

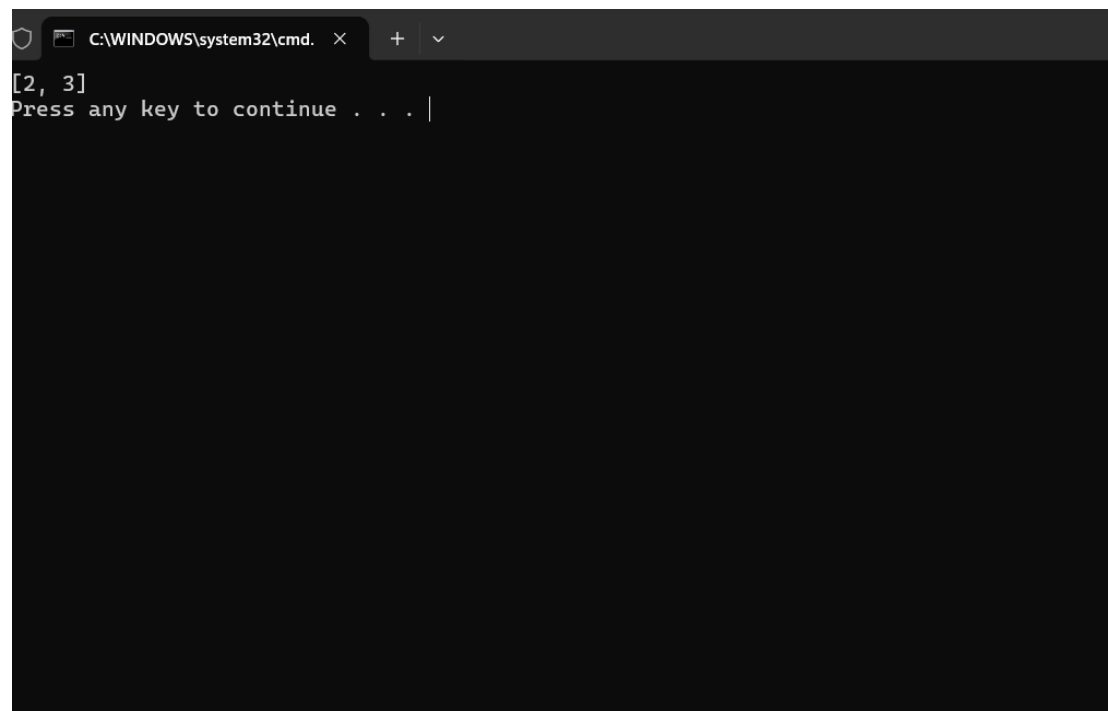
1) Two sum

CODE:

```
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)
```

OUTPUT:

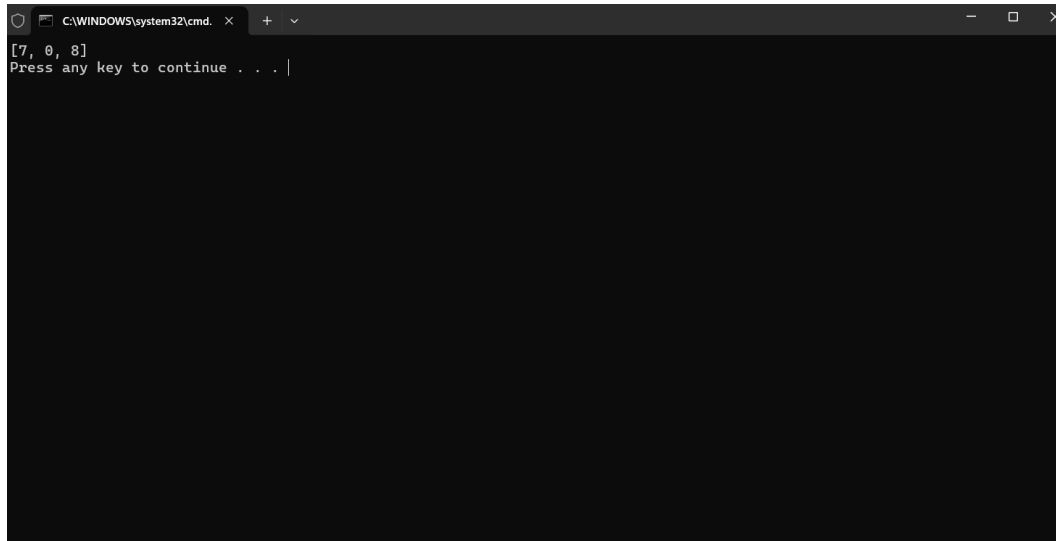
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the output '[2, 3]' on the first line and 'Press any key to continue . . . |' on the second line, indicating the program has finished execution and is waiting for a key press.

