

# 作业二

## HOMEWORK 2



### 作业如何提交

#### 1. 访问作业网站:

<http://120.132.20.20:8080/thrall-web/main#home>

#### 2. 登陆名为学生中文名全拼, 初始密码 123456

#### 3. 第一次登陆后, 请马上修改密码, 把密码记住, 防止被盗号

#### 4. 选择作业的题目, 点击进入题目

#### 5. 进入题目内容后点右上角提交, 复制程序代码, 选择 c++语言, 提交作业

#### 6. 得分 100 分为满分。如果看到 AC, 代表 accepted 表示正确, 否则是错误, 可再次提交

完善程序专题, 在空缺处补全代码, 实现题目要求的功能。

### 第一题 读入整数 (网站第 237 题)

请完善下面的程序, 使得程序能够读入两个 int 范围内的整数, 并将这两个整数分别输出, 每行一个。

输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

输入样例:

123 -789

输出样例

123

-789

### 程序

```
#include<iostream>
```

```
using namespace std;
```

```
int readint() {  
    int num = 0;           // 存储读取到的整数  
    int negative = 0;      // 负数标志  
    char c;                // 存储当前读取到的字符  
    c = cin.get();  
    while ((c < '0' || c > '9') && c != '-')  
        c = ____ (1) ____;  
    if (c == '-')  
        negative = 1;  
    else  
        ____ (2) ____;  
    c = cin.get();  
    while (____ (3) ____) {  
        ____ (4) ____;  
    }  
}
```

## 作业二

### HOMEWORK 2



```
        c = cin.get();
    }
    if (negative == 1)
        ____ (5) ____;
    return num;
}

int main() {
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl << b << endl;
    return 0;
}
```

## 作业二

### HOMEWORK 2



#### 第二题 郊游活动（网站第 239 题）

有  $n$  名同学参加学校组织的郊游活动，已知学校给这  $n$  名同学的郊游总经费为  $A$  元，与此同时第  $i$  位同学自己携带了  $M_i$  元。为了方便郊游，活动地点提供  $B (\geq n)$  辆自行车供人租用，租用第  $j$  辆自行车的价格为  $C_j$  元，每位同学可以使用自己携带的钱或者学校的郊游经费，为了方便账务管理，每位同学只能为自己租用自行车，且不会借钱给他人，他们想知道最多有多少位同学能够租用到自行车。

本题采用二分法。对于区间  $[l, r]$ ，我们取中间点  $mid$  并判断租用到自行车的人数能否达到  $mid$ 。判断的过程是利用贪心算法实现的。

#### 程序

```
#include <iostream>
using namespace std;
#define MAXN 1000000

int n, B, A, M[MAXN], C[MAXN], l, r, ans, mid;

bool check(int nn) {
    int count = 0, i, j;
    i = ____ (1) ____;
    j = 1;
    while (i <= n) {
        if (____ (2) ____ )
            count += C[j] - M[i];
        i++;
        j++;
    }
    return ____ (3) ____;
}

void sort(int a[], int l, int r) {
    int i=l, j=r, x=a[(l+r)/2], y;
    while (i<=j) {
        while (a[i] < x) i++;
        while (a[j] > x) j--;
        if (i <= j) {
            y = a[i]; a[i] = a[j]; a[j] = y;
            i++; j--;
        }
    }
    if (i < r) sort(a, i, r);
}
```

## 作业二

### HOMEWORK 2



```
        if (l < j) sort(a, l, j);
    }

int main() {
    int i;
    cin >> n >> B >> A;
    for (i=1; i<=n; i++)
        cin >> M[i];
    for (i=1; i<=B; i++)
        cin >> C[i];
    sort(M, 1, n);
    sort(C, 1, B);
    l = 0;
    r = n;
    while (l <= r) {
        mid = (l + r) / 2;
        if (____(4)____) {
            ans = mid;
            l = mid + 1;
        } else
            r = ____ (5) ____;
    }
    cout << ans << endl;
    return 0;
}
```

## 作业二

### HOMEWORK 2



#### 第三题 最大子矩阵和（网站第 238 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。程序最终输出的最大的子矩阵和。

输入样例：

2 2

1 2

1 -1

输出样例：

3

程序

```
#include<iostream>
```

```
using namespace std;
```

```
const int SIZE = 100;
```

```
int matrix[SIZE+1][SIZE+1];
```

```
int rowsum[SIZE+1][SIZE+1]; //rowsum[i][j]记录第 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) ____;
```

```
            rowsum[i][j] = rowsum[i][j-1] + matrix[i][j];
```

```
    for (first=1; first <= n; first++)
```

```
        for (last=first; last <= n; last++) {
```

```
            ____ (4) ____;
```

```
            for (i=1; i<=m; i++) {
```

```
                area += ____ (5) ____;
```

```
                area += ;
```

```
                if (area > ans)
```

```
                    ans = area;
```

## 作业二

### HOMEWORK 2



---

```
        if (area < 0)
            area = 0;
        }
    }
    cout << ans << endl;
    return 0;
}
```