计算概论（C语言）习题课讲义11

内容概要

- 机考题目讲解
- 文件操作
- 习题讲解

机考题目讲解

矩形面积

```c
// 想法：逐个情况判断
#include <stdio.h>

int recArea(int ldx, int ldy, int rux, int ruy)
{
    return (rux-ldx)*(ruy-ldy);
}

int area(int A, int B, int C, int D, int E, int F, int G, int H)
{
    int AREA=recArea(A,B,C,D)+recArea(E,F,G,H);
    if(A<=E)
    {
        if(C<=E)
            return AREA;
        if(B>=F)
        {
            if(B>=H)
                return AREA;
            if(D<H)
                return AREA-recArea(E,B,C,D);
            return AREA-recArea(E,B,C,H);
        }
        if(D<F)
            return AREA;
        if(D<H)
            return AREA-recArea(E,F,C,D);
        return AREA-recArea(E,F,C,H);
    }
    return area(E,F,G,H,A,B,C,D);
}

int main()
{
    int A,B,C,D,E,F,G,H;
    scanf("%d %d %d %d %d %d %d %d", &A,&B,&C,&D,&E,&F,&G,&H);
```

```
    printf("%d", area(A,B,C,D,E,F,G,H));
    return 0;
}
```

```
// 思路:容斥原理;计算重叠区域面积可以分解为两个一维问题
#include <stdio.h>
#include <math.h>
int main()
{
    int A,B,C,D,E,F,G,H;
    scanf("%d %d %d %d %d %d %d %d",&A,&B,&C,&D,&E,&F,&G,&H);
    int s1,s2,s3,h,l;
    s1 = (C - A) * (D - B);
    s2 = (G - E) * (H - F);
    h = min(D, H) - max(B, F);
    l = min(C, G) - max(A, E);
    if((h > 0)&(l>0))
        s3 = h * l;
    else
        s3 = 0;

    printf("%d", s1 + s2 - s3);
    return 0;
}

int max(int a, int b)
{
    if(a>b)
        return a;
    return b;
}

int min(int a, int b)
{
    if(a<b)
        return a;
    return b;
}
```

和最大子序列

```
// 思路: 暴力枚举
#include<stdio.h>
#include<stdlib.h>
int add(int *A,int i,int j){
    int s=0;
    for(;i<=j;i++){
        s+=A[i];
    }
    return s;
}
int maxsum(int *A, int len){
```

```
    int i,j,sum=0,maxsum=-1000000;
    for(i=0;i<len;i++){
        for(int j=i;j<len;j++){
            sum=add(A,i,j);
            if(sum>maxsum)maxsum=sum;
        }
    }
    return maxsum;
}
int main(){
    int n,*A,k;
    scanf("%d",&n);
    A=(int *)malloc(n*sizeof(int));
    for(k=0;k<n;k++)scanf("%d",&A[k]);
    printf("%d",maxsum(A,n));
    free(A);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
const int INF = -2100000000;
int maxsum(int * A, int n)
{
    int thissum = 0, maxm = INF;
    for(int i = 0; i < n; ++i)
    {
        thissum += A[i];
        if(thissum > maxm)
        {
            maxm = thissum;
        }
        if(thissum < 0) //有技巧地断开子序列
        {
            thissum = 0;
        }
    }
    return maxm;
}
```

最小操作数

```
// 思路:递归
#include <stdio.h>
#include <stdlib.h>

int Calc(int x,int y)
{
    if(x>=y)
    {
        return x-y;
    }
```

```
    else if(y%2==0)
    {
        return Calc(x,y/2)+1;
    }
    else if(y%2==1)
    {
        return Calc(x,y+1)+1;
    }
}

int main()
{
    int x,y;
    scanf("%d%d",&x,&y);
    printf("%d",Calc(x,y));
    return 0;
}
```

```
// 思路：做标记(广度优先搜索)
#include <stdio.h>

int flag[2000];

int Calc(int x, int y)
{
        if(x>=y)
                return x-y;
        //x<y
        int N=1+y;
        for(int i=0;i<N;i++)
        {
                flag[i]=-1;
        }
        flag[x-1]=0;
        int n=0;
        while(flag[y-1]==-1)
        {
                for(int i=0;i<N;i++)
                {
                        if(flag[i]==n)
                        {
                                if(i-1>=0&&flag[i-1]==-1)
                                        flag[i-1]=n+1;
                                if(2*i+1<N&flag[2*i+1]==-1)
                                        flag[2*i+1]=n+1;
                        }
                }
                n=n+1;
        }
        return flag[y-1];
}
```

二进制补码

```
//思路：模拟补码计算过程
#include <stdio.h>

int bit[32]={0};

void to_complement(int n, int bits)
{
        int tmp=n;
        if(n<0)
        {
                tmp=-1*tmp;
        }
        // Convert to binary
        for(int i=0;i<bits-1;i++)
        {
                if(tmp%2==1)
                {
                        bit[i]=1;
                }
                tmp=tmp/2;
        }
        if(tmp!=0)
        {
                printf("%d is out of the range of representation with %d
bits!\n",n,bits);
                return;
        }
        if(n<0)
        {
                // convert
                for(int i=0;i<bits;i++)
                {
                        if(bit[i]==0)
                                bit[i]=1;
                        else
                                bit[i]=0;
                }
                // +1
                for(int i=0;i<bits;i++)
                {
                        if(bit[i]==1)
                                bit[i]=0;
                        else
                        {
                                bit[i]=1;
                                break;
                        }
                }
        }
        // Output
        for(int i=bits-1;i>=0;i--)
```

```
                printf("%d",bit[i]);
        printf("\n");
}

int main()
{
        int n, bits;
        scanf("%d %d",&n,&bits);
        to_complement(n,bits);
        return 0;

}
```

