1.Two sum

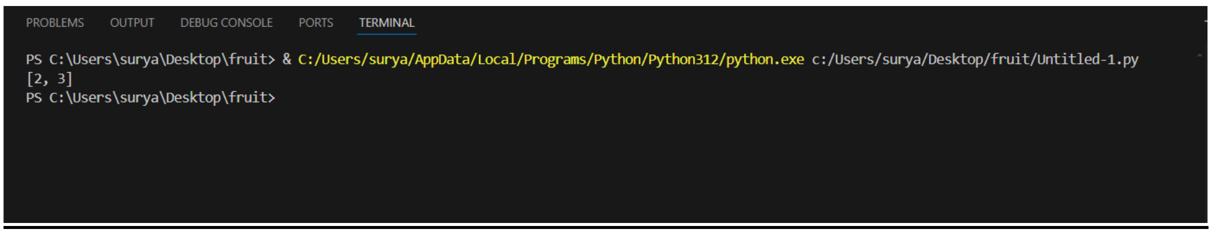
PROGRAM:

```
def two_sum(nums, target):
    temp= {}
    for i in range(len(nums)):
        complement = target - nums[i]
        if complement in temp:
            return [temp[complement], i]
        temp[nums[i]] = i
    return None
nums = [2, 7, 11, 15]
target = 26
result = two_sum(nums, target)
print(result)
```

TIME COMPLEXITY:O(n)

INPUT: 2,7,11,15

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
[2, 3]
PS C:\Users\surya\Desktop\fruit>
```

2.Add two numbers:

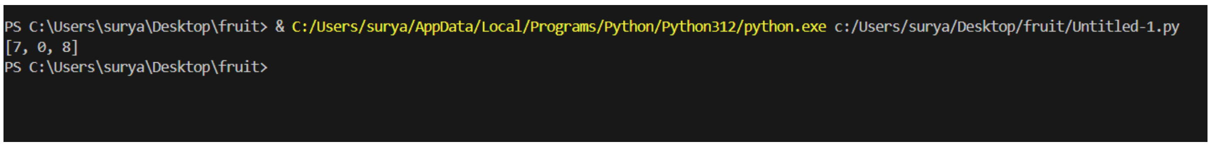
PROGRAM:

```
def add(a,b):
    a.reverse()
    b.reverse()
    anum=int(''.join(map(str,a)))
    bnum=int(''.join(map(str,b)))
    c=[]
    d=anum+bnum
    while d>0:
        r=d%10
        c.append(r)
        d=d//10
    return c
a=[2,4,3]
b=[5,6,4]
print(add(a,b))
```

TIME COMPLEXITY:O(N)

INPUT: 2,4,3,5,6,4

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
[7, 0, 8]
PS C:\Users\surya\Desktop\fruit>
```

3. Median of 2 sorted arrays:

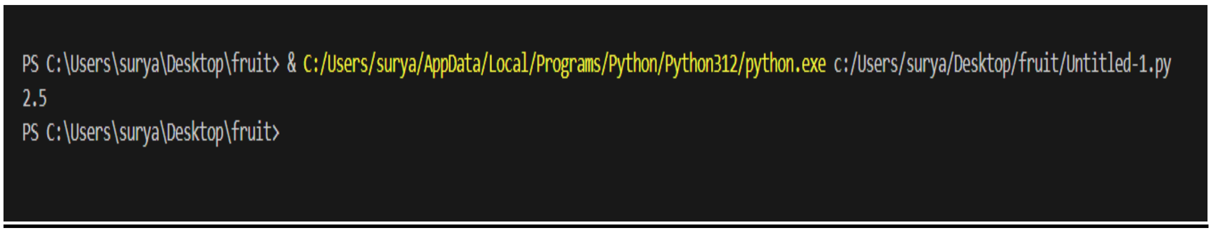
PROGRAM:

```
def median(nums1, nums2):
    merged = sorted(nums1 + nums2)
    n = len(merged)
    if n % 2 == 0:
        return (merged[n // 2 - 1] + merged[n // 2]) / 2
    else:
        return merged[n // 2]
nums1 = [1, 2]
nums2 = [3,4]
print(median(nums1, nums2))
```

TIME COMPLEXITY: O(n)

