E: > PY_file > C2 > 2_2.py > MoneyStr

```python
MoneyStr = input("输入需要换算的带单位的金额：")

while MoneyStr[-1] not in ['&','*']:
    if MoneyStr[-1] in ['$']:
        RMB = (eval(MoneyStr[0:-1])*6)
        print("对应的人民币金额为：{:.2f}¥".format(RMB))
    if MoneyStr[-1] in ['¥20']:
        DOLLAR = (eval(MoneyStr[0:-1])/6)
        print("对应的美元金额为：{:.2f}$".format(DOLLAR))
    MoneyStr = input("输入需要换算的带单位的金额：")
```

E: > PY_file > C2 > 2_3.py > ...

```python
import turtle

import random

turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.seth(-40)
for i in range(4):
#    turtle.pencolor((int(random.random()*255),int(random.random()*255),int(random.random()*255)))
    turtle.pencolor((random.random(), random.random(), random.random()))
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)

turtle.done()
```

E: > PY_file > C2 > 2_7.py > ...

```python
import turtle

pen = turtle.Turtle()

pen.speed(10)
pen.color("pink")

def draw_tri():
    for i in range(3):
        pen.forward(100)
        pen.left(120)
    pen.forward(100)
    pen.right(60)

while True:
    pen.right(30)
    for _ in range(6):
        draw_tri()
    turtle.done()
```

E:\PY_file\C2\2_8.py

```python
import turtle

pen = turtle.Turtle()

pen.speed(100)
pen.color("pink")

def draw_line(a):
    pen.forward(a)
    pen.right(90)

while True:
    for i in range(200):
        draw_line(2*i)
    turtle.done()
```

```python
current_weight_earth = float(input("请输入你当前的体重（kg）："))

moon_weight_ratio = 0.165

annual_growth = 0.5

print("\n未来10年体重状况：")
print("{:<10} {:<15} {:<15}".format("年份", "地球体重(kg)", "月球体重(kg)"))
for year in range(1, 11):
    future_weight_earth = current_weight_earth + year * annual_growth
    future_weight_moon = future_weight_earth * moon_weight_ratio
    print("{:<10} {:<15.2f} {:<15.2f}".format(year, future_weight_earth, future_weight_moon))
```

E: > PY_file > C3 > 3_4.py > is_palindrome

```python
def is_palindrome(n):
    str_n = str(n)
    return str_n == str_n[::-1]

num = int(input("请输入一个5位数字： "))

if is_palindrome(num):
    print(f"{num} 是一个回文数")
else:
    print(f"{num} 不是一个回文数")
```

E: > PY_ E:\PY_file\C3\3_5.py > print_tic_tac_toe

```python
def print_tic_tac_toe():
    for i in range(11):
        if i % 5 == 0:
            print("+ - - - - + - - - - +")
        else:
            print("|         |         |")

print_tic_tac_toe()
```

E: > PY_file > C3 > 3_6.py > ...

```python
import time

def print_progress_bar(duration=5):
    print("Starting", end="")
    for i in range(10):
        print(".", end="")
        time.sleep(duration / 10)
    print("Done!")

# 调用函数，打印进度条
print_progress_bar()
```

E: > PY_file > C4 > 4_1.py > ...

```python
import random

def guess_number():
    target = random.randint(0, 9)
    attempts = 0

    while True:
        try:
            guess = int(input("请输入你猜的数字（0-9）："))
            if guess < 0 or guess > 9:
                print("请输入一个0到9之间的整数。")
                continue
            attempts += 1
            if guess == target:
                print(f"预测{attempts}次，你猜中了！")
                break
            elif guess > target:
                print("遗憾，太大了")
            else:
                print("遗憾，太小了")
        except ValueError:
            print("请输入一个有效的整数。")


if __name__ == "__main__":
    guess_number()
```

```python
def count_characters(input_string):
    count_alpha = 0
    count_digit = 0
    count_space = 0
    count_other = 0

    for char in input_string:
        if char.isalpha():
            count_alpha += 1
        elif char.isdigit():
            count_digit += 1
        elif char.isspace():
            count_space += 1
        else:
            count_other += 1

    return count_alpha, count_digit, count_space, count_other

if __name__ == "__main__":
    input_string = input("请输入一行字符: ")
    count_alpha, count_digit, count_space, count_other = count_characters(input_string)
    print("英文字符个数:", count_alpha)
    print("数字个数:", count_digit)
    print("空格个数:", count_space)
    print("其他字符个数:", count_other)
```

4-2

```python
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

def lcm(a, b):
    return a * b // gcd(a, b)

if __name__ == "__main__":
    num1 = int(input("请输入第一个整数: "))
    num2 = int(input("请输入第二个整数: "))
    print("最大公约数是:", gcd(num1, num2))
    print("最小公倍数是:", lcm(num1, num2))
```