字符串比较

```
// 思路：递归
#include <stdio.h>
#include <string.h>

int isMatch(const char *p, const char *s)
{
    int n, m;
    n=strlen(p);
    m=strlen(s);
    if(n==1)
    {
        if(*p=='*')
            return 1;
        if(*p=='?')
            return m==1?1:0;
        return m==1&&*p==*s;
    }
    if(*p=='*')
    {
        int i;
        for(i=0;i<m;i++)
        {
            if(isMatch(p+1, s+i))
                return 1;
        }
        return 0;
    }
    if(*p=='?')
        return isMatch(p+1, s+1);

    return *p==*s&&isMatch(p+1, s+1);
}
```

文件读写

C语言中的文件读写，本质上和对标准输入输出的读和写相同. 文件读写最显著的特征是在于需要 使用文件指针，来具体指明读写的文件.

```
// 打开文件
FILE *fopen( const char * filename, const char * mode );
// 关闭文件
int fclose( FILE *fp );
```

其中，文件操作模式有`r`,`w`,`a`,`+`, 即读/写/附加/更新.

文件的读写的常用函数为

1. `fscanf`
2. `fprintf`

用法和 `scanf` , `printf` 类似. 其他的文件读写函数, 例如 `fgetc`, `fgets`, 都可以在一定程度上都可以看作是`fscanf`函数的特例.

> 注意: `fgets` 和 `fscanf(*,"%s",*)` 的区别

其他常用的文件操作函数

```
int feof(FILE *st); //判断是否到达文件末尾
int fseek(FILE *stream, long offset, int whence); //移动文件指针的位置, 但参数有些复杂, 不常用
void rewind(FILE * st); // 将文件指针移动到开头
```

习题讲解

电费统计

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

double ssum=0.0;

void crg(FILE *fp){
    char name[256];
    double sum=0.0,deg,upr;
    while(fgets(name,256,fp)!=NULL)
    {
        sum=0;
        while(fscanf(fp,"%lf%lf",&deg,&upr)==2) //不断统计
            sum+=deg*upr;
        printf("%s----------------------%.2f\n",name,sum);//输出
        ssum+=sum;
    }
}

int main(int argc,char *argv[]){
    char name[256];
    FILE *fp;
    while(1){
        printf("File name (end-of-file to end):\n");
        if(scanf("%255s",name)==EOF) //处理结束
```

```
        {
            printf("The total charge is %.2f.\n",ssum);
            break;
        }
        if((fp=fopen(name,"r"))==NULL) //打开失败
            fprintf(stderr,"Can't open file:\n%s\n",name);
        else //打开成功
        {
            crg(fp);
            fclose(fp);
        }
    }
    printf("Bye!\n");
    return 0;
}
```

文件单词在单词表中行号的统计

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void process(FILE * fpr2, FILE * fpw, char * word){
    int i;
    char line[200]={0};
    fprintf(fpw, "%s ", word);
    rewind(fpr2); // 必要的操作, 为什么?
    for (i = 1; 1; i++)
    {
        if (fgets(line,200,fpr2)==NULL)
        {
            break;
        }
        if (strstr(line,word)!=NULL)
        {
            fprintf(fpw,"%d ", i);
        }
    }
    fprintf(fpw,"\n");
    return;
}

int main(int argc, char const *argv[])
{
    FILE *fpr1, *fpr2, *fpw;
    char word[20]={0};
    if ((fpr1=fopen(argv[1],"r"))==NULL || (fpr2=fopen(argv[2],"r"))==NULL)
    {
        printf("Error!\n");
        return 0;
    }
    fpw=fopen("8write","w");
    while ((fscanf(fpr1,"%s",word))==1) //不断读入单词
```

```
    {
        process(fpr2,fpw,word); //进行处理
    }
    fclose(fpr1);
    fclose(fpr2);
    fclose(fpw);
    return 0;
}
```

简单复习

第六章

1. 数组定义/声明/初始化的区别和联系.
2. 二维数组初始化为`int a[2][3]={0,1,2,3,4,5};`, 那么 `a[0][2]` 是? `a[0][3]` 是? 这说明了什么?
3. `int a[5]={0,1,2,3,4}` 所以 `sizeof(a)` 多大呢? 和 `sizeof(int)` 有关么?

第七章

1. 这样的操作可以么?

```
int* a;
*a = 3;
```

2.`b[3]`的操作合法么? 值为多少?

```
int a[4]={0,1,2,3};
int *b=a;
```

3.`sizeof(p1)`和`sizeof(p2)`,`sizeof(p3)`的关系是?

```
int *p1;
double *p2;
char *p3;
```

4.`color[0][3]`的值为? `color[1][6]`的值? 定义中的`6` 可以去掉么?

```
char color[][6]={"Red","GREEN","BLUE"};
```

5.解释`f`,`g`,`h`的含义: `int (*f)[8]`, `int (*g)(int)`, `char *h[]`;