AI 程序设计@NJU

实验 4 Python 控制结构&函数&面向对象&异常处理参考答案

1．计算指定范围内的数幂

```
n = int(input())

i = 2
lst = []
for i in range(2, 101):
    for j in range(2, 101):
          if i**j > n:
              break
          lst.append(i**j)
result = sorted(set(lst))
print(result)
```

2．数组的距离

```
x = input()
y = input()

xlist = x.split()
ylist = y.split()
l = []
for i in xlist:
    for j in ylist:
        d = abs(int(i)-int(j))
        if d not in l:
            l.append(d)
print(min(l))
```

3. 全数字问题

```
def pandigital(nums):
    lst = []
    for item in nums:
        temp = str(item)
        lenth = len(temp)
        num = set(temp)
```

```
        nummax = int(max(num))
        numm = len(num)
        if numm == lenth and nummax == lenth:
            lst.append(item)
    return lst

if __name__ == "__main__":
      lst = pandigital(eval(input()))
      if lst:
          for item in lst:
              print(item)
      else:
          print("not found")
```

4. 验证命题：37 的倍数特性

```
for num in range(100, 1000):
    if num % 37 == 0:
        num_new_1 = num % 100 *10 + num // 100
        num_new_2 = num % 10 * 100 + num // 10
        if num_new_1 % 37 != 0 or num_new_2 % 37 != 0:
            print("It's a false proposition.")
            break
else:
    print("It's a true proposition.")
```

5. 咖啡名称提取

```
def clean_list(lst):
    s = ''
    for item in lst:
        for c in item:
            if c.isalpha():
                 s += c
        s += ','
    cleaned_list = s[:-1].split(',')
    return cleaned_list

if __name__ == "__main__":
    coffee_list = eval(input())
    cleaned_list = clean_list(coffee_list)
    coffee_enum = list(enumerate(cleaned_list, start = 1))
    i = int(input())
    print(coffee_enum[i-1][1])
```

6.两数之和

```
def twonum(n,lst):
    for i in range(1,n):
        if i in lst and n-i in lst:
            return lst.index(i), lst.index(n-i)
    return False

if __name__ == "__main__":
    n = int(input())
    lst = [1,4,5,6,7,8,9,10,11,12,13,15,18,19,20,21,29,34,54,65]
    y = twonum(n,lst)
    if y == False:
        print('not found')
    else:
        print(y)
```

7. 疯狂的游戏玩家

```
    class Person(object):
        Counter = 0
        def __init__(self, name, gender, age, fight_value):
            Person.Counter += 1
            self.name = name
            self.gender = gender
            self.age = age
            self.fig = fight_value

        def battle(self):
            self.fig -= 100

        def practise(self):
            self.fig += 200

        def eat(self):
            self.fig += 80

        def info(self):
            print("I am player {} {}, I have {} fighting
                  value.".format(Person.Counter, self.name, self.fig))

    player1 = Person('xiaohong', 'F', 18, 2000)
    player1.info()
    player1.battle()
    player1.eat()
    player1.info()
    player2 = Person('xiaoming', 'M', 19, 1500)
```

```
    player2.practise()
    player2.battle()
    player2.eat()
    player2.eat()
    player2.info()
```

8. 计算BMI

```
class BMI:
    def __init__(self, height, weight):
        self.bmi = weight / height ** 2
    def printBMI(self):
        print("Your BMI index is {0:.1f} ".format(self.bmi))

class ChinaBMI(BMI):
    def printBMI(self):
        print("Your bmi is {0:.1f}.".format(self.bmi))
        if self.bmi < 18.5:
            print("Skinny")
        elif self.bmi < 24:
            print("Normal")
        elif self.bmi < 27:
            print("Fat")
        elif self.bmi < 30:
            print("Obesity")
        else:
            print("Severe obesity")

if __name__ == "__main__":
     h, w = eval(input())
     x = ChinaBMI(h, w)
     x.printBMI()
```

9. 模仿定义内建函数sum

```
from collections import Iterable

def sum_plus(x):
    if not isinstance(x, Iterable):
          raise TypeError("object of type 'int' has no len()!!!")
    s = 0
    for item in x:
        s += item
    return s

result = sum_plus(eval(input()))
```

```
print(result)
```

10. 筛选数字

```
def isfloat(s):
      if s[-1] == '.':
           s = s.strip('.')
      try:
           float(s)
      except ValueError:
           return False
      return float(s)

if __name__ == "__main__":
      text = input()
      for ch in ',?"!':
          text = text.replace(ch, '')
      words = text.split()
      found = 0
      for word in words:
            if isfloat(word) != False:
                 print(isfloat(word))
                 found = 1
      if found == 0:
            print('Not Found!')
```