

# Assignment #3: 语法练习

Updated 1440 GMT+8 Sep 23, 2025

2025 fall, Complied by 韩旭 元培学院

## 说明:

### 1. 解题与记录:

对于每一个题目, 请提供其解题思路 (可选), 并附上使用Python或C++编写的源代码 (确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted)。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。(推荐使用Typora <https://typoraio.cn> 进行编辑, 当然你也可以选择Word。) 无论题目是否已通过, 请标明每个题目大致花费的时间。

2. 提交安排: \*\*提交时, 请首先上传PDF格式的文件, 并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像, 提交的文件为PDF格式, 并且“作业评论”区包含上传的.md或.doc附件。
3. 延迟提交: 如果你预计无法在截止日期前提交作业, 请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业, 以保证顺利完成课程要求。

## 1. 题目

### E28674: 《黑神话：悟空》之加密

<http://cs101.openjudge.cn/pctbook/E28674/>

用时: 15min

#### 思路

需要使用函数chr()和ord()来实现字母和Unicode编码的互换, 从而进行按字母表的移位。从密码反推原码的逆向思维也增加了一定难度, 所以具体公式需要一些细节上的推敲和尝试。

#### 代码

```

def shift(c,num):
    if c.islower():
        return chr(ord('z')-(ord('z')-ord(c)+num)%26)
    else:
        return chr(ord('Z')-(ord('Z')-ord(c)+num)%26)

num=int(input())
string=input()
ans=""
for i in range(len(string)):
    ans+=shift(string[i],num)
print(ans)

```

代码运行截图

#50247721提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def shift(c,num):
    if c.islower():
        return chr(ord('z')-(ord('z')-ord(c)+num)%26)
    else:
        return chr(ord('Z')-(ord('Z')-ord(c)+num)%26)

num=int(input())
string=input()
ans=""
for i in range(len(string)):
    ans+=shift(string[i],num)
print(ans)

```

基本信息

#: 50247721  
 题目: E28674  
 提交人: hanxu  
 内存: 3628kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2025-10-07 17:05:33

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## E28691: 字符串中的整数求和

<http://cs101.openjudge.cn/pctbook/E28691/>

用时: 3min

思路

比较简单，用到切片。

代码

```
a,b=input().split()
a=int(a[:2])
b=int(b[:2])
print(a+b)
```

## 代码运行截图

### #50247787提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
a,b=input().split()
a=int(a[:2])
b=int(b[:2])
print(a+b)
```

基本信息

#: 50247787

题目: E28691

提交人: hanxu

内存: 3588kB

时间: 20ms

语言: Python3

提交时间: 2025-10-07 17:10:46

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## M28664: 验证身份证号

<http://cs101.openjudge.cn/pctbook/M28664/>

用时: 15min

## 思路

题目的逻辑并不复杂，但是细节上有一些难点。这题的索引要用到字典，`dict(zip(keys, values))`是一个很方便的方式。此外，由于数字和'X'混合出现，需要特别注意变量类型。

## 代码

```
keys = list(range(0, 11))
values = ['1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2']
d = dict(zip(keys, values))

N=int(input())
for _ in range(N):
    factor=[7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]
    id=input()
    total=0
    for i in range(len(id)-1):
        total+=factor[i]*int(id[i])
    r=total%11
    if d[r]==id[-1]:
        print("YES")
```

```
else:  
    print("NO")
```

## 代码运行截图

### #50248002提交状态

查看 提交 统计 提问

状态: Accepted

#### 源代码

```
keys = list(range(0, 11))  
values = ['1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2']  
d = dict(zip(keys, values))  
  
N=int(input())  
for _ in range(N):  
    factor=[7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2]  
    id=input()  
    total=0  
    for i in range(len(id)-1):  
        total+=factor[i]*int(id[i])  
    r=total%11  
    if d[r]==id[-1]:  
        print("YES")  
    else:  
        print("NO")
```

#### 基本信息

#: 50248002  
题目: M28664  
提交人: hanxu  
内存: 3924kB  
时间: 24ms  
语言: Python3  
提交时间: 2025-10-07 17:28:32

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## M28678: 角谷猜想

<http://cs101.openjudge.cn/pctbook/M28678/>

用时: 5min

#### 思路

这题需要注意输出细节，比如"/2"之后要化为整数，以及print出来的各个部分之间不能有空格（使用sep选项）等。

#### 代码

```
x=int(input())  
while x>1:  
    if x%2:  
        print(x, "*3+1=", x*3+1, sep="")  
        x=int(x*3+1)  
    else:  
        print(x, "/2=", int(x/2), sep="")  
        x=int(x/2)  
print("End")
```

