实验二

利用栈，计算后缀表达式的值

输入23+，输出5

输入145\*+3/3-，输出4

void main()

{

stack S;

int i = 0, sum;

int x, y;

char s[100];

InitStack(&S);

scanf("%s", s);

while (s[i] != '\0')

{

if (s[i] <= '9'&&s[i] >= '0')

Push(&S, s[i] - 48);

if (s[i] == '+')

{

x = Pop(&S, &x);

y = Pop(&S, &y);

sum = x + y;

Push(&S, sum);

}

if (s[i] == '-')

{

x = Pop(&S, &x);

y = Pop(&S, &y);

sum = y - x;

Push(&S, sum);

}

if (s[i] == '\*')

{

x = Pop(&S, &x);

y = Pop(&S, &y);

sum = x \* y;

Push(&S, sum);

}

if (s[i] == '/')

{

x = Pop(&S, &x);

y = Pop(&S, &y);

sum = y / x;

Push(&S, sum);

}

i++;

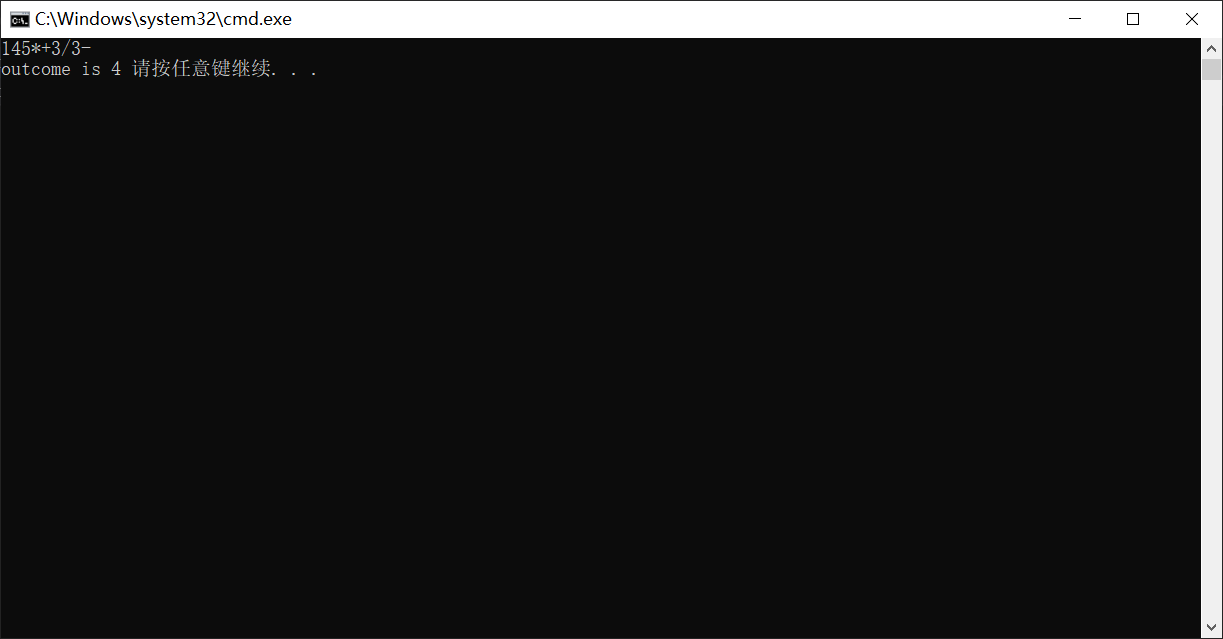
}

printf("outcome is %d ", sum);

}

}

运行效果



总结：后缀表达式的求值规则为：从左到右扫描后缀表达式，如果遇到操作数，将其压入栈中，如果遇到操作符，则从栈中弹出两个操作数，计算结果，然后把结果入栈，直到遍历完后缀表达式，则计算完成，此时的栈顶元素即为计算结果。