

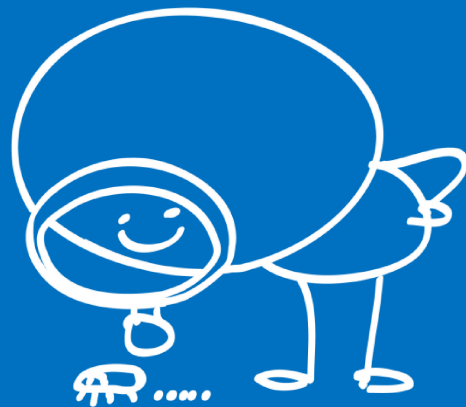
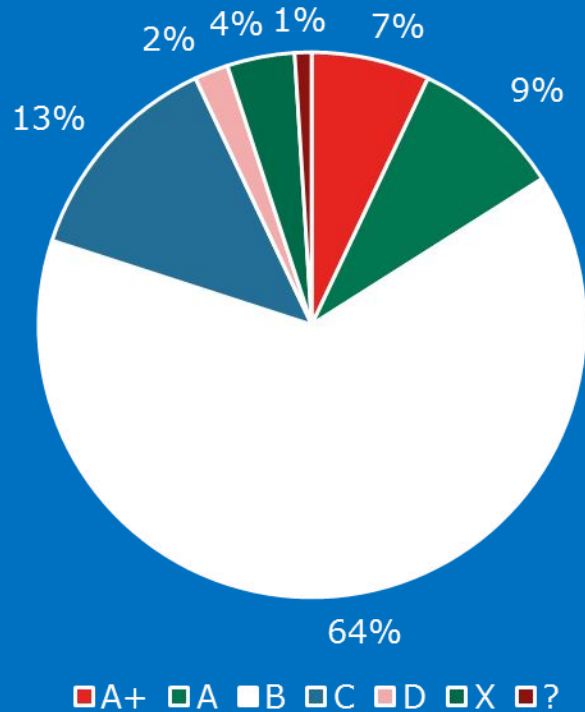
# 上机作业1总结

by 易超



# 上机作业1

成绩分布



评分标准：

- 一般完成的，没有太多错误，即可评为B；
- 优秀的作业，用心努力，并有超出预期的特色的，评为A；（特别优秀的记为A+）
- 错误较多，视情况评为C或者D；
- 延迟交作业的，降低1档；
- 抄袭无分。

## 存在的问题

- 作业压缩文件命名不符合要求（dsa1\_学号\_姓名.7z/.zip/.rar）；
- 作业.py文件里面写的题号等需要注释掉；
- 有的作业语法错误较多；
- 调用函数忘记加括号；
- 在做题时，请看清题目的各项要求，包括功能描述、参数类型、返回值类型等等。

## 第1题

创建一个函数，接受一个字符串和一个正整数n 作为参数，返回把原字符串字符位置向右移动n 个字符的字符串，例如接受的参数是"abcd"和1，返回的字符串是"dabc"；接受的参数是"mnbol"和2，返回的字符串是"olmnb"。

```
def dsal_reverse(s,n):  
    l=len(s)  
    if l==0:return s  
    offset=l-n%l  
    l=s[:offset]  
    r=s[offset:]  
    return r+l
```

## 第2题

创建一个函数，接受一个正整数 $n$ 作为参数，返回  $s=1!+2!+3!+\dots$   
 $\dots+n!$ 的值。

```
def dsa1_factorialSum(n):  
    s=0  
    f=1  
    for i in [j+1 for j in range(n)]:  
        f=f*i  
        s+=f  
    return s
```

## 第3题

创建一个函数，接受两个长度相同的元组，用这两个元组中的所有数据组成一个字典并返回。如元组(1, 2, 3)和("abc", "def", "ghi")生成字典{1:"abc", 2:"def", 3:"ghi"}。

```
def dsal_generateDict(keys, values):  
    return dict(zip(keys, values))
```

## 第4题

创建一个函数，接受一个英文字符串作为参数，返回该字符串的整数表示。如输入"eight-nine"，返回89，输入"one-two-three-four-five"，返回12345。

```
def dsal_en2num(s):  
    d={'one':'1','two':'2','three':'3','four':'4','five'\  
      : '5','six':'6','seven':'7','eight':'8','nine':'9'}  
    return int("".join([d[i] for i in s.split('-')]))
```