INPUT: 1,2 AND 3,4

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
2.5
PS C:\Users\surya\Desktop\fruit>
```

4.Longest substring palindrome:

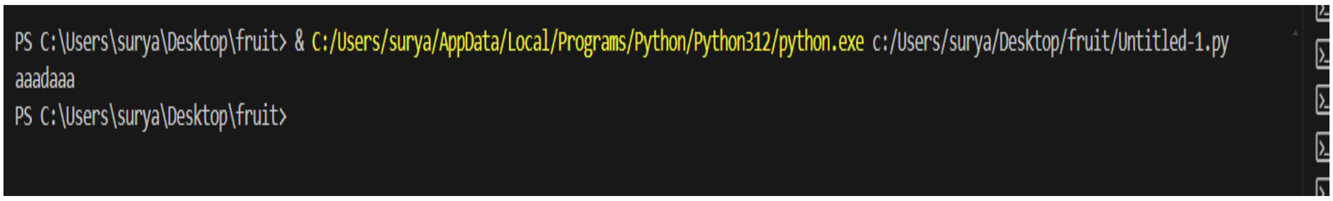
PROGRAM:

```
def palin(s):
    maxpalin=""
    for i in range(len(s)):
        for j in range(i,len(s)):
            substr=s[i:j+1]
            if substr==substr[::-1] and len(substr)>len(maxpalin):
                maxpalin=substr
    return maxpalin
string="babaaadaaa"
print(palin(string))
```

TIME COMPLEXITY: O(n^3)

INPUT:”babaadaaa”

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
aaadaaa
PS C:\Users\surya\Desktop\fruit>
```

5.Reverse a number:

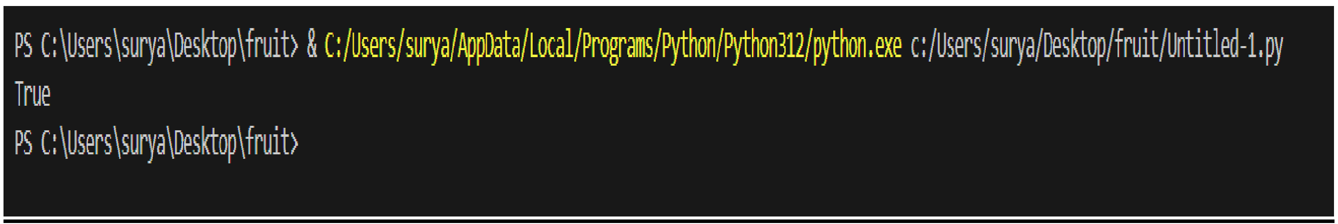
PROGRAM:

```
def rev(num):
    n=0
    while num>0:
        r=num%10
        n=(n*10)+r
        num=num//10
    return n
a=123
print(rev(a))
```

TIME COMPLEXITY:O(logn)

INPUT:123

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
True
PS C:\Users\surya\Desktop\fruit>
```

6.String to int:

PROGRAM:

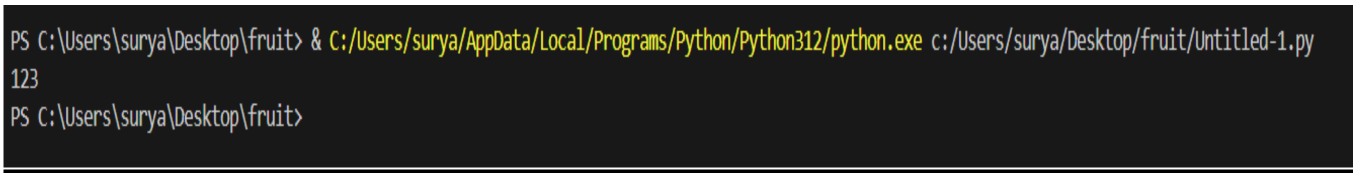
```
def string(str):
    return int(str)
```

```
a="123"
print(string(a))
```

TIME COMPLEXITY:O(n)

INPUT:123

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
123
PS C:\Users\surya\Desktop\fruit>
```

7.Palindrome or not:

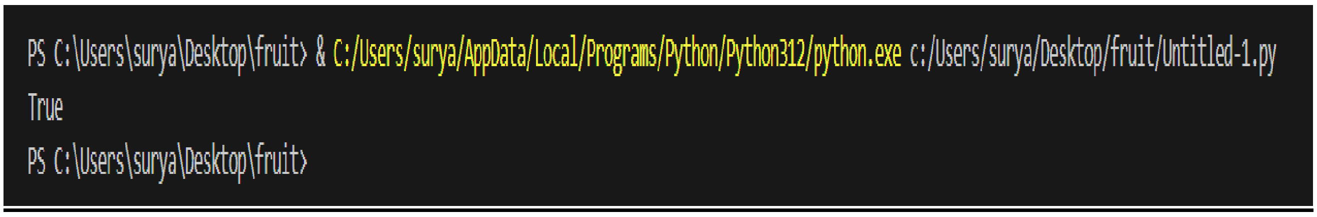
PROGRAM:

```
def rev(num):
    og=num
    n=0
    while num>0:
        r=num%10
        n=(n*10)+r
        num=num//10
    if n==og:
        return True
    else:
        return False
a=121
print(rev(a))
```

