实验报告

Lab06 C Programming

PB19030861 王湘峰

实验要求

1.将 lab02 至 lab05 用 C 语言实现

2.思考并写出低级语言和高级语言的区别

实验设计

对于实验 2，采用辗转相处法求最小公约数，其中利用 C 语言的“%”进行带余除法。

对于实验 3，采用冒泡排序对链表进行排序，数据结构上使用结点的结构体实现。

对于实验 4，利用 C 语言的标准输入输出函数进行读取操作。

对于实验 5，主要运用 C 语言 windows.h 中的 kbhit（）函数进行中断检测。

代码部分

头函数：

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<conio.h>
4  #include<windows.h>
```

结构体：

```
typedef struct node{
    long value;
    node *next;
}node;
```

子函数:

Lab02:

```
void lab02(){
    long a,b,r;
    long A,B;
    long cd;
    printf("please input two integers\n");
    printf("first number:");
    scanf("%ld",&A);
    printf("second number:");
    scanf("%ld",&B);

    a=A;b=B;
    if(a>=b){
        r=a-b*(a/b);
        while(r){
            a=b;b=r;
            r=a-b*(a/b);
        }
        cd=b;
    }
    else {
        r=b-a*(b/a);
        while(r){
            b=a;a=r;
            r=b-a*(b/a);
        }
        cd=a;
    }
    printf("The greatest common divisor of %ld and %d is %ld\n",A,B,cd);
}
```

Lab03:

```
    int i;
    int n=0;
    long value;
    do{
    printf("input how many nodes you want to create the linklist\n");
    scanf("%d",&n);
    if(n<=0)
    printf("invalid input!\n");
    }while(n<=0);
    head=(node*)malloc(sizeof(node));
    if(!head) exit(-1);
    head->next=NULL;
    printf("input value\n");
    for(i=0;i<n;i++){
        p=(node*)malloc(sizeof(node));
        if(!p) exit(-1);
        scanf("%ld",&value);
        p->next=head->next;
        head->next=p;
        p->value=value;
    }
    /*printf("The linklist now is:\n");
    for(p=head->next;p;p=p->next)
    printf("%ld\n",p->value);*/
    node* New;
    node* pmax;
    node* q;
    New=(node*)malloc(sizeof(node));
    New->next=NULL;
    long max;
    for(i=0;i<n;i++){
        max=-65534;
        for(p=head->next;p;p=p->next){
            if(p->value>max){
                max=p->value;
                pmax=p;
            }
        }
        pmax->value=-65535;
        q=(node*)malloc(sizeof(node));
        q->value=max;
        q->next=New->next;
        New->next=q;
    }
    printf("The linklist after sorted is:\n");
    for(q=New->next;q;q=q->next)
    printf("%ld\n",q->value);

}
```

Print:

```c
void Print(int a,int b, int c){
    int k;
    printf("ROWA:");
    for(k=0;k<a;k++)
    putchar('o');
    printf("\n");
    printf("ROWB:");
    for(k=0;k<b;k++)
    putchar('o');
    printf("\n");
    printf("ROWC:");
    for(k=0;k<c;k++)
    putchar('o');
    printf("\n");
}
```

Lab 04:

```
void lab04(){
    int i=1;
    int k;
    int a,b,c;
    char in[100];
    char row;
    int num;
    a=3;b=5;c=8;

    do{
        if(a==0&&b==0&c==0){
            if(i>0)printf("Player1 Wins.\n");
            else printf("Player2 Wins.\n");
            //printf("Press Enter to Continue\n");
            break;
        }
    Print(a,b,c);
    if(i>0)
    printf("Player1,choose a row and number of rocks:");
    else
    printf("Player2,choose a row and number of rocks:");
    gets(in);
    row=in[0];
    num=in[1]-'0';
    if(row=='A'){
        if(a<num||num<=0){
            printf("invalid move ,try again\n");
            continue;
        }

        else {
            a=a-num;
            i=-i;
            continue;
        }
    }
    else if(row=='B'){
        if(b<num||num<=0){
            printf("invalid move ,try again\n");
            continue;
        }
        else {
            b=b-num;
            i=-i;
            continue;
        }
    }
    else if(row=='C'){
        if(c<num||num<=0){
        printf("invalid move ,try again\n");
        continue;
     }
     else {
     c=c-num;
     i=-i;
     continue;
    }
   }
    else {
    printf("invalid move ,try again\n");
    continue;
    }
 }while(1);
}
```

Lab05:

