一、代码实现：

#include <stdio.h>

#include <stdlib.h>

typedef struct

{

float \*base;

float \*top;

int StackSize;

}Stack;

void Createstack(Stack &s)

{

s.base=(float \*)malloc(100 \* sizeof(float));

s.top=s.base;

s.StackSize=100;

}

float top(Stack s)

{

if(s.top==s.base)

{

return -1;

printf("error!\n");

}

return \*(s.top-1);

}

void push(Stack &s,float elem)

{

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

{

s.base=(float \*)realloc(s.base,(s.StackSize+10)\*sizeof(float));

s.top=s.base+s.StackSize;

s.StackSize+=10;

}

\*s.top++ = elem;

}

void pop(Stack &s)

{

if(s.top==s.base)

{

printf("error!\n");

return ;

}

s.top--;

}

int operate(char c)

{

switch(c)

{

case '+' :return 1;

case '-' :return 1;

case '\*' :return 1;

case '/' :return 1;

default :return 0;

}

}

float cal(float a,float b,char c)

{

switch(c)

{

case '+' :return a+b;

case '-' :return a-b;

case '\*' :return a\*b;

case '/' :return a/b;

}

}

int main()

{

float a,b,result;

Stack s;

Createstack(s);

char c1='0';

printf("输入后缀表达式：");

while(c1!='A')

{

c1=getchar();

if(!operate(c1))

{

push(s,(float) (c1-48));

}

else

{

b=top(s);

pop(s);

a=top(s);

pop(s);

result=cal(a,b,c1);

push(s,result);

}

}

printf("%f\n",result);

return 0;

}

二、实验总结：

本次实验课对于之前的知识进行了巩固，对于本次实验中的嵌套、遍历和返回和其他人进行了深入的探讨，在其他人的帮助下完成了对实验目标的实现。虽然过程并不顺利，但成功完成了预期目标，对所要求掌握的知识有了更深入的认识和了解，收获较多。