**《数据结构》**

**课程设计报告**

|  |  |
| --- | --- |
| 课程名称： | 《数据结构》课程设计 |
| 课程设计题目： | 魔王语言 |
| 姓 名： | 应宇杰 |
| 院 系： | 计算机 |
| 专 业： | 计算机科学与技术 |
| 班 级： | 19052321 |
| 学 号： | 19151633 |
| 指导教师： | 葛瑞泉 |

2020年12月1日

1. **需求分析**

#### 功能需求：

#### 此魔王语言解释系统的具体作用是:针对魔王所说的人听不懂的语言，用符合题目要求的规则进行输出（翻译），解释成人类能听得懂的语言。在此用到数据结构所学习的栈的思想和队列的思想进行设计，将魔王所说的语言按照一定的规则进行一系列的入栈、出栈、入队、出队的操作，再按照魔王的语言对应的汉字进行输出。

#### 界面需求：

输入魔王语言，输出翻译的英文文字

1. **概要设计**

Main

基本操作

栈的操作

输出

run（运行

print（显示）打印魔王语言

Push（入栈）

Gettop（出栈）

输入魔王语言

输入结束符‘#‘

#### 接口设计

void InitStack(sqstack \*s)//重置栈

void push(sqstack \*s,char e)//输入栈

char getTop(sqstack \*s)//输出栈

#### 数据结构设计

typedef struct sqstack//创建一个栈

{

char \*base;

char \*top;

int stacksize;

}sqstack;

1. **详细设计**

#include<stdio.h>

#include<stdlib.h>

#define STACK\_INIT\_SIZE 100;

#define STACKINCREMENT 10;

#define null 0;

#include<queue>

using namespace std;

typedef struct sqstack//创建栈

{

char \*base;

char \*top;

int stacksize;

}sqstack;

void InitStack(sqstack \*s)//重置栈

{

s->base =(char\*) malloc(1009 \* sizeof(char));

s->top=s->base;

s->stacksize=STACK\_INIT\_SIZE;

}

void push(sqstack \*s,char e)

{

\*(s->top)=e;

s->top++;

}

char getTop(sqstack \*s)//输出栈

{

char e;

if(s->top==s->base)

return -1;

e=\*(s->top-1);

s->top--;

return e;

}

int main()

{

queue<char> q;

sqstack \*s;

int m,j,number=0;

int i=0;

s=(sqstack\*)malloc(sizeof(sqstack));

InitStack(s);

//先输入魔王语言，保存在一维数组demon

printf("请输入魔王语言:(最后输入#号结束) ");

char demon[100];

char c;

while(c!='#')

{

scanf("%c",&c);

//printf("%d",c);

if(c<=0)

{

printf("你输入了全角字符，程序退出");

return -1;

}

demon[i]=c;

i++;

}

i--;

//遍历数组，将数组中括号内元素入栈，插入相应首字母

for(j=0;j<i-1;j++)

{

if(demon[j]=='(')

{

m=j;

push(s,demon[m+1]);

while(demon[j+2]!=')')

{

push(s,demon[j+2]);

push(s,demon[m+1]);

j++;

}

}

}

int p=0;

while(p<=i)

{

char ch=demon[p];

if(ch=='#') break;//输入#结束

if(ch>='a'&&ch<='z')//入队列

{

q.push(ch);

ch=q.front();

q.pop();

printf("%c",ch);

//printf("%c",ch);

}

else if(ch=='A')

{

printf("%s"," sae ");

}

else if(ch=='B')

{

printf("%s"," tsaedsae ");

}else if(ch=='(')

{

while(s->top!=s->base)//出栈入队列

{

number++;

q.push(getTop(s));

ch=q.front();

q.pop();

printf("%c",ch);

//printf("%c",getTop(s));

}

p=p+number/2+1;

}

p++;

}

return 0;

}

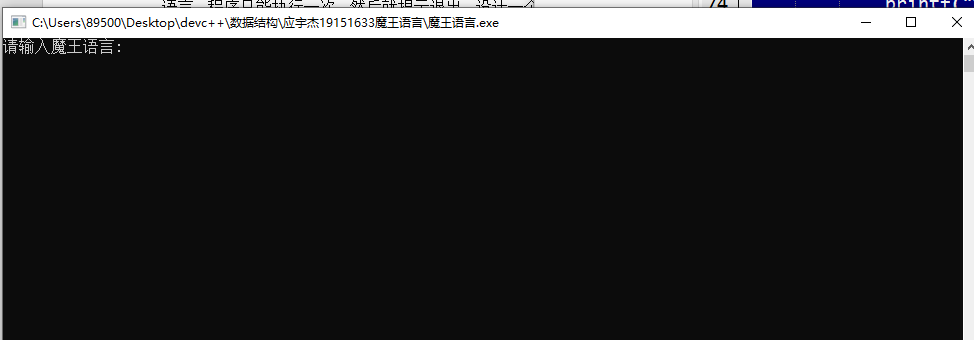
1. **调试分析**

(1）括号的匹配问题。输入不匹配的括号之后不能判断语言是否符合魔王语言规则，，当循环出栈输出元素‘(’时，另把字符输入进队列，继续出栈输出元素‘)’时，括号不匹配，输出“魔王语言错误!”。

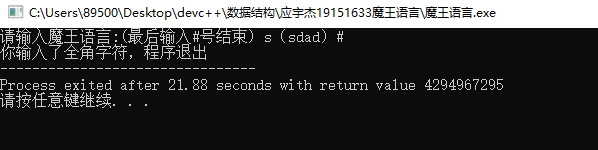
(2）程序执行次数不能被有限次地控制。当要解释魔王语言时，输入魔王的语言，程序只能执行一次，然后就提示退出，设计一个总体循环控制整个程序执行，提示“再次输入魔王语言（按#键退出）:”。

（3）注意输入全角还是半角字符，全角字符是2个字节，会使程序出错。

1. **用户手册**
2. 本程序的执行文件为：魔王语言（final）.exe
3. 进入演示程序后，将显示如下的界面



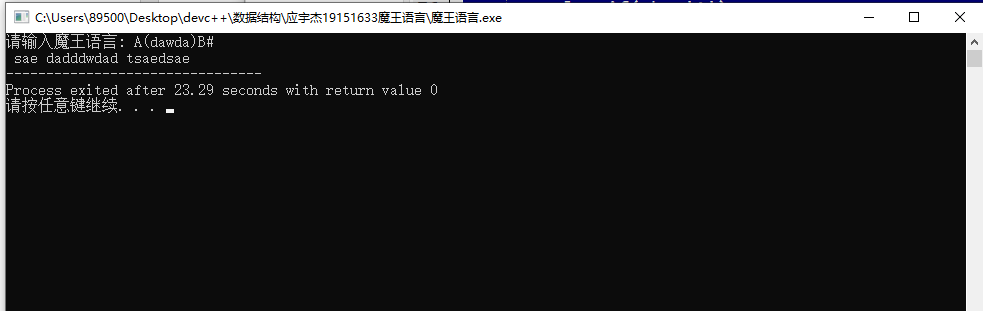
3、如果输入全角字符，会退出并且提示你输入了全角字符



4、输入魔王语言并在最后输入‘#’然后程序会输出翻译后的英文字母。

1. **测试结果**

由于魔王语言是输入后立马出现的，这里截图表示一下



PS：详细的运行结果请打开魔王语言.exe文件

1. **附录**

源程序文件名清单：魔王语言.cpp，魔王语言.exe