```
172 void lab05(){
173     int i;
174     int key;
175     printf("\nPress ESC to Exit\n\n");
176     while(1){
177     printf(" ICS 2020 ");
178     Sleep(200);
179     if(kbhit()){
180         printf("\n");
181         key=getch();
182         if(key==27) break;
183         putchar(key);
184         key=key-'0';
185         if(key>9||key<0)
186         printf(" is not a demical digit\n");
187         else printf(" is a demical digit\n");
188     }
189  }
190 }
```

结果测试

```
C:\Users\wonderful\Desktop\学习&工作\ICS实验\PB19030861_王湘峰_LAB06\program.exe
which function to run?
2 for lab02
3 for lab03
4 for lab04
5 for lab05
please input two integers
first number:180
second number:450
The greatest common divisor of 180 and 450 is 90
which function to run?
```

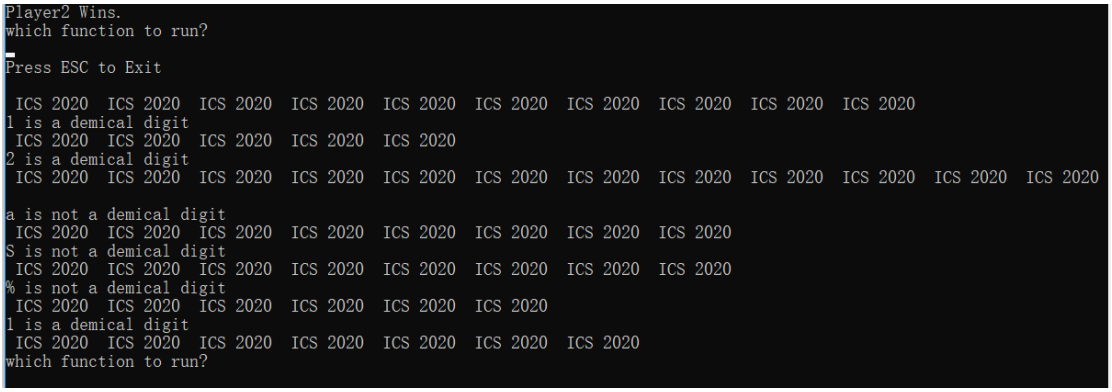
（test of lab02）

```
which function to run?
input how many nodes you want to create the linklist
4
input value
123
654
963
741
The linklist after sorted is:
123
654
741
963
which function to run?
```

（test of lab03）

```
C:\Users\wonderful\Desktop\学习&工作\ICS实验\PB19030861_
which function to run?
2 for lab02
3 for lab03
4 for lab04
5 for lab05
ROWA:ooo
ROWB:ooooo
ROWC:oooooooo
Player1,choose a row and number of rocks:A3
ROWA:
ROWB:ooooo
ROWC:oooooooo
Player2,choose a row and number of rocks:A1
invalid move ,try again
ROWA:
ROWB:ooooo
ROWC:oooooooo
Player2,choose a row and number of rocks:B5
ROWA:
ROWB:
ROWC:oooooooo
Player1,choose a row and number of rocks:C8
Player2 Wins.
which function to run?
```

（test of lab04）



（test of lab05）

思考

高级语言例如 C、C++、Python、Java 等和低级语言相比，使用更多的指令（函数）实现，更利于程序的理解和记忆。在程序的行文上更接近人类的自然语言的习惯，因此在**编程时**更高效。最重要的是，高级语言易于移植，代码的复用性强。

低级语言如汇编语言、机器语言的优点是在**运行时**更高效。通过对汇编语言的直接编写，可大大简短机器执行指令的时间。而通过汇编器汇编的低级语言往往不够简洁，代码执行重复低效（相比于人类直接汇编）。但其缺点是程序较难思考、错误较为晦涩以及难以记忆，移植性也较弱。

综合来看，二者更有千秋。在实际问题的解决过程中，可根据具体情况综合使用二者。例如，对于汇编语言较为复杂的程序可以用高级语言的函数调用实现；而对那些对运算速度有较高要求的程序，则可人工编写汇编程序，大大缩短程序执行的速度。