4-3

```python
import random #4-6

def simulate_monty_hall(trials):
    switch_wins = 0
    stay_wins = 0

    for _ in range(trials):
        # 随机选择一扇门放汽车，其余放山羊
        doors = [0, 1, 2]
        car_door = random.choice(doors)
        player_choice = random.choice(doors)

        # 主持人打开一扇山羊门
        host_opens = [door for door in doors if door != car_door and door != player_choice][0]
```

```
        # 参赛者更换选择
        remaining_doors = [door for door in doors if door != player_choice and door != host_opens]
        player_switches = remaining_doors[0]

        # 判断是否获胜
        if player_switches == car_door:
            switch_wins += 1
        else:
            stay_wins += 1

    return switch_wins / trials, stay_wins / trials

trials = 100000
switch_probability, stay_probability = simulate_monty_hall(trials)
print("更换选择后获胜的概率：", switch_probability)
print("坚持选择后获胜的概率：", stay_probability)
```

E: > PY_file > C5 > 2.py > ...

```
def isOdd(num):
    if num % 2 != 0:
        return True
    else:
        return False

number = int(input("请输入一个整数："))
if isOdd(number):
    print(f"{number} 是奇数")
else:
    print(f"{number} 不是奇数")
```

E: > PY_file > C5 > 4.py > ...

```
def multi(*args):
    product = 1
    for num in args:
        product *= num
    return product

input_numbers = input("请输入多个数字，用空格分隔：")
numbers = map(int, input_numbers.split())
result = multi(*numbers)
print("乘积是：", result)
```

E: > PY_file > C5 > 5.py > isPrime

```python
def isPrime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

try:
    number = int(input("请输入一个整数："))
    if isPrime(number):
        print(f"{number} 是质数")
    else:
        print(f"{number} 不是质数")
except ValueError:
    print("输入错误，请输入一个有效的整数。")
```

from datetime import datetime #5-6

def format_birthday(birthday_str):

    birthday = datetime.strptime(birthday_str, "%Y-%m-%d")

    date_formats = [
        "%Y-%m-%d",
        "%d/%m/%Y",
        "%m-%d-%Y",
        "%B %d, %Y",
        "%b %d, %Y",
        "%d %B %Y",
        "%d %b %Y",
        "%A, %B %d, %Y",
        "%a, %b %d, %Y",
        "%Y 年%m 月%d 日",
        "%Y.%m.%d",
        "%d.%m.%Y",
        "%d-%m-%Y",
        "%m/%d/%Y",
        "%Y/%m/%d",
        "%d %B, %Y",
        "%d %b, %Y",
        "%B %d, %Y",
        "%b %d, %Y",

```
        "%A, %d %B %Y",
        "%a, %d %b %Y",
        "%Y-%j",
        "%Y/%j",
    ]

    for fmt in date_formats:
        try:
            print(birthday.strftime(fmt))
        except ValueError as e:
            print(f"Error with format {fmt}: {e}")

birthday_input = input("请输入你的生日（格式为 YYYY-MM-DD）：")
format_birthday(birthday_input)
```

E: > PY_file > C6 > 1.py > ...

```python
import random
import string

def generate_password(length=8):
    characters = string.ascii_letters + string.digits
    password = ''.join(random.choice(characters) for _ in range(length))
    return password

def generate_multiple_passwords(count, length=8):
    return [generate_password(length) for _ in range(count)]

passwords = generate_multiple_passwords(10)
for i, pwd in enumerate(passwords, start=1):
    print(f"密码 {i}: {pwd}")
```

E: > PY_file > C6 > 2.py > has_duplicates

```python
def has_duplicates(lst):
    seen = set()
    for element in lst:
        if element in seen:
            return True
        seen.add(element)
    return False

test_list1 = [1, 2, 3, 4, 5]
test_list2 = [1, 2, 3, 4, 5, 3]
test_list3 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
test_list4 = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'a']

print("测试 1:", test_list1, "是否有重复?", has_duplicates(test_list1))
print("测试 2:", test_list2, "是否有重复?", has_duplicates(test_list2))
print("测试 3:", test_list3, "是否有重复?", has_duplicates(test_list3))
print("测试 4:", test_list4, "是否有重复?", has_duplicates(test_list4))
```

E: > PY_file > C6 > 5.py > ...

```python
import random

def simulate_birthdays(num_samples, num_people):
    match_count = 0

    for _ in range(num_samples):
        birthdays = [random.randint(1, 365) for _ in range(num_people)]
        if len(set(birthdays)) != len(birthdays):
            match_count += 1

    return match_count / num_samples

def main():
    num_people = 23
    num_samples_list = [1000, 10000, 100000]

    for num_samples in num_samples_list:
        probability = simulate_birthdays(num_samples, num_people)
        print(
            f"样本数量: {num_samples},至少两人同一天生日概率: {probability:.4f}")

if __name__ == "__main__":
    main()
```