TIME COMPLEXITY:O(logn)

INPUT:121

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
True
PS C:\Users\surya\Desktop\fruit>
```

8.Longest substring without repeating chars:

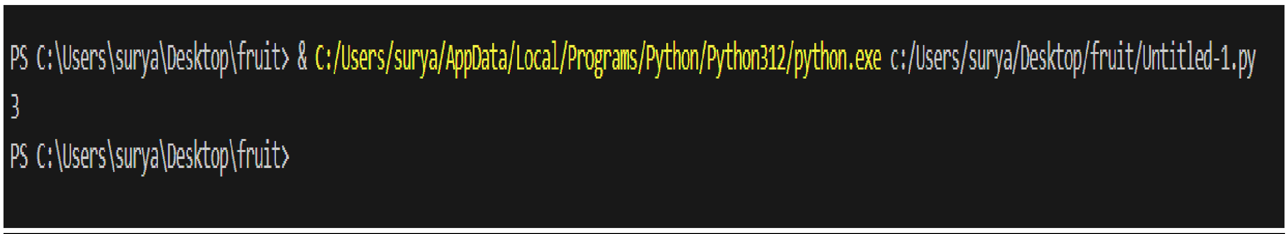
PROGRAM:

```
def length_of_longest_substring(s):
    char_index = {}
    start = 0
    max_length = 0
    for end in range(len(s)):
        if s[end] in char_index:
            start = max(start, char_index[s[end]] + 1)
        char_index[s[end]] = end
        max_length = max(max_length, end - start + 1)
    return max_length
s = "pwwkew"
print(length_of_longest_substring(s))
```

TIME COMPLEXITY:O(n)

INPUT:pwwkew

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
3
PS C:\Users\surya\Desktop\fruit>
```

9.Zigzag conversion:

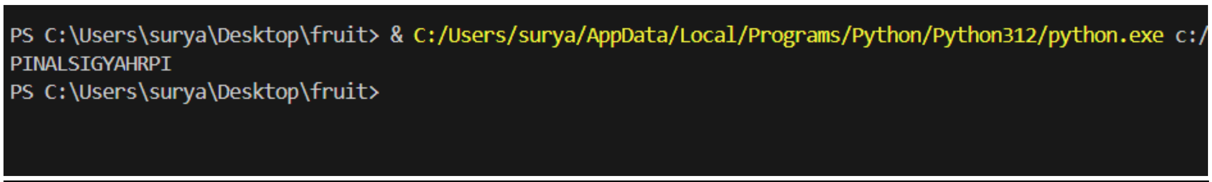
PROGRAM:

```
def convert(s, numRows):
    if numRows == 1 or numRows >= len(s):
        return s
    rows = [''] * numRows
    index, step = 0, 1
    for char in s:
        rows[index] += char
        if index == 0:
            step = 1
        elif index == numRows - 1:
            step = -1
        index += step
    return ''.join(rows)
a="PAYPALISHIRING"
b=4
print(convert(a,b))
```

TIME COMPLEXITY:O(n)

INPUT: PAYPALISHRING

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/
PINALSIGYAHRPI
PS C:\Users\surya\Desktop\fruit>
```

10.Regular Expression matching:

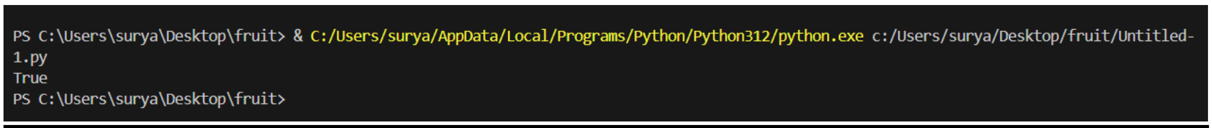
PROGRAM:

```
import re

def is_match(s, p):

    pattern = re.compile(p)

    return bool(pattern.fullmatch(s))

s = "ab"

p = ".*"

print(is_match(s, p))
```

TIME COMPLEXITY: O(n)

INPUT:ab

OUTPUT:



```
PS C:\Users\surya\Desktop\fruit> & C:/Users/surya/AppData/Local/Programs/Python/Python312/python.exe c:/Users/surya/Desktop/fruit/Untitled-1.py
True
PS C:\Users\surya\Desktop\fruit>
```