## 第5题

创建一个函数，接受两个字符串作为参数，返回两个字符串字符集合的并集。如接受的两个字符串为"abc"和"bcd"，返回set(['a', 'b', 'c', 'd'])。

```
def dsal_getUnion(s1,s2):  
    return set(s1)|set(s2)
```

## 第6题

创建一个函数，接受两个参数y和m，分别表示年和月，返回此年此月的天数。（如大月有31天，小月有30天，而闰年的2月有29天，平年则只有28天，年份如果能被4整除但不能被100整除或者能被400整除为闰年）

```
def dsal_getDays(y,m):
    a=[31,28,31,30,31,30,31,31,30,31,30,31]
    l=[31,29,31,30,31,30,31,31,30,31,30,31]
    def is_leapyear(y):
        if (y%400==0) or ((y%4==0)and(y%100!=0)):
            return True
        return False
    return (l[m-1] if is_leapyear(y) else a[m-1])
```

## 第7题

创建一个函数，接受一个参数 $n(n \geq 100)$ ，判断这个数是否为水仙花数（即满足如果这个数为 $m$ 位数，则每个位上的数字的 $m$ 次幂之和等于它本身，例如 $1^3 + 5^3 + 3^3 = 153$ ， $1^4 + 6^4 + 3^4 + 4^4 = 1634$ ），返回True 或者False。

```
def dsal_isNarcNum(n):  
    s=str(n)  
    l=len(s)  
    sm=0  
    for i in s:  
        sm+=int(i)**l  
    return (True if sm==n else False)
```

## 第8题

创建一个函数，接受一个参数 $\text{max}(\text{max} \geq 1000)$ ，调用第7题编写的函数，求100到 $\text{max}$ 之间的水仙花数，并且输出到一个名为`narcissistic.txt`的文件中，每个数作为一行输出。

```
def dsal_writeNarcNum(max):  
    f=open(r'narcissistic.txt','w')  
    for i in range(100,max+1):  
        if dsal_isNarcNum(i):  
            f.write(str(i)+'\n')  
    f.close()
```

## 第9题

创建一个函数，读入第8题中输出的文件narcissistic.txt，以列表的形式返回文件中的水仙花数，列表的每个元素类型为int型，例如返回[153,370,371]。

```
def dsal_readNarcNum():  
    f=open(r'narcissistic.txt','r')  
    l=[]  
    for line in f:  
        l.append(int(line[:-1]))  
    f.close()  
    return l
```

## 第10题

创建一个函数，接受一个参数n，在屏幕上打印出杨辉三角形的前n行。

```
def dsa1_printTri(n):  
    a,b=[1],[ ]  
    for i in [t+1 for t in range(n)]:  
        print(" ".join([str(t) for t in a]))  
        b.append(1)  
        for j in range(1,i):  
            b.append(a[j-1]+a[j])  
        b.append(1)  
        a,b=b[:],[ ]
```

## 第11题

创建一个People 类，People 的属性有name 和age，两个参数在实例化的时候传给构造器，People 类的方法有getName 和getAge，分别返回姓名和年龄。

```
class People:
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def getName(self):
        return self.name
    def getAge(self):
        return self.age
```



## 第12题

创建一个Student类，Student类继承上一题中的People类，并且增加了属性sno，sno表示学号，和增加了方法getSno。

```
class Student(People):
    def __init__(self, name, age, sno):
        People.__init__(self, name, age)
        self.sno=sno
    def getSno(self):
        return self.sno
```



## 第13题

创建一个Xdict类，继承自dict类(python中的字典类)，使得Xdict类支持dict类的所有操作，并且增加一个方法getKeys，用于返回给定值对应的键的列表。(例如Xdict一个实例为xd=Xdict({2:"a", 3:"a", 4:(2,3)}), 则xd.getKeys("a")返回值为[2,3])

```
class Xdict(dict):  
    def getKeys(self,value):  
        return [k for k,v in self.items() if v==value]
```

**谢谢!**