

CS150

初赛专题 集训



CS150 初赛专题集训公布资料的固定网站
请每次课前自行将资料下载到电脑

<https://pan.baidu.com/s/12ZsJgSE-p17Vxl-ObMe7Mg>

快快编程地址

<http://120.132.18.213:9062>

请登陆网站提交作业

递归

200901

```
#include <iostream>
using namespace std;
int a,b;
int work(int a,int b){
    if (a%b)
        return work(b,a%b);
    return b;
}
int main(){
    cin >> a >> b;
    cout << work(a,b) << endl;
    return 0;
}
```

输入: 20 12

输出:

如何理解递归程序

1

分析函数参数和返回值
找出递归的边界条件

```
int work(int a, int b) {  
    if (a%b)  
        return work(b, a%b);  
    return b;  
}
```

2个参数

当 $a\%b$ 余数大于0时,
递归调用

当 $a\%b$ 余数等于0时,
到达递归边界, 返回b

如何理解递归程序

2

用树状图表示递归过程

参数	数值
a	20
b	12
返回值return	4

参数	数值
a	12
b	8
返回值return	4

参数	数值
a	8
b	4
返回值return	4

——→ 递归调用顺序
- - - - -> 返回值顺序

201402

```
#include<iostream>
using namespace std;
int fun(int n) {
    if (n == 1)
        return 1;
    if (n == 2)
        return 2;
    return fun(n - 2) - fun(n - 1);
}
int main() {
    int n;
    cin >> n;
    cout << fun(n) << endl;
    return 0;
}
```

输入： 7

输出：

201402

1

分析函数参数和返回值
找出递归的边界条件

```
int fun(int n) {  
    if (n == 1)  
        return 1;  
    if (n == 2)  
        return 2;  
    return fun(n - 2) - fun(n - 1);  
}
```

