

# 练习

## Exercise



### 完善程序

要求：

- 1.在演草纸上填写答案
- 2.待所有空填写完毕后，打开程序，将草稿纸上的答案填入程序的空白处
- 3.运行程序，记录所有出现的错误
- 4.调试程序，尝试改正错误的答案
- 5.完成整个程序，可以顺利得到期望的结果
- 6.记录错误和心得经验

#### 1. 最大连续子段和（网站第 639 题）

给出一个数列（元素个数不多于 100），数列元素均为负整数、正整数、0。请找出数列中的一个连续子数列，使得这个子数列中包含的所有元素之和最大，在和最大的前提下还要求该子数列包含的元素个数最多，并输出这个最大和以及该连续子数列中元素的个数。例如数列为 4, -5, 3, 2, 4 时，输出 9 和 3；数列为 1 2 3 -5 0 7 8 时，输出 16 和 7。

```
#include<iostream>
using namespace std;
int a[101];
int n,i,ans,len,tmp,beg;
int main(){
    cin>>n;
    for(i=1;i<=n;i++) cin>>a[i];
    tmp=0;
    ans=0;
    len=0;
    beg=____(1)____;
    for(i=1;i<=n;i++){
        if(tmp+a[i]>ans){
            ans=tmp+a[i];
            len=i-beg;
        }
        else if(____(2)____ && i-beg>len)
            len=i-beg;
        if(tmp+a[i]____(3)____) {
            beg=____(4)____;
            tmp=0;
        } else
            ____ (5) ____;
    }
    cout<<ans<<" "<<len<<endl;
```

# 练习

## Exercise



```
    return 0;  
}
```

找出关键变量，解释变量作用

理清程序控制结构和运行顺序

经验心得

## 练习

### Exercise



#### 2. 最大子矩阵和 (网站第 640 题)

给出  $m$  行  $n$  列的整数矩阵, 求最大的子矩阵和 (子矩阵不能为空)。

输入第一行包含两个整数  $m$  和  $n$ , 即矩阵的行数和列数。之后  $m$  行, 每行  $n$  个整数, 描述整个矩阵。程序最终输出最大的子矩阵和。

```
#include<iostream>
using namespace std;
const int SIZE=100;
int matrix[SIZE+1][SIZE+1];
int rowsum[SIZE+1][SIZE+1];
//rowsum[i][j]记录前 i 行前 j 个数的和
int m,n,i,j,first,last,area,ans;
int main(){
    cin>>m>>n;
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
            cin>>matrix[i][j];
    ans=matrix____(1)____;
    for(i=1;i<=m;i++)
        ____ (2) ____;
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
            rowsum[i][j]=____(3)____;
    for(first=1;first<=n;first++){
        for(last=first;last<=n;last++){
            ____ (4) ____;
            for(i=1;i<=m;i++) {
                area+=____(5)____;
                if(area>ans) ans=area;
                if(area<0) area=0;
            }
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

# 练习

## Exercise



找出关键变量，解释变量作用

理清程序控制结构和运行顺序

经验心得

## 练习

### Exercise



#### 3. 国王放置 (网站第 641 题)

在  $n*m$  的棋盘上放置  $k$  个国王, 要求  $k$  个国王互相不攻击, 有多少种不同的放置方法。假设国王放在第  $(x,y)$  格, 国王的攻击的区域是:  $(x-1,y-1)$ ,  $(x-1,y)$ ,  $(x-1,y+1)$ ,  $(x,y-1)$ ,  $(x,y+1)$ ,  $(x+1,y-1)$ ,  $(x+1,y)$ ,  $(x+1,y+1)$ 。读入三个数  $n, m, k$ , 输出答案。题目利用回溯法求解。棋盘行标号为  $0 \sim n-1$ , 列标号为  $0 \sim m-1$ 。

```
#include <iostream>
using namespace std;
int n,m,k,ans;
int hash[5][5];
void work(int x,int y,int tot){
    int i,j;
    if (tot==k){
        ans++;
        return;
    }
    do{
        while (hash[x][y]){
            y++;
            if (y==m){
                x++;
                y=____(1)____;
            }
            if (x==n)
                return;
        }
        for (i=x-1;i<=x+1;i++)
            if (i>=0&&i<n)
                for (j=y-1;j<=y+1;j++)
                    if (j>=0&&j<m)
                        ____ (2) ____;
        ____ (3) ____;
        for (i=x-1;i<=x+1;i++)
            if (i>=0&&i<n)
                for (j=y-1;j<=y+1;j++)
                    if (j>=0&&j<m)
                        ____ (4) ____;
        y++;
        if (y==m){
            x++;
            y=0;
        }
        if (x==n)
            return;
    }
```

## 练习

### Exercise



```
    }  
    while (1);  
}  
int main(){  
    cin >> n >> m >> k;  
    ans=0;  
    memset(hash,0,sizeof(hash));  
    ____ (5) ____;  
    cout << ans << endl;  
    return 0;  
}
```

找出关键变量，解释变量作用

理清程序控制结构和运行顺序

经验心得