# ICS lab6-C语言实现

## 整体源代码

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
    int value;
    struct Node *next;
};
typedef struct Node Node;
void print(int a,int b,int c){
    printf("ROW A:");
    for(int i=1;i<=a;i++){
        printf("o");
    }
    printf("\n");
    for(int i=1;i<=b;i++){
        printf("o");
    }
    printf("\n");
    for(int i=1;i<=c;i++){
        printf("o");
    }
    printf("\n");
}
int lab4(){
    int A=2,B=5,C=8;
    int turn=1;
    print(A,B,C);
    int choose;
    char c;
    while(A+B+C!=1){
        printf("The player %d should input:",turn);
        scanf("%c %d",&c,&choose);
        if(c=='A'){
            if(choose<=A){
                A-=choose;
                turn=(turn+3)%3+1;
            }
            else{
                printf("Error\n");
            }
        }
        else if(c=='B'){
            if(choose<=B){
                B-=choose;
                turn=(turn+3)%3+1;
            }
            else{
                printf("Error\n");
            }
        }
        else if(c=='C'){
```

```
            if(choose<=C){
                C-=choose;
                turn=(turn+3)%3+1;
            }
            else{
                printf("Error\n");
            }
        }
        else{
            printf("Error\n");
        }

    }
    printf("The player %d win\n",turn);
}
int lab2(int x,int y){
    if(x==0||y==0){
        return 0;
    }
    if(x==y){
        return x;
    }
    else if(x>y){
        return gcd(x-y,y);
    }
    else{
        return gcd(y,x);
    }
}
void lab3(Node *header){
    Node *p1;
    Node *p2;
    int temp;
    p1=header;
    p2=header->next;
    while(p1!=NULL){
        while(p2!=NULL){
            if(p1->value<p2->value){
                temp=p1->value;
                p1->value=p2->value;
                p2->value=temp;
            }
            p2=p2->next;
        }
        p1=p1->next;
        p2=p1->next;
    }
}
short cycle(short origin){
    short new_origin=origin<<1;
    if(origin<0){
        return new_origin+1;
    }
    else{
        return new_origin;
    }
}
int lab1(short origin,unsigned short times){
```

```
    for(int i=1;i<=times;i++){
        origin=cycle(origin);
    }
    return origin;
}
int lab5(){
    while(1){
        printf("ICS2020 ICS2020");
        char c=getchar();
        if(c>='0'&&c<='9'){
            printf("%c is a decimal digit");
        }
        else{
            printf("%c is not a decimal digit");
        }
    }
}
int main(){
    int x,y;
    scanf("%d%d",&x,&y);
    int result;
    result=lab1(x,y);
    printf("%d\n",result);
    result=lab2(x,y);
    printf("%d\n",result);
    Node *header;
    header=(Node *)malloc(sizeof(Node));
    Node *tempNode;
    tempNode=header;
    tempNode->next=(Node*)malloc(sizeof(Node));
    tempNode->next->next=(Node*)malloc(sizeof(Node));
    tempNode->next->next->next=NULL;
    tempNode->value=3;
    tempNode->next->value=5;
    tempNode->next->next->value=2;
    lab3(tempNode);
    while(tempNode){
        printf("tempNode is %d\n",tempNode->next);
    }
    lab4();
    lab5();
}
```

## 主控函数，调用各个函数模块

```
int lab5(){
    while(1){
        printf("ICS2020 ICS2020");
        char c=getchar();
        if(c>='0'&&c<='9'){
            printf("%c is a decimal digit");
        }
        else{
            printf("%c is not a decimal digit");
        }
    }
}
```

```
int main(){
    int x,y;
    scanf("%d%d",&x,&y);
    int result;
    result=lab1(x,y);
    printf("%d\n",result);
    result=lab2(x,y);
    printf("%d\n",result);
    Node *header;
    header=(Node *)malloc(sizeof(Node));
    Node *tempNode;
    //initial the linklist
    tempNode=header;
    tempNode->next=(Node*)malloc(sizeof(Node));
    tempNode->next->next=(Node*)malloc(sizeof(Node));
    tempNode->next->next->next=NULL;
    tempNode->value=3;
    tempNode->next->value=5;
    tempNode->next->next->value=2;
    //initial the link list
    lab3(tempNode);
    while(tempNode){
        printf("tempNode is %d\n",tempNode->next);
    }
    lab4();
    lab5();
}
```

# lab1