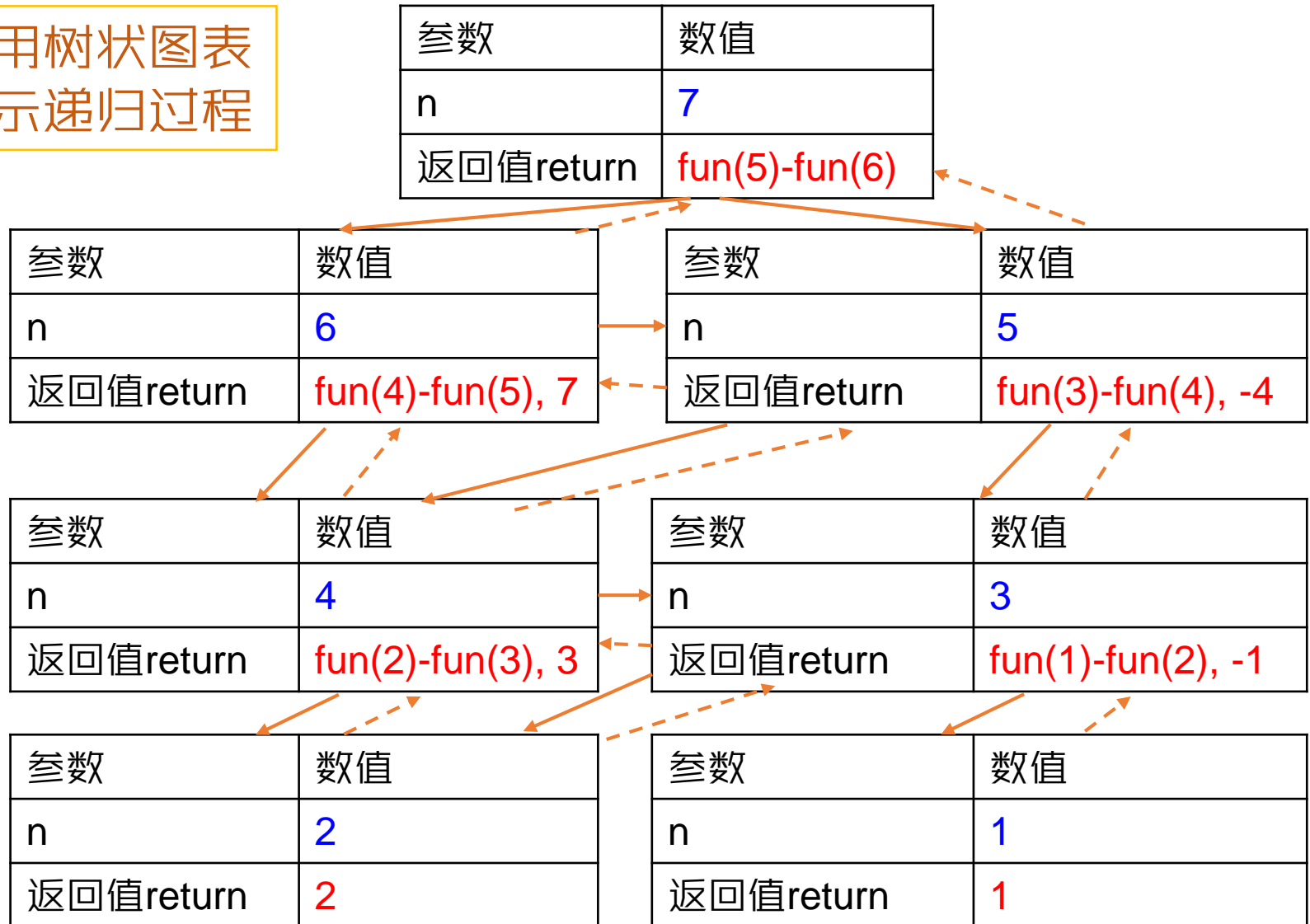
1个参数

边界条件: $n==1$ 或 $n==2$

201402

2

用树状图表示递归过程



201702

```
#include<iostream>
using namespace std;
int g(int m, int n, int x) {
    int ans = 0;
    int i;
    if(n == 1)
        return 1;
    for(i = x; i <= m / n; i++)
        ans += g(m - i, n - 1, i);
    return ans;
}
int main() {
    int t, m, n;
    cin>> m >> n;
    cout<< g(m, n, 0) << endl;
    return 0;
}
```

输入：7 3

输出：

201702

1

分析函数参数和返回值
找出递归的边界条件

```
int g(int m, int n, int x) {  
    int ans = 0;  
    int i;  
    if (n == 1)  
        return 1;  
    for (i = x; i <= m / n; i++)  
        ans += g(m - i, n - 1, i);  
    return ans;  
}
```