from PIL import Image #7-2
import os

```
def compress_image(input_path, output_path, target_size_kb=10):
    target_size = target_size_kb * 1024

    img = Image.open(input_path)

    quality = 85
    img.save(output_path, format='JPEG', quality=quality)

    while os.path.getsize(output_path) > target_size:
        quality -= 5
        img.save(output_path, format='JPEG', quality=quality)

        if quality < 10:
            break

    if os.path.getsize(output_path) > target_size:
        width, height = img.size
        img = img.resize((width // 2, height // 2), Image.LANCZOS)

        img.save(output_path, format='JPEG', quality=quality)
```

```
        while os.path.getsize(output_path) > target_size:
            quality -= 5
            img.save(output_path, format='JPEG', quality=quality)

            if quality < 10:
                break

compress_image('111.jpg', 'output.jpg')



from PIL import Image ##7-3

CHARS = "    的一是了不在有我他她它们这那你们是个好一个人们和为上来去对说而也就要"

def image_to_ascii(input_image_path, output_text_path, width=100):

    img = Image.open(input_image_path)
    img = img.convert("L")

    # 计算图像的高度
    aspect_ratio = img.height / img.width
    height = int(aspect_ratio * width * 0.5)

    img = img.resize((width, height))

    pixels = img.getdata()

    ascii_str = ""
    num_chars = len(CHARS)

    for pixel in pixels:

        char_index = min(pixel * num_chars // 256, num_chars - 1)
        ascii_str += CHARS[char_index]

    ascii_str_len = len(ascii_str)
    ascii_image = "\n".join(ascii_str[i:i + width] for i in range(0, ascii_str_len, width))

    with open(output_text_path, "w", encoding="utf-8") as f:
        f.write(ascii_image)
```

```
image_to_ascii('111.jpg', 'output.txt')



import os

def load_dictionary(file_path):

    if os.path.exists(file_path):
        with open(file_path, 'r', encoding='utf-8') as f:
            return {line.split()[0]: line.split()[1] for line in f.readlines()}
    else:
        open(file_path, 'w').close()
        return {}

def save_dictionary(file_path, dictionary):

    with open(file_path, 'w', encoding='utf-8') as f:
        for word, meaning in dictionary.items():
            f.write(f"{word} {meaning}\n")

def add_word(dictionary):

    english_word = input("请输入英文单词: ").strip()
    chinese_word = input("请输入中文翻译: ").strip()

    if english_word in dictionary:
        print("该单词已添加到字典库")
    else:
        dictionary[english_word] = chinese_word
        print("单词添加成功")

def query_word(dictionary):

    english_word = input("请输入要查询的英文单词: ").strip()

    if english_word in dictionary:
        print(f"{english_word} 的中文翻译是: {dictionary[english_word]}")
    else:
        print("字典库中未找到这个单词")

def main():
    file_path = 'dictionary.txt'
    dictionary = load_dictionary(file_path)
```

```
    while True:
        print("\n 欢迎使用英文学习词典")
        print("1. 添加单词")
        print("2. 查询单词")
        print("3. 退出")

        choice = input("请选择操作: ").strip()

        if choice == '1':
            add_word(dictionary)
            save_dictionary(file_path, dictionary)
        elif choice == '2':
            query_word(dictionary)
        elif choice == '3':
            print("感谢使用，程序退出。")
            break
        else:
            print("输入有误，请重新选择")

if __name__ == "__main__":
    main()
```

创建用户指定范围的随机数数组，并输出其中的质数

```
import random

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def generate_random_numbers(start, end, count):
    return [random.randint(start, end) for _ in range(count)]

def main():
    start = int(input("请输入起始范围: "))
    end = int(input("请输入结束范围: "))
    count = int(input("请输入生成随机数的数量: "))
```

```
    random_numbers = generate_random_numbers(start, end, count)
    print("生成的随机数数组:", random_numbers)

    primes = [num for num in random_numbers if is_prime(num)]
    print("数组中的质数:", primes)

if __name__ == "__main__":
    main()
```

判断 0~100000 内的素数并输出

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def find_primes_in_range(start, end):
    return [num for num in range(start, end + 1) if is_prime(num)]

def main():
    start = 0
    end = 100000
    primes = find_primes_in_range(start, end)
    print(f"{start}到{end}之间的素数有: {primes}")

if __name__ == "__main__":
    main()
```

判断 num.txt 内的素数并输出

假设 num.txt 文件内容如下:

```
17
23
45
67
```

```
89
101

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def read_numbers_from_file(filename):
    with open(filename, 'r') as file:
        return [int(line.strip()) for line in file]

def main():
    filename = 'num.txt'
    numbers = read_numbers_from_file(filename)
    print("文件中的数字:", numbers)

    primes = [num for num in numbers if is_prime(num)]
    print("文件中的素数:", primes)

if __name__ == "__main__":
    main()
```

添加空格

```
# 方法一: 直接在字符串中使用空格字符
print("Hello", "World")  # 输出: Hello World

# 方法二: 使用多个逗号分隔参数
print("Hello", " ", "World")  # 输出: Hello  World
```

添加换行

```
# 方法一: 使用换行符

print("Hello
World")  # 输出: Hello
                    #         World

# 方法二: 在 print 函数末尾不加任何参数（默认行为）
```

```
print("Hello")
print("World")   # 输出: Hello
                 #       World
```

组合使用空格和换行

```
print("Hello", " ", "World", "
", "This is a new line.")
# 输出: Hello   World
#                 This is a new line.
```