

Lab 3 利用栈实现进制转换

姓名：阳焘

学号：2016213391

班级：2016215114

1. 问题描述：

利用栈实现如下两种类型的进制转换：

- 十进制转换为二进制；
- 二进制转换为十进制；

要求：从键盘输入要转换的十进制或二进制数，如输入的为十进制数则

以'\$'结束，如输入的为二进制数则以'#'结束，利用结束符'\$' '#'自动识别要

做的转换类型，并将对应的转换结果输出。

2. 程序结构

- 主程序函数 `main()`
- 判断转换种类函数 `void determine(char *input)`
函数参数：字符串 `*input`，函数返回值：`void`
- 十进制转二进制函数
`LinkStack conversion1(int dec, LinkStack top)`

函数参数：整型参数 dec、LinkStack类型指针 top

函数返回值：LinkStack

- 二进制转十进制函数 void conversion2(LinkStack top)

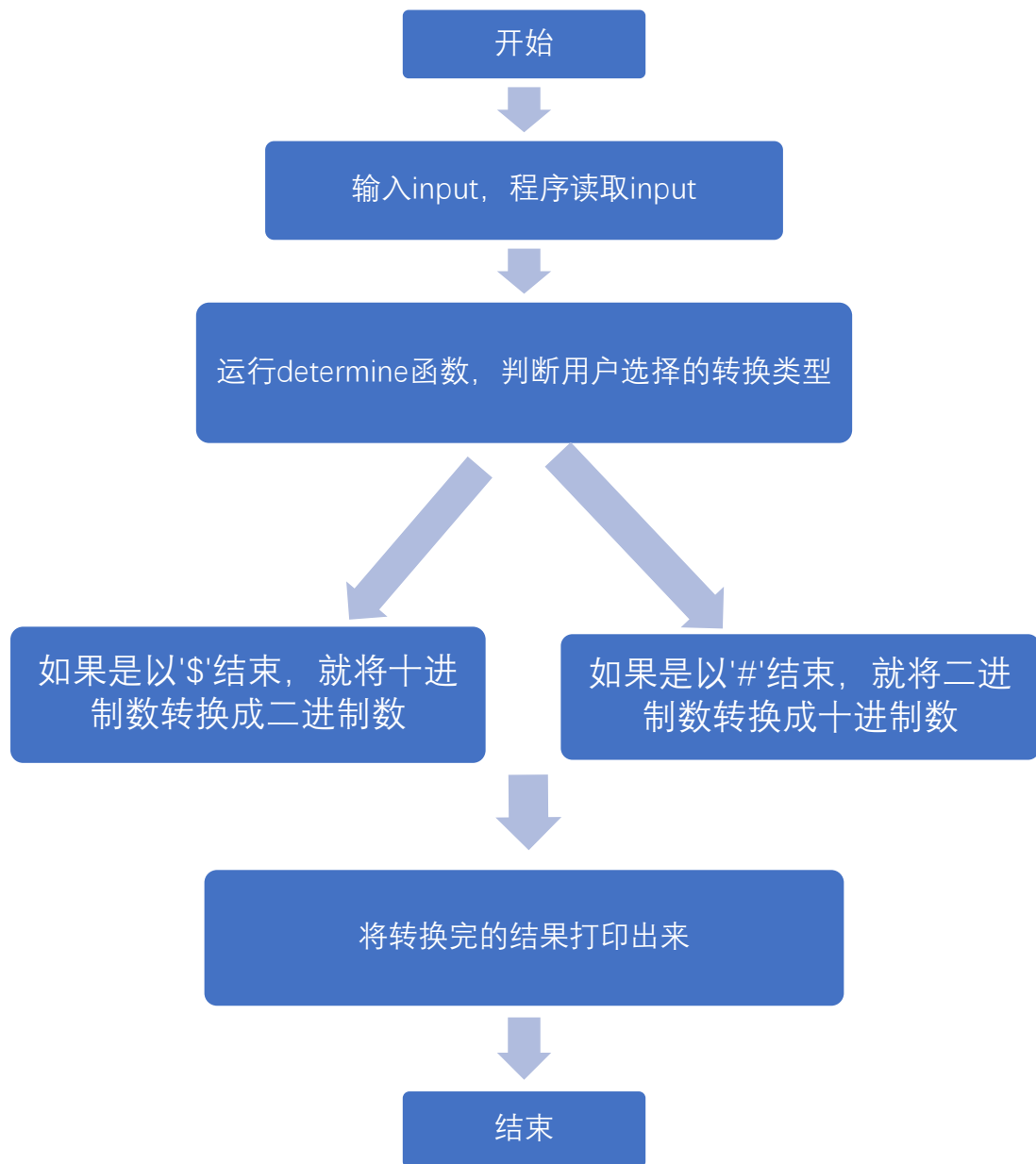
函数参数：LinkStack类型指针 top，函数返回值：LinkStack

- 输出二进制数函数void printStack(LinkStack top)

函数参数：LinkStack类型指针 top，函数返回值：void

- 阶乘函数 int factorial(int n)

函数参数：整型参数 n，函数返回值：整型参数



3. 源码

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXSIZE 1024
char input[MAXSIZE];
int digit = 0, outcome = 0; //digit表示位数, outcome表示二进制转十进制的结果

typedef struct node //栈结构
{
    char data;
    struct node *next;
} LStackNode, *LinkStack;
LinkStack top = NULL; //初始化栈

