

# 实验报告：模拟洗牌

## 实验题目：

模拟洗牌

## 实验要求：

1. 以数组表示一副牌
2. 实现切牌操作
3. 实现洗牌操作

## 编程思路：

1. 以数组 1-13、21-33、41-53、61-73 表示四种花色的牌，更改均为数组内更改
2. 编写函数，分别实现洗牌、切牌、打分、界面设置、花色判断、当前牌打印

## 核心代码：

1. 洗牌

```
int Wash(int *poke){//洗牌
    int a[N];//暂时存储的数组
    for(int i=0;i<(N+1)/2;i++){//前半部分为奇数组
        a[i]=poke[i*2];
    }
    for(int i=(N+1)/2;i<N;i++){//后半部分为偶数组
        a[i]=poke[(i-((N+1)/2))*2+1];
    }
    for(int i=0;i<N;i++){//返回给牌组
        poke[i]=a[i];
    }
    return 0;
}
```

2. 切牌

```
int Devide(int *poke){//切牌,返回切分点
    srand(rand()+165+time(NULL));//随机化
    int n=rand()%(N-1)+1;//n 为切分点
    int a[N];//暂时存储的数组
    for(int i=0;i<N-n;i++){
        a[i]=poke[i+n];
    }
    for(int i=N-n;i<N;i++){
        a[i]=poke[i-N+n];
    }
    for(int i=0;i<N;i++){
        poke[i]=a[i];
    }
    return n;
}
```

3. 花色判断

```

int group(int a){//以四种返回值代表四种花色
    if(a>=1&&a<=13) return 1;
    if(a>=21&&a<=33) return 2;
    if(a>=41&&a<=53) return 3;
    if(a>=61&&a<=73) return 4;
    return -1;
}
4. 打分
int Mark(int *poke){//打分函数，返回扣分总数
    int m=0;//m 为扣分总数
    for(int i=0;i<N-2;i++){//花色相同，扣分总数加一

        if((group(poke[i])==group(poke[i+1]))&&(group(poke[i])==group(poke[i+2])))
            m++;
    }
    return m;
}

```

#### 实验结果：

1. 第 3 次洗牌，第 5 次切牌：

切分点为：14

当前扑克为：2 12 27 42 50 65 73 7 22 30 45 53 72 6 21 29 44 52 67 1 9 24 32 47 62 70 8  
23 31 46 61 69 3 11 26 41 49 64 68 4 10 25 33 48 63 71 5 13 28 43 51 66

本次洗牌扣分：0 分

2. 第 10 次洗牌，第 11 次切牌：

切分点为：15

当前扑克为：51 9 66 24 4 30 68 21 64 29 10 31 2 69 12 61 73 46 7 3 50 41 65 26 45 11  
53 49 22 70 28 8 43 23 27 62 42 72 48 6 33 67 25 1 63 44 13 52 5 32 71 47

本次洗牌扣分：0 分

3. 第 36 次洗牌，第 33 次切牌：

当前扑克为：49 67 62 31 24 53 33 43 2 71 51 13 61 68 45 65 47 66 46 26 12 11 44 23 29  
63 41 4 27 30 72 42 6 73 3 7 70 8 32 64 50 1 25 5 22 52 9 28 10 69 21 48

本次洗牌扣分：0 分

#### 源码全文：

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define M 52//所有扑克牌的种类
#define N 52//选出的扑克牌数量
int initial(int *poke);//初始化
int PrintPoke(int *poke);//打印当前扑克
int Devide(int *poke);//切牌,返回切分点
int Wash(int *poke);//洗牌
int group(int a);//以四种返回值代表四种花色
int Mark(int *poke);//打分函数，返回扣分总数

```

```

int AllPoke[M]={
    1,2,3,4,5,6,7,8,9,10,11,12,13,
    21,22,23,24,25,26,27,28,29,30,31,32,33,
    41,42,43,44,45,46,47,48,49,50,51,52,53,
    61,62,63,64,65,66,67,68,69,70,71,72,73
};
int main(){
    int poke[N];//建立扑克堆
    initial(poke);//初始化
    PrintPoke(poke);//打印当前扑克
    int D=0,W=0;//D、W 为切牌、洗牌次数
    char n;//操作
    while(1){
        printf("请选择操作： \n");
        printf("洗牌： 1\n");
        printf("切牌： 2\n");
        printf("退出： 0\n");
        scanf("%c",&n);
        fflush(stdin);
        if(n=='0') break;
        switch(n){
            case '1':
                printf("第%d 次洗牌， 第%d 次切牌： \n",++W,D);
                Wash(poke);//洗牌
                PrintPoke(poke);//打印当前扑克
                break;
            case '2':
                printf("第%d 次洗牌， 第%d 次切牌： \n",W,++D);
                printf("切分点为： %d\n",Devide(poke));
                PrintPoke(poke);//打印当前扑克
                break;
            default:
                break;
        }
    }
}

int initial(int *poke){//初始化
    for(int i=0;i<N;i++){//初始为顺序牌
        poke[i]=AllPoke[i];
    }
    return 0;
}

int PrintPoke(int *poke){//打印当前扑克
    printf("当前扑克为： ");

```

```

    for(int i=0;i<N;i++){
        printf("%d ",poke[i]);
    }
    printf("\n");
    printf("本次洗牌扣分: %d 分\n",Mark(poke));
    return 0;
}

int Devide(int *poke){//切牌,返回切分点
    srand(rand()+165+time(NULL));//随机化
    int n=rand()%(N-1)+1;//n 为切分点
    int a[N];//暂时存储的数组
    for(int i=0;i<N-n;i++){
        a[i]=poke[i+n];
    }
    for(int i=N-n;i<N;i++){
        a[i]=poke[i-N+n];
    }
    for(int i=0;i<N;i++){
        poke[i]=a[i];
    }
    return n;
}

int Wash(int *poke){//洗牌
    int a[N];//暂时存储的数组
    for(int i=0;i<(N+1)/2;i++){
        a[i]=poke[i*2];
    }
    for(int i=(N+1)/2;i<N;i++){
        a[i]=poke[(i-((N+1)/2))*2+1];
    }
    for(int i=0;i<N;i++){
        poke[i]=a[i];
    }
    return 0;
}

int group(int a){//以四种返回值代表四种花色
    if(a>=1&&a<=13) return 1;
    if(a>=21&&a<=33) return 2;
    if(a>=41&&a<=53) return 3;
    if(a>=61&&a<=73) return 4;
    return -1;
}

int Mark(int *poke){//打分函数, 返回扣分总数
    int m=0;//m 为扣分总数

```

```
    for(int i=0;i<N-2;i++){  
        if((group(poke[i])==group(poke[i+1]))&&(group(poke[i])==group(poke[i+2])))  
m++;  
    }  
    return m;  
}
```