```
short cycle(short origin){
    short new_origin=origin<<1;
    if(origin<0){
        return new_origin+1;
    }
    else{
        return new_origin;
    }
}
int lab1(short origin,unsigned short times){
    for(int i=1;i<=times;i++){
        origin=cycle(origin);
    }
    return origin;
}
```

有两个函数，lab1调用cycle，cycle时对其参数进行一次循环位移的函数，循环位移之前，判断是否为负数，如果是负数，就在移位结果中加上1

# lab2

```
int lab2(int x,int y){
    if(x==0||y==0){
        return 0;
    }
```

```
    if(x==y){
        return x;
    }
    else if(x>y){
        return lab2(x-y,y);
    }
    else{
        return lab2(y,x);
    }
}
```

递归调用自身，利用最大公约数的性质

# lab3

```
void lab3(Node *header){
    Node *p1;
    Node *p2;
    int temp;
    p1=header;
    p2=header->next;
    while(p1!=NULL){
        while(p2!=NULL){
            if(p1->value<p2->value){
                temp=p1->value;
                p1->value=p2->value;
                p2->value=temp;
            }
            p2=p2->next;
        }
        p1=p1->next;
        if(p1!=NULL)
            p2=p1->next;
        else
            break;
    }
}
```

冒泡排序

# lab4

```
int lab4(){
    int A=2,B=5,C=8;
    int turn=1;
    print(A,B,C);
    int choose;
    char c;
    while(A+B+C!=1){
        printf("The player %d should input:",turn);
        scanf("%c %d",&c,&choose);
        if(c=='A'){
            if(choose<=A){
                A-=choose;
                turn=(turn+3)%3+1;
            }
```

```
            else{
                printf("Error\n");
            }
        }
        else if(c=='B'){
            if(choose<=B){
                B-=choose;
                turn=(turn+3)%3+1;
            }
            else{
                printf("Error\n");
            }
        }
        else if(c=='C'){
            if(choose<=C){
                C-=choose;
                turn=(turn+3)%3+1;
            }
            else{
                printf("Error\n");
            }
        }
        else{
            printf("Error\n");
        }

    }
    printf("The player %d win\n",turn);
}
```

A,B,C记录剩下的每一排的石子数目，choose记录当前的选择

# lab5

```
int lab5(){
    while(1){
        printf("ICS2020 ICS2020");
        char c=getchar();
        if(c>='0'&&c<='9'){
            printf("%c is a decimal digit");
        }
        else{
            printf("%c is not a decimal digit");
        }
    }
}
```

就是简单的if-else

# 输出结果

```
stdout=Microsoft-MIEngine-Out-04ianc1.ae3   --stderr=Microsoft-MIEngine-Error-ai5iqhsv.cii   --pid=Microsoft-MIEngine-Pid-ljcx4xou.mqk   --dbg
ngw64\bin\gdb.exe' '--interpreter=mi'
100 10
-28671
10
tempNode is 5
tempNode is 3
tempNode is 2
ROW A:oo
ROW B:ooooo
ROW C:oooooooo
The player 1 should input:A 2
ROW A:
ROW B:ooooo
ROW C:oooooooo
The player 3 should input:B 3
ROW A:
ROW B:oo
ROW C:oooooooo
The player 2 should input:B 1
ROW A:
ROW B:o
ROW C:oooooooo
The player 1 should input:C 8
ROW A:
ROW B:o
ROW C:
The player 3 win
ICS2020 ICS20202
 is a decimal digitICS2020 ICS2020 is not a decimal digitICS2020 ICS2020a
 is not a decimal digitICS2020 ICS2020 is not a decimal digitICS2020 ICS2020
```