南京信息工程大学 数据结构实验(实习)报告

实验(实习)名称 **栈的应用** 实验(实习)日期 2021.10 得分 指导老师 马瑞

系 **计软院** 专业  **计科** 班级 20（1） 姓名 鲁哲豪 学号 202083290400

一、实验目的



二、 实验内容与步骤





1.

#include<cstdio>

#include<cstdlib>

#define SIZE 100

using namespace std;

struct stack{

    int data[SIZE];

    int top;

    stack():top(0){}

};

int bug(int n){

    printf("Error %d: stack is ",n);

    switch(n){

    case 1:

        printf("full!\n");

        break;

    case 2:

        printf("empty!\n");

        break;

    }

    return n;

}

void push(stack& s,int n){

    if(s.top>=SIZE)

        throw bug(1);

    s.data[s.top++]=n;

}

int pop(stack& s){

    if(s.top<=0)

        throw bug(2);

    return s.data[--s.top];

}

void show(stack& s){

    if(s.top<=0)

        throw bug(2);

    const char d[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

    while(s.top)

        putchar(d[pop(s)]);

    putchar('\n');

}

stack convert(int n,int base){

    stack s;

    while(n){

        push(s,n%base);

        n/=base;

    }

    return s;

}

int main(){

    srand(0);

    int n=rand();

    printf("%d\n",n);

    stack s;

    s=convert(n,10);

    printf("Base 10: ");

    show(s);

    s=convert(n,2);

    printf("Base 2 : ");

    show(s);

    s=convert(n,8);

    printf("Base 8 : ");

    show(s);

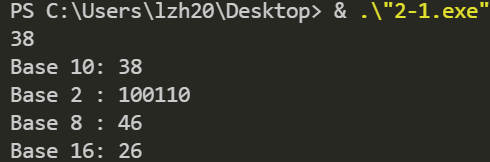
    s=convert(n,16);

    printf("Base 16: ");

    show(s);

    return 0;

}



2.

#include<cstdio>

#define SIZE 100

using namespace std;

struct stack{

    char data[SIZE];

    int top;

    stack() :top(0){}

};

int bug(int n){

    printf("Error %d: stack is ",n);

    switch(n){

    case 1:

        printf("full!\n");

        break;

    case 2:

        printf("empty!\n");

        break;

    }

    return n;

}

void push(stack& s,char n){

    if(s.top>=SIZE)

        throw bug(1);

    s.data[s.top++]=n;

}

stack set(char\* st){

    stack s;

    for(int i=0;st[i];i++)

        push(s,st[i]);

    return s;

}

char pop(stack& s){

    if(s.top<=0)

        throw bug(2);

    return s.data[--s.top];

}

// void show(stack s){

//  if(s.top<=0)

//      throw bug(2);

//  stack reversed;

//  while(s.top)

//      push(reversed,pop(s));

//  while(reversed.top)

//      putchar(pop(reversed));

//  putchar('\n');

// }

bool palindrome(stack s){

    bool odd=s.top%2;

    stack a;

    int n=s.top/2;

    for(int i=0;i<n;i++)

        push(a,pop(s));

    if(odd) pop(s);

    while(s.top)

        if(pop(a)!=pop(s))

            return false;

    return true;

}

int main(){

    char st[100];

    scanf("%s",st);

    stack s=set(st);

    // show(s);

    if(palindrome(s))

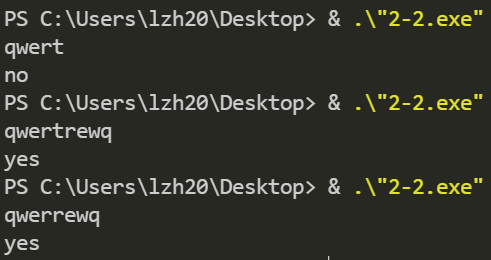
        printf("yes");

    else

        printf("no");

    return 0;

}



三、 实验心得

Debug时，为了确定bug种类，我使用了抛出异常的功能。

本次两个题，共用了一部分代码(pop/push等函数)。但是数据类型不同，复用前需进行一些修改。可以通过c++模板/泛型的方式拓展功能。

以上功能为c++特性，但均没用到面向对象（类）。