

《程序设计与算法基础I》在线测试题目说明

电子科技大学 信息与软件工程学院

版本：2018-2019-1（学生版）

实验三（Experiment 3）

1. 翻译（Translation）

题目编号：

07-04

题目内容：

编写程序可以把字母格式的电话号码翻译成数值格式。如果没有电话在身边,参考这里给出的字母在键盘上的对应关系：（2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ。）原始电话号码中的非字母字符（例如数字或标点符号）保持不变：

```
Enter phone number: 1-800-COL-LECT
1-800-265-5328
```

可以假设任何用户输入的字母都是大写字母。

输出范例：

```
Enter phone number: 1-DCKS-A2D-OLED
1-3257-223-6533
```

输入输出：

输入号码是数字和英文的组合，且英文字母全大写。

温馨提示：

如果采用 `printf("2")` 形式输出，因未使用变量，则当前测评框架会由于没有检测到 `printf()` 格式串中 `%` 开头的转换说明而无法有效采集待检输出，故而请使用 `putchar()` 函数输出，或者变通采用 `printf("%d", 2)` 形式输出。

2. 表达式求值 (Expression Evaluation)

题目编号:

07-12

题目内容:

编写程序对表达式求值:

```
Enter an expression: 1+2.5*3
Value of expression: 10.5
```

表达式中的操作数是浮点数, 运算符是+、-、*和/。表达式从左向右求值 (所有运算符的优先级都一样)。

输出范例:

```
Enter an expression: 2+3*4-5/2
Value of expression: 7.5
```

输入输出:

表达式中的所有操作数均为float类型, 运算结果采用最简方式展示。

3. 出现次数 (Number Occurrences)

题目编号:

08-02

题目内容:

修改 8.1 节的程序 repdigit.c, 使其打印一份列表, 显示出每个数字在数中出现的次数。

输出范例:

```
Enter a number: 41271092
Digit:          0  1  2  3  4  5  6  7  8  9
Occurrences:    1  2  2  0  1  0  0  1  0  1
```

输入输出:

- 读入输入数字类型设定为长整型;
- 输出 *Digit:* 以及 *Occurrences:* 之后采用且仅才用Tab实现上下数字对齐。
- 每两个数字之间空2格, 最初一列数字之前以及最后一列数字之后不应有空格。

4. 随机步法（Random Walk）

题目编号：

08-09

题目内容：

编写程序,生成一种贯穿10×10字符数组（初始时全为字符'.'）的“随机步法”。程序必须随机地从一个元素“走到”另一个元素，每次都向上、向下、向左或向右移动一个元素位置。已访问过的元素按访问顺序用字母A到Z进行标记。下面是一个输出示例：

```
A . . . . . . . . . .
B C D . . . . . . . .
. F E . . . . . . . .
H G . . . . . . . . .
I . . . . . . . . . .
J . . . . . . . Z . .
K . . R S T U V Y . .
L M P Q . . . W X . .
. N O . . . . . . . .
. . . . . . . . . .
```

输出范例：

```
A . . . . . . . . . .
B C D . . . . . . . .
. F E . . . . . . . .
H G . . . . . . . . .
I . . . . . . . . . .
J . . . . . . . Z . .
K . . R S T U V Y . .
L M P Q . . . W X . .
. N O . . . . . . . .
. . . . . . . . . .
```

温馨提示：

利用srand函数和rand函数（见程序deal.c）产生随机数，然后查看次数除以4的余数。余数一共有4种可能的值（0、1、2和3），指示下一次移动的4种可能方向。在执行移动之前，需要检查两项内容：一是不能走到数组外面，二是不能走到已有字母标记的位置。只要一个条件不满足，就得尝试换一个方向移动。如果4个方向都堵住了，程序就必须终止了。下面是提前结束的一个示例：

A	B	G	H	I
.	C	F	.	J	K
.	D	E	.	M	L
.	.	W	X	Y	P	Q	.	.	.
.	.	V	U	T	S	R	.	.	.
.
.
.
.

因为Y的4个方向都堵住了，所以没有地方可以放置下一步的Z了。

测评说明：

- 本题目涉及随机数，未采用统一测评框架，而是单独撰写测评脚本。测评脚本不检查相邻两列之间到底有多少空格，只按逻辑打分。但是测评结果输出统一按照相邻两列之间空1格的形式展示。共测试5次，每次20分。

5. 加密 (Encryption)

题目编号:

08-15

题目内容:

已知的最古老的一种加密技术是凯撒加密 (得名于 Julius caesar)。该方法把一条消息中的每个字母用字母表中固定距离之后的那个字母来替代。(如果越过了字母Z, 会绕回到字母表的起始位置。例如, 如果每个字母都用字母表中两个位置之后的字母代替, 那么Y就被替换为A, Z就被替换为B。)编写程序用凯撒加密方法对消息进行加密。用户输入待加密的消息和移位计数 (字母移动的位置数目):

```
Enter message to be encrypted: Go ahead, make my day.  
Enter shift amount (1-25): 3  
Encrypted message: Jr dkhdg, pdh pb gdb.
```

注意,当用户输入26与移位计数的差值时, 程序可以对消息进行解密:

```
Enter message to be encrypted: Jr dkhdg, pdh pb gdb.  
Enter shift amount (1-25): 23  
Encrypted message: Go ahead, make my day.
```

不是字母的那些字符不要改动。此外, 加密时不要改变字母的大小写。

输出范例:

```
Enter message to be encrypted: Hello, world!  
Enter shift amount (1-25): 3  
Encrypted message: Koor, zruog!  
  
Enter message to be encrypted: Koor, zruog!  
Enter shift amount (1-25): 23  
Encrypted message: Hello, world!
```

输入输出:

输入为80个字符以内的任意字符串。

温馨提示:

为了解决前面提到的绕回问题, 可以用表达式 $((ch - 'A') + n) \% 26 + 'A'$ 计算大写字母的密码, 其中ch存储字母, n存储移位计数。(小写字母也需要一个类似的表达式。)

评分规则：

- 编译通过20分
- 格式规范10分（忽略换行，以输出变量为分隔，分段检查输出内容）
- 测试用例70分

注意事项：

- C语言文件请采用utf8编码；
- 默认最后一个printf格式语句结束时都需要采用\n换行；
- 尽量避免printf语句换行（如下例）；除非格式全对，否则会被认为全错。

```
printf("a: %d\n"  
"b: %d\n"  
"c: %d\n"  
"d: %d\n",a,b,c,d);
```

////////////////////////////////////

编辑：SynX.Rumba Group

审校：张翔

创建：2018.10.02

修订：2018.11.24

////////////////////////////////////