2)Add two numbers:

CODE:

```
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))
OUTPUT:
```

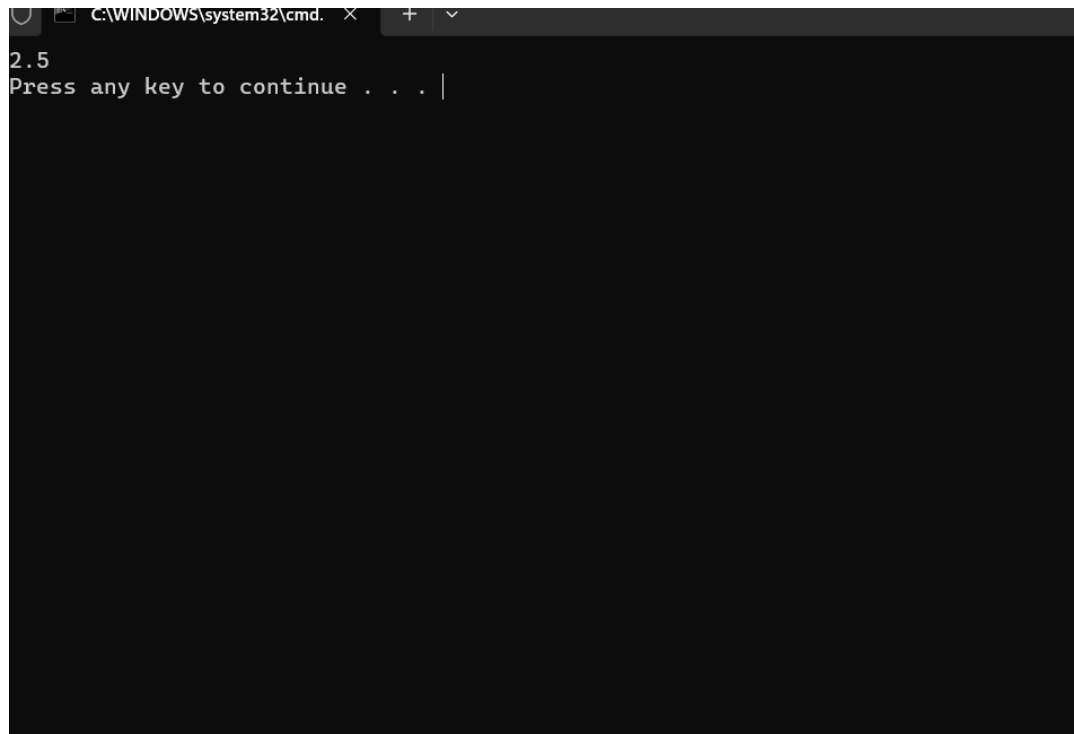
A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the output of a Python script: '[7, 0, 8]' followed by a prompt 'Press any key to continue . . . |'. The rest of the window is dark and empty.

3)Median of 2 sorted arrays:

CODE:

```
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))
```

OUTPUT:

