

## 第二次书面作业解析

---

### 1. 计算多边形面积

我们可以任意选择一个点，比如原点，或者  $(x[1], y[1])$  来按照逆时针顺序，对相邻的点求一次外积，累加起来除以 2。

这里以原点为例子：

```
double area(int n, double x[], double y[]){
    double ans=0;
    for(int i=1; i<n; i++){
        ans+=(x[i]*y[i+1]-y[i]*x[i+1])/2;
    }
    ans+=(x[n]*y[0]-x[0]*y[n])/2;
    return ans;
}
```

### 2. 字符串翻转

第一小问比较简单。我们从  $a$  到  $(l+r)/2$ ，依次调换对应位置。

```
void reverse(char s[], int a, int b){
    int mid=(a+b)/2;
    for(int i=a; i<=mid; i++){
        char tmp=s[i];
        s[i]=s[b-i];
        s[b-i]=tmp;
    }
}
```

第二小题，我们可以先做一次处理：每次遇到一个负数（或者最高位为1）都和下一位交换，最后反转整个字符串。

```

void reverse_GBK(char s[],int len){
    for(int i=0;i<len;i++){
        if(s[i]&(128)){//最高位为1
            char tmp=s[i];
            s[i]=s[i+1];
            s[i+1]=tmp;
            i++;//这两位连着处理了，所以i要多加1
        }
    }
    reverse(s,0,len-1);
}

```

### 3. 前缀和

第一问求前缀和非常简单。只要循环求前*i*项累加的值就可以了。

```

void init_prefix_sum(){
    int sum=0;
    for(int i=1;i<=n;i++){
        sum+=a[i];
        s[i]=sum;
    }
}

```

注意，这里 `s[0]=0`，非常自然：前0项和为0。

然后就是求 `a[l]` 到 `a[r]` 的加和：

```

printf("a[%d] 到 a[%d]的和为%d\n",l,r,s[r]-s[l-1]);

```

这里注意是减去 `s[l-1]`。

### 4. 差分序列

按照定义我们求出差分序列，存放到数组 `s` 中。

```
void init_diff_seq(){
    s[1]=a[1];
    for(int i=2;i<=n;i++)
        s[i]=a[i]-a[i-1];
}
```

对于区间加操作，按照图示，其实就是希望我们将  $a[l]$  到  $a[r]$  整体「抬高」一段。

因此，抬高以后， $a[l]$  比  $a[l-1]$  多增加了 1， $a[r+1]$  比  $a[r]$  降低了 1。

所以我们给出区间加的代码。

```
void interval_add1(int l,int r){
    s[l]+=1;
    s[r+1]-=1;//注意这里的r+1是可能越界的，真正实现的时候要扩大数组或者作判定
}
```

最后，我们对  $s$  数组作前缀和，还原出修改后的  $a$  数组。

```
a[1]=s[1];
for(int i=2;i<=n;i++)
    a[i]=a[i-1]+s[i];
```