## 代码运行截图

#50248080提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
x=int(input())
while x>1:
    if x%2:
        print(x, "*3+1=", x*3+1, sep="")
        x=int(x*3+1)
    else:
        print(x, "/2=", int(x/2), sep="")
        x=int(x/2)
print("End")
```

基本信息

#: 50248080  
题目: M28678  
提交人: hanxu  
内存: 3608kB  
时间: 22ms  
语言: Python3  
提交时间: 2025-10-07 17:34:46

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## M28700: 罗马数字与整数的转换

<http://cs101.openjudge.cn/pctbook/M28700/>

用时: 20min

## 思路

罗马数字转整数相对简单，只需要往后看一位判断是加还是减就可以。整数转罗马数字则需要用到贪心，还是从比较大的整数开始一点点从原始数字中减去即可，要特别注意单独讨论900,400,90,40,9,4这几个数字。这里用到两个函数。s.isdigit()判断是否是数字，当然按题目提示用第一位是否在0-9之间也可以 (`'0' <= s[0] <= '9'`)。  
divmod()函数是AI教给我的，它可以同时返回整除值和余数，当然也可以分别求出。

## 代码

```
def roman_to_int(s):
    val={'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000}
    ans=0
    for i, ch in enumerate(s):
        v=val[ch]
        if i+1<len(s) and v<val[s[i+1]]:
            ans-=v
        else:
            ans+=v
    return ans

def int_to_roman(num):
    pairs = [
        (1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'),
        (100, 'C'), (90, 'XC'), (50, 'L'), (40, 'XL'),
```

```

        (10, 'X'),   (9, 'IX'),   (5, 'V'),   (4, 'IV'),
        (1, 'I')
    ]
ans = []
for val, sym in pairs:
    if num == 0:
        break
    q, num = divmod(num, val)
    ans.append(sym * q)
return ''.join(ans)

s=input()
if s.isdigit():
    print(int_to_roman(int(s)))
else:
    print(roman_to_int(s))

```

## 代码运行截图

### #50250005提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

def roman_to_int(s):
    val={'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000}
    ans=0
    for i, ch in enumerate(s):
        v=val[ch]
        if i+1<len(s) and v<val[s[i+1]]:
            ans-=v
        else:
            ans+=v
    return ans

def int_to_roman(num):
    pairs = [
        (1000, 'M'), (900, 'CM'), (500, 'D'), (400, 'CD'),
        (100, 'C'), (90, 'XC'), (50, 'L'), (40, 'XL'),
        (10, 'X'), (9, 'IX'), (5, 'V'), (4, 'IV'),
        (1, 'I')
    ]
    ans = []
    for val, sym in pairs:
        if num == 0:
            break
        q, num = divmod(num, val)
        ans.append(sym * q)
    return ''.join(ans)

s=input()
if s.isdigit():
    print(int_to_roman(int(s)))
else:
    print(roman_to_int(s))

```

基本信息

#: 50250005  
 题目: M28700  
 提交人: hanxu  
 内存: 3684kB  
 时间: 21ms  
 语言: Python3  
 提交时间: 2025-10-07 20:29:51

## 158B. Taxi

\*special problem, greedy, implementation, 1100, <https://codeforces.com/problemset/problem/158/B>

用时：15min

思路

本题用贪心，优先考虑人多的小组，然后用后面人少的小组插补空缺。

代码

```
N=int(input())
L=list(map(int,input().split()))
L.sort(reverse=True)
wait1=0
wait2=0
wait3=0
ans=0
for i in range(N):
    if L[i]==4:
        ans+=1
    elif L[i]==3:
        ans+=1
        wait1+=1
    elif L[i]==2:
        if wait2>0:
            wait2-=1
        else:
            ans+=1
            wait2+=1
    else:
        if wait1: wait1-=1
        elif wait2:
            wait2-=1
            wait1+=1
        elif wait3:
            wait3-=1
            wait2+=1
        else:
            ans+=1
            wait3+=1
print(ans)
```

代码运行截图

342435275	Oct/07/2025 20:17 UTC+8	xuhanecon	B - Taxi	Python 3	Accepted	248 ms	2900 KB
-----------	-------------------------	-----------	----------	----------	----------	--------	---------

## 2. 学习总结和收获

---

- 这次前几题出现了很多**映射**类问题。例如移位加密用 `ord/chr` 把字母映射到 `[0, 25]` 后做模运算。左右端点统一成一种公式，能避免大小写分支出错。
- 第三题的关键是类型与映射：校验位用字符比较（`'x'` 需要 `upper()`），映射用 `dict(zip(...))` 很有帮助。
- 第五题罗马整数互换：首先，这种多功能的题目写函数是很方便的，可以分块解决问题。**罗马→整型是相邻比较**（前小后大则减），**整型→罗马是贪心**（含  $900/400/90/40/9/4$  的组合）。
- 第五题整型→罗马以及第六题Taxi已经是贪心题了。贪心题的关键就是排序、先吃大的、然后需要的话用小的补齐。Taxi这题的特点还有一个**容量不变量**（因为基本都能放满，装箱子就不能这么考虑）：优先放 4；用 3 去“吃” 1；2 与 2 配。
- 调试习惯：先跑样例，再补极端（空串、全大写、`x` 结尾、最小/最大值）。很多 WA 是输入输出细节引起的。