3个参数

边界条件: $n==1$

201702

2

用树状图表示递归过程

参数	数值
m	7
n	3
x	0
返回值	8

参数	数值
m	7
n	2
x	0
返回值	4

参数	数值
m	6
n	2
x	1
返回值	3

参数	数值
m	5
n	2
x	2
返回值	1

2

用树状图表示递归过程

参数	数值
m	7
n	2
x	0
返回值	4

参数	数值
m	4
n	1
x	3
返回值	1

参数	数值
m	7
n	1
x	0
返回值	1

参数	数值
m	6
n	1
x	1
返回值	1

参数	数值
m	5
n	1
x	2
返回值	1

2

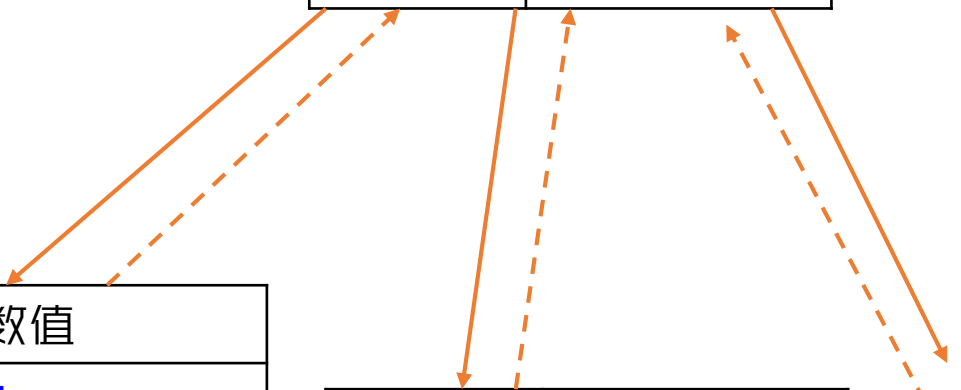
用树状图表示递归过程

参数	数值
m	6
n	2
x	1
返回值	3

参数	数值
m	5
n	1
x	1
返回值	1

参数	数值
m	4
n	1
x	2
返回值	1

参数	数值
m	3
n	1
x	3
返回值	1

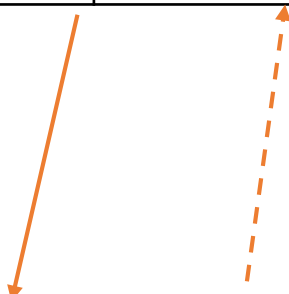


2

用树状图表示递归过程

参数	数值
m	5
n	2
x	2
返回值	1

参数	数值
m	3
n	1
x	2
返回值	1



201104

```
#include <iostream>
using namespace std;
int solve(int n, int m) {
    int i, sum;
    if (m == 1)
        return 1;
    sum = 0;
    for (i = 1; i < n; i++)
        sum += solve(i, m - 1);
    return sum;
}
int main() {
    int n, m;
    cin>>n>>m;
    cout<<solve(n, m)<<endl;
    return 0;
}
```

输入： 7 4

输出：

201104

1

分析函数参数和返回值
找出递归的边界条件

```
int solve(int n, int m) {  
    int i, sum;  
    if (m == 1)  
        return 1;  
    sum = 0;  
    for (i = 1; i < n; i++)  
        sum += solve(i, m - 1);  
    return sum;  
}
```