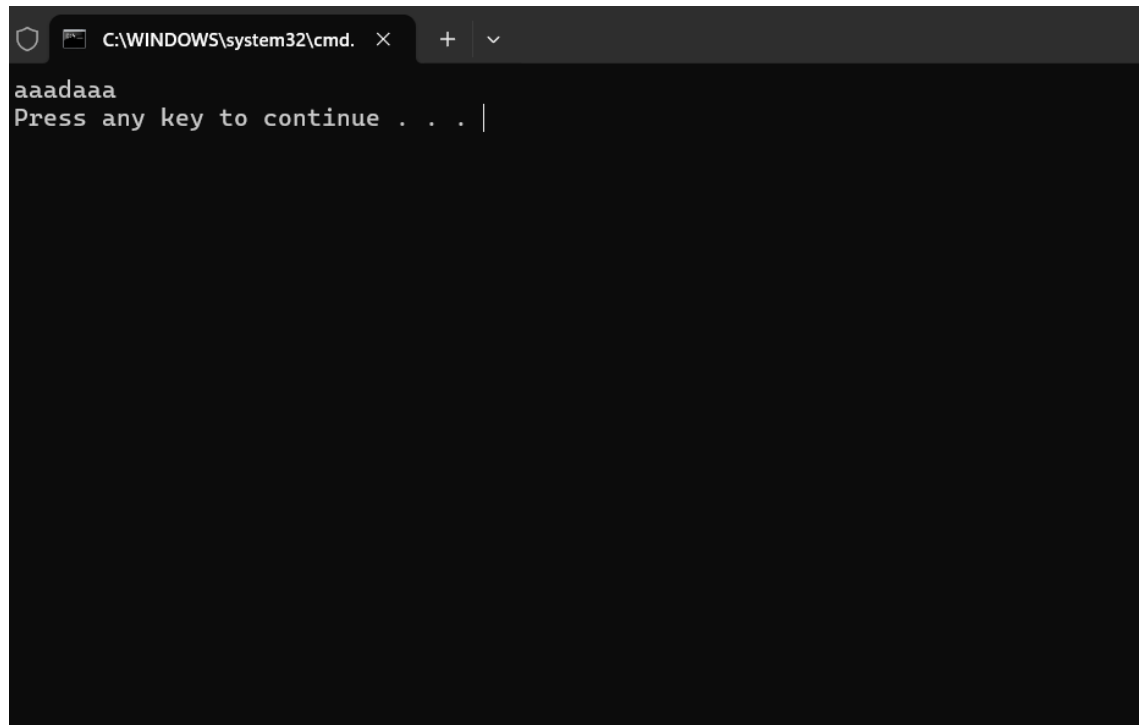


4) Longest substring palindrome:

CODE:

```
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="babaaadaaaa"
print(palin(string))
```

OUTPUT:

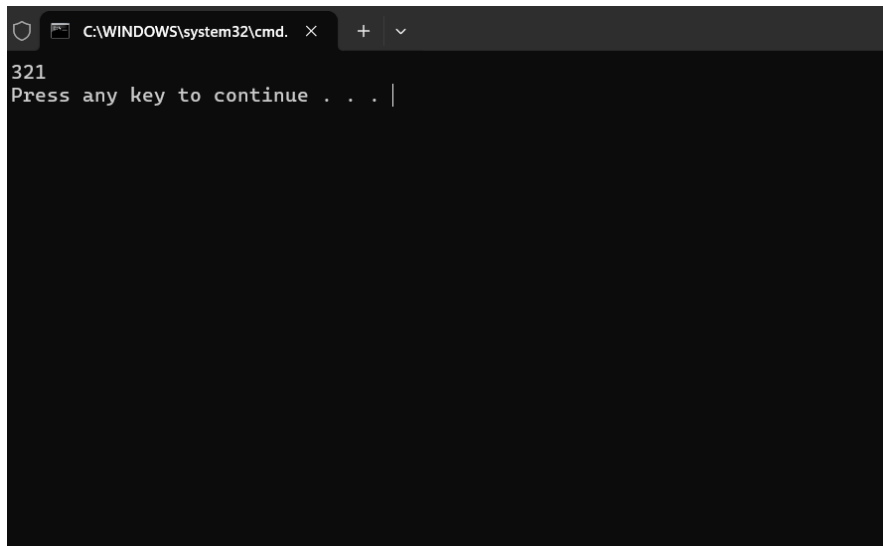
A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' with a close button, a plus sign, and a dropdown arrow. The command prompt displays the text 'aaadaaa' on the first line and 'Press any key to continue . . . |' on the second line, with a vertical cursor at the end of the second line.

5)Reverse a number:

CODE:

```
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))
```

OUTPUT:



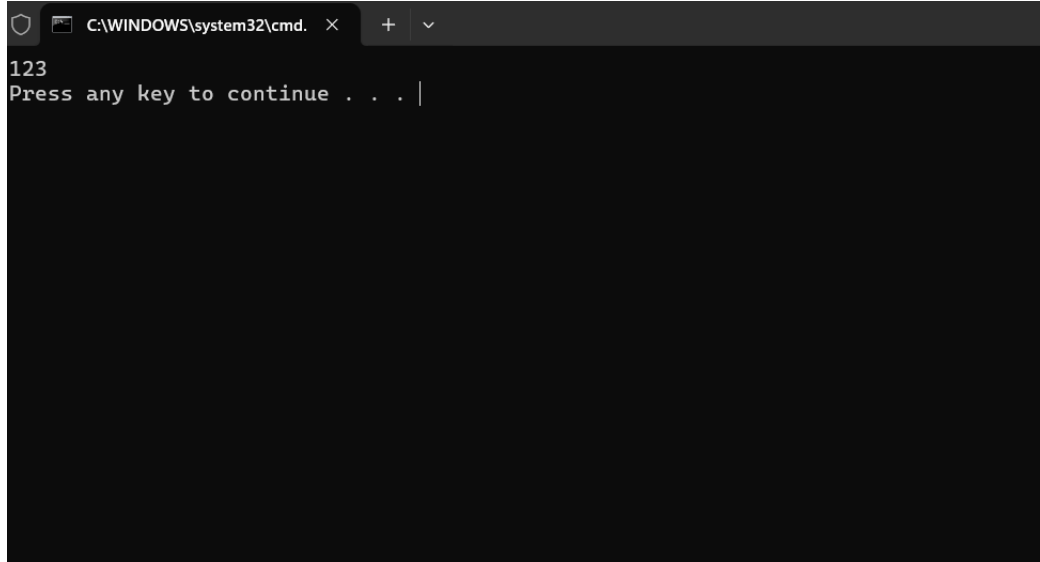
```
C:\WINDOWS\system32\cmd. 321
Press any key to continue . . . |
```

6) String to int:

CODE:

```
def string(str):
    return int(str)
a="123"
print(string(a))
```

OUTPUT:



```
C:\WINDOWS\system32\cmd. 123
Press any key to continue . . . |
```

7)Palindrome or not number:

CODE:

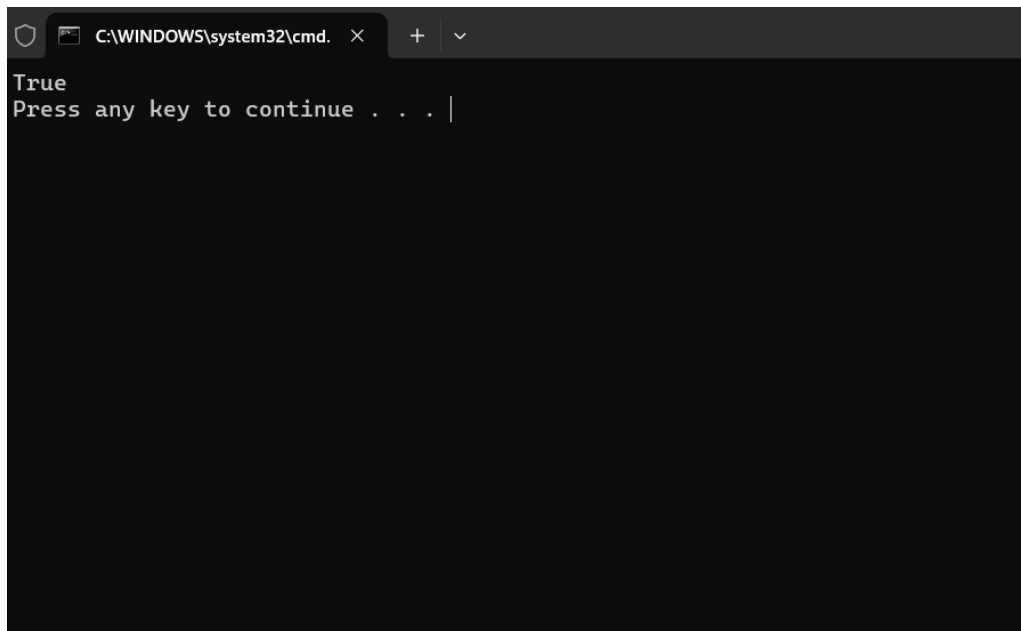
```
def rev(num):
    og=num
    n=0
    while num>0:
        r=num%10
```

```

        n=(n*10)+r
        num=num//10
    if n==og:
        return True
    else:
        return False
a=121
print(rev(a))

```

OUTPUT:



```

True
Press any key to continue . . . |

```

8) Longest substring without repeating chars:

CODE:

```

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))

```

OUTPUT:

```
C:\WINDOWS\system32\cmd. x + v
3
Press any key to continue . . . |
```

9)Zigzag coversion:

CODE:

```
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))
```

OUTPUT:

```
C:\WINDOWS\system32\cmd. x + v
PINALSIGYHRPI
Press any key to continue . . . |
```

10)Regular Expression matching:

CODE:

```
import re

def is_match(s, p):
    pattern = re.compile(p)
    return bool(pattern.fullmatch(s))

s = "ab"
p = ".*"
print(is_match(s, p))
OUTPUT:
```



