**编译原理实践第1次课-初步实验**

1. **编程环境**

Python 2.7

1. **中缀表达式转后缀表达式**

实现中缀表达式转化为后缀表达式，如： 9 - 5 + 2 转化为 9 5 - 2 +

中缀表达式转后缀表达式的规则：

1.遇到操作数，直接输出；

2.栈为空时，遇到运算符，入栈；

3.遇到左括号，将其入栈；

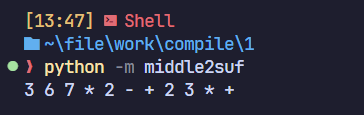
4.遇到右括号，执行出栈操作，并将出栈的元素输出，直到弹出栈的是左括号，左括号不输出； （3） 3

5.遇到其他运算符’+”-”\*”/’时，弹出所有优先级大于或等于该运算符的栈顶元素，然后将该运算符入栈； 1+2+3 左结合

6.最终将栈中的元素依次出栈，输出。

经过上面的步骤，得到的输出既是转换得到的后缀表达式。

**运行结果：**

****

**代码：**

**expression ="3+(6\*7-2)+2\*3"**

**# expression =u"3+2+3"**

**# print(expression[-1])**

**stack = []**

**Ans = []**

**priority = {'+':1, '-':1, '\*':2, '/':2, '(':0}**

**for i in expression:**

**if i.isdigit():**

**Ans.append(i)**

**elif len(stack) == 0 and i in '+-\*/' or i == '(':**

**stack.append(i)**

**elif i == ')':**

**while stack[-1] != '(':**

**Ans.append(stack.pop())**

**stack.pop()**

**else:**

**while len(stack) > 0 and priority[stack[-1]] >= priority[i]:**

**Ans.append(stack.pop())**

**stack.append(i)**

**while len(stack) > 0:**

**Ans.append(stack.pop())**

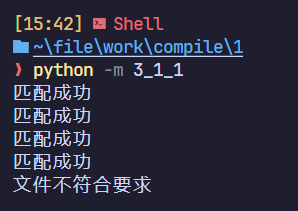
**print(' '.join(Ans))**

1. **正则表达式**

**3.1验证**

（1）验证jpg,jpeg,gif,bmp格式的文件名，大小写均可?

运行结果：



代码：

import re

pattern=re.compile('(.\*)\.(jpg|jpeg|gif|bmp)',re.I)

name = ['a.jpg', 'b.jpeg', 'c.Gif', 'd.bmp', 'e.avi']

for s in name:

m=pattern.match(s)

if m is not None:

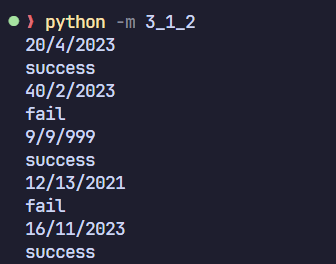
print('匹配成功')

else:

print('文件不符合要求')

（2）匹配日期格式：13/5/2021

运行截图：



代码：

import re

pattern = re.compile('([1-9]|[1-2][0-9]|3[0-1])/([1-9]|1[0-2])/([0-9]{1,4})$')

S = ['20/4/2023', '40/2/2023', '9/9/999', '12/13/2021', '16/11/2023']

for s in S:

print s

if pattern.match(s) is not None:

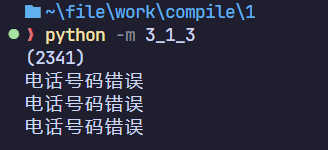
print "success"

else:

print "fail"

（3）电话号码(区号提取)：(0579) 68078800-6852

运行截图



代码

import re

pattern = re.compile('(\((\d{4})\)) (\d{8})-(\d{4})')

number = ["(2341) 82373452-3219", "(3295) 28467123 3255", "(3287) 9482374628-3215", '3218 27332678-2893']

for s in number:

m = re.match(pattern, s)

if m is not None:

print m.groups()[0]

else:

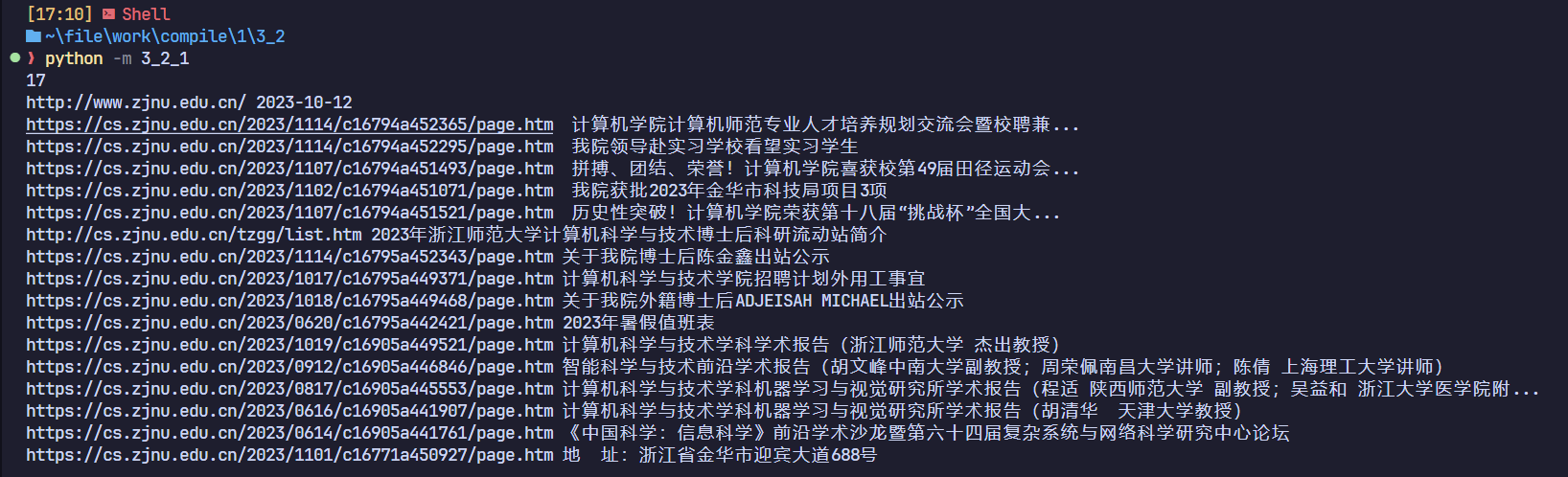
print '电话号码错误'

**3.2提取**

（1）提取list.html中的所有超链接

（2）提取content中的标题与正文

运行截图



代码

import re

pattern = re.compile(r'<a href="(.+?)" target="\_blank">(.|\n)+? <span (.+?)>(.+?)</span>')

content = open("cs\_zjnu.html",'r').read()

matches = pattern.findall(content)

print len(matches)

for m in matches:

print "cs.zjnu.edu.cn" + m[0],m[3]

**参考示例，提取浙师大首页网站的新闻标题和超链接**

**心得总结：**

中缀表达式转换后缀表达式运用了栈的相关知识，并且要对符号进行特判。利用正则表达式可以提取文本串中自己需要的信息十分灵活，学习正则表达式对于检索很有帮助。

**附录**

|  |  |
| --- | --- |
| 代码 | 说明 |
| . | 匹配除换行符以外的任意字符 |
| \w | 匹配字母或数字或下划线或汉字 |
| \s | 匹配任意的空白符 |
| \d | 匹配数字 |
| \b | 匹配单词的开始或结束 |
| ^ | 匹配字符串的开始 |
| $ | 匹配字符串的结束 |

|  |  |
| --- | --- |
| 代码/语法 | 说明 |
| \* | 重复零次或更多次 |
| + | 重复一次或更多次 |
| ? | 重复零次或一次 |
| {n} | 重复n次 |
| {n,} | 重复n次或更多次 |
| {n,m} | 重复n到m次 |

1. **完成实验报告，并提交。**