



3.1 若按教科书 3.1.1 节中图 3.1(b)所示铁道进行车厢调度（注意：两侧铁道均为单向行驶道），则请回答：

(1) 如果进站的车厢序列为 123，则可能得到的出站车厢序列是什么？

(2) 如果进站的车厢序列为 123456，则能否得到 435612 和 135426 的出站序列，并请说明为什么不能得到或者如何得到（即写出以 ‘S’ 表示进栈和以 ‘X’ 表示出栈的栈操作序列）。

解：(1) 123 231 321 213 132

(2) 可以得到 135426 的出站序列，但不能得到 435612 的出站序列。因为 4356 出站说明 12 已经在栈中，1 不可能先于 2 出栈。

3.3 写出下列程序段的输出结果（栈的元素类型 SElemType 为 char）。

```
void main()
{
    Stack S;
    char x, y;
    InitStack(S);
    x = 'c'; y = 'k';
    Push(S, x); Push(S, 'a'); Push(S, y);
    Pop(S, x); Push(S, 't'); Push(S, x);
    Pop(S, x); Push(S, 's');
    while(!StackEmpty(S)) { Pop(S, y); printf(y); }
    printf(x);
}
```

解：stack

3.4 简述以下算法的功能（栈的元素类型 SElemType 为 int）。

```

(1) status algo1(Stack S)
{
    int i, n, A[255];
    n=0;
    while(!StackEmpty(S)) { n++; Pop(S, A[n]); }
    for(i=1; i<=n; i++) Push(S, A[i]);
}

(2) status algo2(Stack S, int e)
{
    Stack T; int d;
    InitStack(T);
    while(!StackEmpty(S)) {
        Pop(S, d);
        if(d!=e) Push(T, d);
    }
    while(!StackEmpty(T)) {
        Pop(T, d);
        Push(S, d);
    }
}

```

解：(1) 栈中的数据元素逆置 (2) 如果栈中存在元素 e，将其从栈中清除

3.7 按照四则运算加、减、乘、除和幂运算(\uparrow)优先关系的惯例，并仿照教科书 3.2 节例 3-2 的格式，画出对下列算术表达式求值时操作数栈和运算符栈的变化过程：

$$A-B \times C/D+E \uparrow F$$

解：BC=G G/D=H A-H=I $E^{\wedge}F=J$ I+J=K

步骤	OPTR 栈	OPND 栈	输入字符	主要操作
1	#		<u>A</u> -B*C/D+E [^] F #	PUSH(OPND, A)
2	#	A	<u>-</u> B*C/D+E [^] F#	PUSH(OPTR, -)
3	#-	A	<u>B</u> *C/D+E [^] F#	PUSH(OPND, B)
4	#-	A B	<u>*</u> C/D+E [^] F#	PUSH(OPTR, *)
5	#-*	A B	<u>C</u> /D+E [^] F#	PUSH(OPND, C)
6	#-*	A B C	<u>/</u> D+E [^] F#	Operate(B, *, C)
7	#-	A G	<u>/</u> D+E [^] F#	PUSH(OPTR, /)
8	#-/	A G	<u>D</u> +E [^] F#	PUSH(OPND, D)
9	#-/	A G D	<u>+</u> E [^] F#	Operate(G, /, D)
10	#-	A H	<u>+</u> E [^] F#	Operate(A, -, H)
11	#	I	<u>+</u> E [^] F#	PUSH(OPTR, +)
12	#+	I	<u>E</u> [^] F#	PUSH(OPND, E)
13	#+	I E	<u>^</u> F#	PUSH(OPTR, ^)
14	#+^	I E	<u>F</u> #	PUSH(OPND, F)

15	#+^	I E F	<u>#</u>	Operate(E, ^, F)
16	#+	I J	#	Operate(I, +, J)
17	#	K	#	RETURN