# 有效的括号

给定一个只包括 `'('，')'，'{'，'}'，'['，']'` 的字符串，判断字符串是否有效。

有效字符串需满足：

1. 左括号必须用相同类型的右括号闭合。
2. 左括号必须以正确的顺序闭合。

注意空字符串可被认为是有效字符串。

示例 **1:**

输入: `"()"`

输出: `true`

示例 **2:**

输入: `"()[]{}"`

输出: `true`

示例 **3:**

输入: `"(]"`

输出: `false`

示例 **4:**

输入: `"([)]"`

输出: `false`

示例 **5:**

输入: `"{[]}"`

输出: `true`

```
typedef int STDataType;
typedef struct Stack
{
    STDataType* _data;
    //int _top; // 栈顶
    int _size;
    int _capacity; // 容量
}Stack;
```

```c
//检查容量
void checkCapcity(Stack* ps)
{
    if (ps->_size == ps->_capacity)
    {
        int newCap = ps->_capacity == 0 ? 1 : 2 * ps->_capacity;
        ps->_data = (STDataType*)realloc(ps->_data, sizeof(STDataType)*newC
ap);
        ps->_capacity = newCap;
    }
}
// 初始化栈
void StackInit(Stack* ps)
{
    if (ps == NULL)
        return;
    ps->_data = NULL;
    ps->_size = ps->_capacity = 0;
}
// 入栈
void StackPush(Stack* ps, STDataType data)
{
    if (ps == NULL)
        return;
    checkCapcity(ps);
    ps->_data[ps->_size++] = data;
}
// 出栈
void StackPop(Stack* ps)
{
    if (ps == NULL || ps->_size == 0)
        return;
    ps->_size--;

}
// 获取栈顶元素
STDataType StackTop(Stack* ps)
{
    return ps->_data[ps->_size - 1];
}

// 检测栈是否为空，如果为空返回非零结果，如果不为空返回 0
int StackEmpty(Stack* ps)
{
```

```
    if (ps == NULL||ps->_size==0)
        return 1;
    else
        return 0;
}

bool isValid(char * s){
    char map[3][2]={{'(',')'},{'[',']'},{'{','}'}};
    Stack st;
    StackInit(&st);
    while(*s)
    {
        int flag=0;
        //判断当前字符是否为左括号
        for(int i=0;i<3;i++)
        {
            if(*s==map[i][0])
            {
                StackPush(&st,*s);
                flag=1;
                s++;
                break;
            }
        }
        //如果当前字符为右括号
        if(flag==0)
        {
            if(StackEmpty(&st))
            {
                return false;
            }
            //取出左括号
            char str=StackTop(&st);
            StackPop(&st);
            //判断是哪一个右括号与取出的左括号匹配
            for(int i=0;i<3;i++)
            {
                if(*s==map[i][1])
                {
                    if(map[i][0]==str)//左右匹配正确，继续判断下一个字符
                    {
                        s++;
                        break;
                    }
```

```
                    else //栈顶元素与之不匹配  返回 false;
                        return false;
                }
            }
        }
    }
    if(StackEmpty(&st))
        return true;
    return false;
}
```