CAS0888 – PYTHON PROGRAMMING

DATE: 17-08-23

NAME:G. keerthireddy

REG.NO:

192224165

1.INPUT:

```python
MAX_CHARS = 256

def areIsomorphic(string1, string2):
        m = len(string1)
        n = len(string2)
        if m != n:
                return False
        marked = [False] * MAX_CHARS
        map = [-1] * MAX_CHARS
        for i in xrange(n):
                if map[ord(string1[i])] == -1:
                        if marked[ord(string2[i])] == True:
                                return False
                        marked[ord(string2[i])] = True
                        map[ord(string1[i])] = string2[i]
                elif map[ord(string1[i])] != string2[i]:
                        return False
        return True
print ("aab", "xxy")
```

2.INPUT:

```python
def sumsquare(l):
    odd_sum = 0
    even_sum = 0
    for num in l:
        if num % 2 == 0:
```

```python
            even_sum += num ** 2
        else:
            odd_sum += num ** 2
    return [odd_sum, even_sum]
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
result = sumsquare(numbers)
print(result)
```

3.INPUT:

```python
def is_happy(n):
    def get_next(num):
        next_num = 0
        while num > 0:
            digit = num % 10
            next_num += digit ** 2
            num //= 10
        return next_num
    slow = n
    fast = n
    while True:
        slow = get_next(slow)
        fast = get_next(get_next(fast))
        if slow == fast:
            break
    return slow == 1
num = 19
print(is_happy(num))
```

4.INPUT:

```python
def is_palindrome(x):
    if x < 0:
```

```python
        return False
    original = x
    reverse = 0
    while x > 0:
        digit = x % 10
        reverse = reverse * 10 + digit
        x //= 10
    return original == reverse
num1 = 121
num2 = 123
print(is_palindrome(num1))
print(is_palindrome(num2))
```

5.INPUT:

```python
def calculate_price(num_fresh, num_day_old):
    regular_price = (num_fresh + num_day_old) * 185
    discount = num_day_old * 185 * 0.60
    total_price = regular_price - discount
    return regular_price, discount, total_price


def main():
    try:
        num_fresh = int(input("Enter the number of fresh loaves purchased: "))
        num_day_old = int(input("Enter the number of day-old loaves purchased: "))
        if num_fresh < 0 or num_day_old < 0:
            print("Please enter non-negative numbers of loaves.")
        else:
            regular_price, discount, total_price = calculate_price(num_fresh, num_day_old)
            print("\nRegular Price: {:.2f} rupees".format(regular_price))
            print("Discount: {:.2f} rupees".format(discount))
```

```python
        print("Total Price: {:.2f} rupees".format(total_price))
    except ValueError:
        print("Invalid input. Please enter valid integers.")
if __name__ == "__main__":
    main()
```

6.INPUT:

```python
def max_area(height):
    left = 0
    right = len(height) - 1
    max_area = 0
    while left < right:
        h = min(height[left], height[right])
        w = right - left
        area = h * w
        max_area = max(max_area, area)
        if height[left] < height[right]:
            left += 1
        else:
            right -= 1
    return max_area
heights = [1,8,6,2,5,4,8,3,7]
print(max_area(heights))
```

7.INPUT:

```python
def count_sorted_vowel_strings(n):
    vowels = 'aeiou'
    def generate_strings(curr_str, index):
        if len(curr_str) == n:
            return 1
        count = 0
```

```python
        for i in range(index, len(vowels)):
            count += generate_strings(curr_str + vowels[i], i)
        return count
    return generate_strings('', 0)
n = 2
print(count_sorted_vowel_strings(n))
```

8.INPUT:

```python
import re
def isNumber(s):
    s = s.strip()
    integer_pattern = r'^[+-]?\d+$'
    decimal_pattern = r'^[+-]?(\d+\.\d*|\.\d+)(e[+-]?\d+)?$'
    exponential_pattern = r'^[+-]?\d+e[+-]?\d+$'
    if re.match(integer_pattern, s):
        return True
    if re.match(decimal_pattern, s):
        return True
    if re.match(exponential_pattern, s):
        return True
    return False
valid_numbers = ["2", "0089", "-0.1", "+3.14", "4.", "-.9", "2e10", "-90E3", "3e+7", "+6e-1",
"53.5e93", "-123.456e789"]
invalid_numbers = ["abc", "1a", "1e", "e3", "99e2.5", "--6", "-+3", "95a54e53"]
for num in valid_numbers:
    print(f"{num}: {isNumber(num)}")
for num in invalid_numbers:
    print(f"{num}: {isNumber(num)}")
```

9.INPUT:

```python
def max_guests_within_time_limit(T, E, L):
    max_guests = 0
```

```python
        current_guests = 0
    for i in range(T):
        current_guests += E[i] - L[i]

        max_guests = max(max_guests, current_guests)

    return max_guests

T = 5

E = [7, 0, 5, 1, 3]

L = [1, 2, 1, 3, 4]

result = max_guests_within_time_limit(T, E, L)

print(result)
```

10.INPUT:

```python
def modify_string(s):

    def circular_distance(ch):

        return (ord(ch) - ord('a') + 1) % 26

    char_frequency = {}

    modified_string = ""

    for ch in s:

        if ch in char_frequency:

            char_frequency[ch] += 1

        else:

            char_frequency[ch] = 1

    for ch in s:

        freq = char_frequency[ch]

        distance = circular_distance(ch)

        new_char = chr((ord('a') + distance - 1) % 26 + ord('a'))

        modified_string += new_char

    return modified_string

input_string = "abbccc"

result = modify_string(input_string)
```

```
print(result)
```