2个参数

边界条件: $m==1$

2

用树状图表示递归过程

参数	数值
n	7
m	4
返回值	20

参数	数值
n	6
m	3
返回值	10

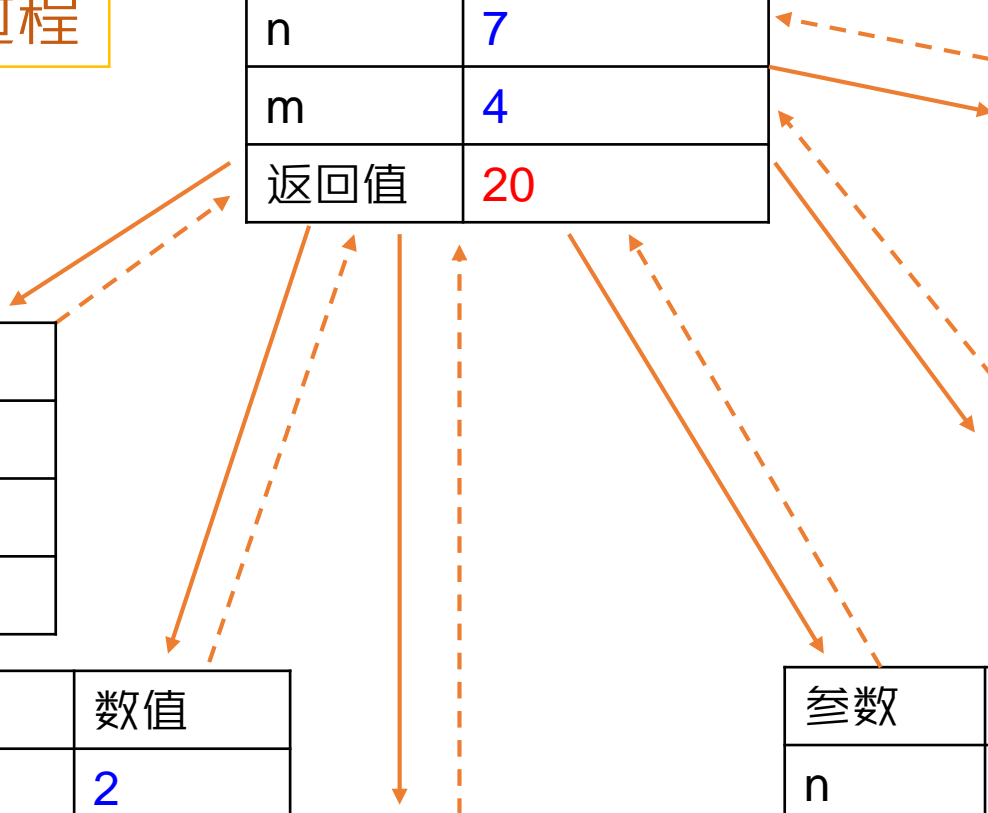
参数	数值
n	5
m	3
返回值	6

参数	数值
n	1
m	3
返回值	0

参数	数值
n	2
m	3
返回值	0

参数	数值
n	3
m	3
返回值	1

参数	数值
n	4
m	3
返回值	3



2

用树状图表示递归过程

参数	数值
n	2
m	3
返回值	0



参数	数值
n	1
m	2
返回值	0

2

用树状图表示递归过程

参数	数值
n	3
m	3
返回值	1

参数	数值
n	2
m	2
返回值	1

参数	数值
n	1
m	1
返回值	1

参数	数值
n	1
m	2
返回值	0

2

用树状图表示递归过程

参数	数值
n	4
m	3
返回值	3

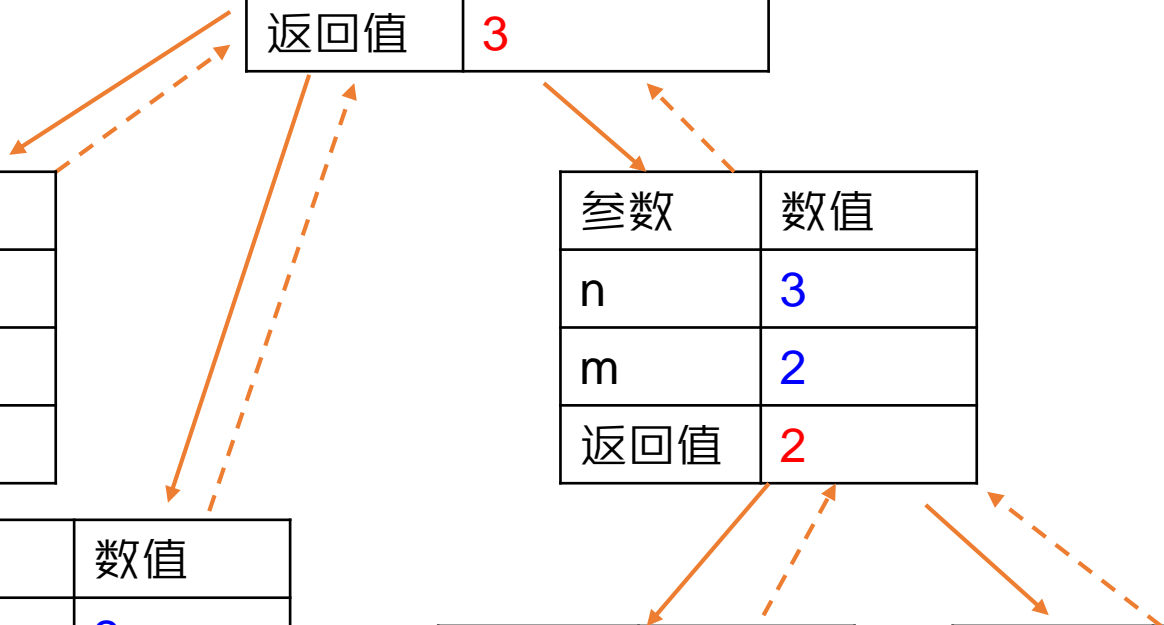
参数	数值
n	1
m	2
返回值	0

参数	数值
n	3
m	2
返回值	2

参数	数值
n	2
m	2
返回值	1

参数	数值
n	1
m	1
返回值	1

参数	数值
n	2
m	1
返回值	1



2

用树状图表示递归过程

参数	数值
n	5
m	3
返回值	6

参数	数值
n	3
m	2
返回值	2

