programming\schoo
  l project\practicals\data.txt','r') as f:
9
10     for line in f:
11         for char in line:
12             if char.isupper() and char not in VOWELS:
13                 consonants_upper += 1
14             elif char in VOWELS:
15                 vowels += 1
16             elif char.islower():
17                 lower_chars += 1
18
19 print("Consonant upper case letters:", consonants_upper)
20 print("Vowels:", vowels)
21 print("Lower case characters:", lower_chars)
```

### OUTPUT: -

```
PS F:\Harsh> python -u "f:\Harsh\my-programms\prog6.py"
O Consonant upper case letters: 14
Vowels: 248
Lower case characters: 292
```