LinkStack conversion1(int dec, LinkStack top) //将十进制数转换为二进制
{
    int q, r; //q为商, r为余数
    LStackNode *p;
    p = (LStackNode *)malloc(sizeof(LStackNode)); //申请一个新的结点
    if(!p) //若空间不足, 申请失败, 退出程序
    {
        printf("空间不足\n");
        return NULL;
    }
}
```

```

    if(dec == 1)    //如果参数为1, 令'1'入栈
    {
        p->data = '1';
        p->next = top;
        top = p;
    }
    else    //如果参数大于1, 令余数入栈, 商作为新的参数转化为二进制
    {
        q = dec / 2;
        r = dec % 2;
        if(r == 1) p->data = '1';
        if(r == 0) p->data = '0';
        p->next = top;
        top = p;
        top = conversion1(q, top);
    }
    return top; //返回2进制数存放的栈的栈顶指针
}

void printStack(LinkStack top){ //2进制数出栈并输出在屏幕上
    if(!top){ //如果栈为空, 返回
        printf("\n");
        return;
    }
    else{ //出栈并释放空间
        LinkStack p;
        printf("%c", top->data);
        p = top;
        top = top->next;
        free(p);
        printStack(top);
    }
}

int factorial(int n){ //阶乘函数, 辅助2进制转化为10进制, n表示数字所在
    位数, 0为个位依此类推
    if(n == 0){
        return 1;
    }
    else{
        return 2*factorial(n-1);
    }
}

void conversion2(LinkStack top){
    if(!top){
        return;
    }
    else{

```

```

        LinkStack p;
        if(top->data == '1') outcome = outcome +
factorial(digit);
        digit++; //位数加一
        p = top;
        top = top->next;
        free(p);
        conversion2(top);
    }
}

void determine(char *input) //判断是十进制转二进制还是二进制转十进制
{
    int num, boolean = 0;
    if(strlen(input)<2){ //不足2位无效
        printf("输入无效!\n");
        return;
    }
    else if(input[strlen(input)-1]=='$') //若以'$'结尾
    {
        for(int i = 0; i <= strlen(input)-2; i++){ //判断每一位是否
都在0到9之间
            if(input[i]<='0' || input[i]>='9'){
                boolean = 1; //出现不在0到9之间到字符, boolean记为1
                break;
            }
        }
        if(boolean){ //若出现非法字符, 退出程序
            printf("输入无效!\n");
            return;
        }
        char input_copy[MAXSIZE]; //则去掉字符串最后一位
        for(int i = 0; i <= strlen(input)-2; i++){
            input_copy[i] = input[i];
        }
        num = atoi(input_copy); //将该十进制字符串转换为整型
        top = conversion1(num, top); //十进制转换成二进制
        if(!top) //若为空, 空间不足, 退出程序
            return;
        printStack(top); //若不为空, 输出栈内储存的2进制数
    }

    else if(input[strlen(input)-1]=='#') //若以'#'结尾
        for(int i = 0; i <= strlen(input)-2; i++){ //判断每一位是否

```

为0或1

```
        if(input[i] != '0' && input[i] != '1'){
            boolean = 1;
            break;
        }
    }
    if(boolean){//出现非法字符, 退出程序
        printf("error!\n");
        return;
    }
    for(int i = 0; i <= strlen(input)-2; i++){//将2进制数存入
```

栈内

```
        LStackNode *p;
        p = (LStackNode *)malloc(sizeof(LStackNode));
        if(!p){//若空间不足, 申请失败, 退出程序
            printf("空间不足\n");
            return;
        }
        p->data = input[i];
        p->next = top;
        top = p;
    }
    conversion2(top);//2转10
    printf("%d\n", outcome);//将10进制结果输出
}

else{//若不以'#'或'$'结尾, 错误, 退出程序
    printf("error!\n");
}

}

int main() {
    printf("1.如果你想将十进制数转换成二进制数, 请将输入的十进制数以'$'
结束.\n2.如果你想将二进制数转换成十进制数, 请将输入的二进制数以'#'结
束.\n\n请输入数字:\n");
    scanf("%s", input);//从键盘输入字符串
    determine(input);
    return 0;
}
```

4. 测试结果

- 1.如果你想将十进制数转换成二进制数，请将输入的十进制数以'\$'结束。
- 2.如果你想将二进制数转换成十进制数，请将输入的二进制数以'#'结束。

请输入数字：

- 1.如果你想将十进制数转换成二进制数，请将输入的十进制数以'\$'结束。
- 2.如果你想将二进制数转换成十进制数，请将输入的二进制数以'#'结束。

请输入数字：

55\$

- 1.如果你想将十进制数转换成二进制数，请将输入的十进制数以'\$'结束。
- 2.如果你想将二进制数转换成十进制数，请将输入的二进制数以'#'结束。

请输入数字：

55\$

110111

Program ended with exit code: 0

- 1.如果你想将十进制数转换成二进制数，请将输入的十进制数以'\$'结束。
- 2.如果你想将二进制数转换成十进制数，请将输入的二进制数以'#'结束。

请输入数字：

110111#

- 1.如果你想将十进制数转换成二进制数，请将输入的十进制数以'\$'结束。
- 2.如果你想将二进制数转换成十进制数，请将输入的二进制数以'#'结束。

请输入数字：

110111#

55

Program ended with exit code: 0