参数	数值
n	1
m	2
返回值	0

参数	数值
n	4
m	2
返回值	3

参数	数值
n	3
m	1
返回值	1

参数	数值
n	2
m	2
返回值	1

参数	数值
n	1
m	1
返回值	1

参数	数值
n	2
m	1
返回值	1

2

用树状图表示递归过程

参数	数值
n	1
m	2
返回值	0

参数	数值
n	2
m	2
返回值	1

参数	数值
n	1
m	1
返回值	1

参数	数值
n	3
m	2
返回值	2

参数	数值
n	2
m	1
返回值	1

参数	数值
n	4
m	2
返回值	3

参数	数值
n	5
m	2
返回值	4

参数	数值
n	6
m	3
返回值	10

2

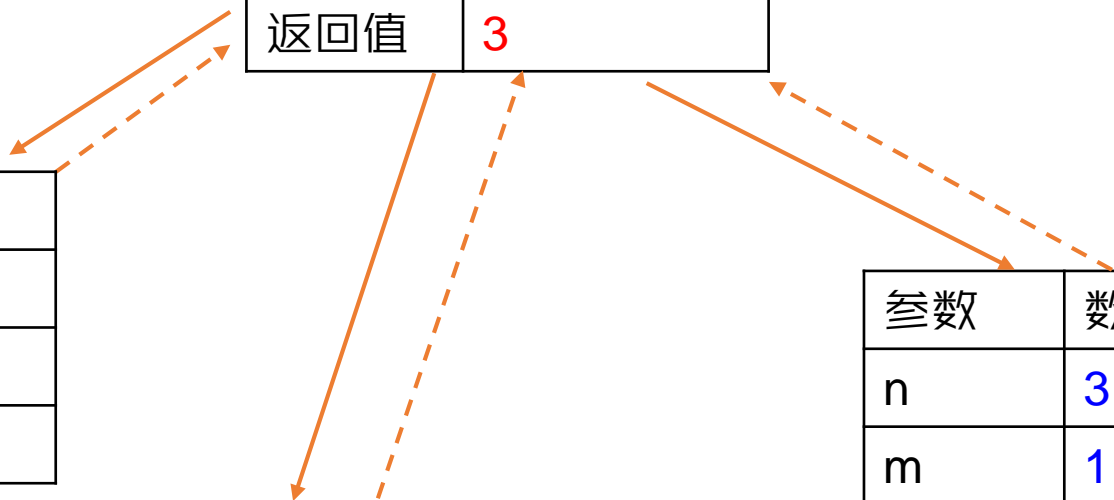
用树状图表示递归过程

参数	数值
n	4
m	2
返回值	3

参数	数值
n	1
m	1
返回值	1

参数	数值
n	2
m	1
返回值	1

参数	数值
n	3
m	1
返回值	1



2

用树状图表示递归过程

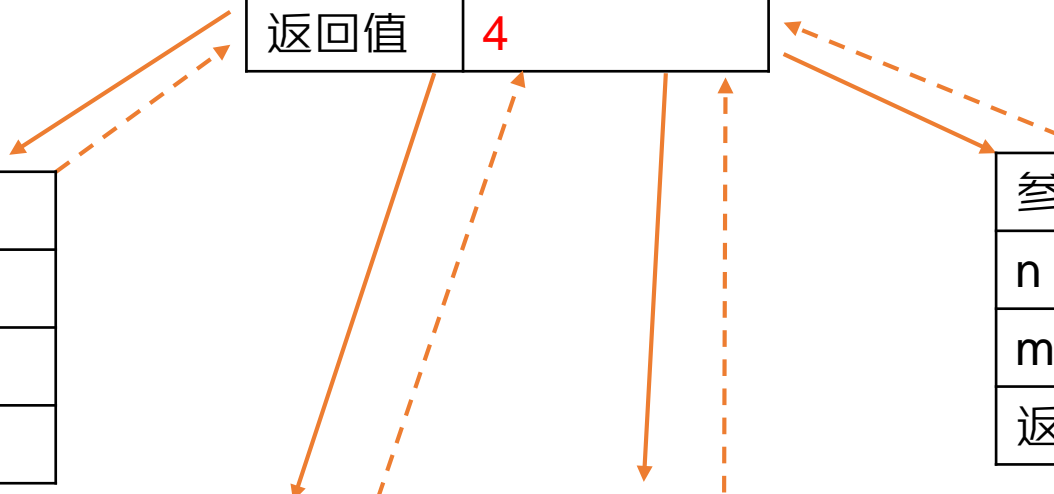
参数	数值
n	5
m	2
返回值	4

参数	数值
n	1
m	1
返回值	1

参数	数值
n	2
m	1
返回值	1

参数	数值
n	3
m	1
返回值	1

参数	数值
n	3
m	1
返回值	1



201004

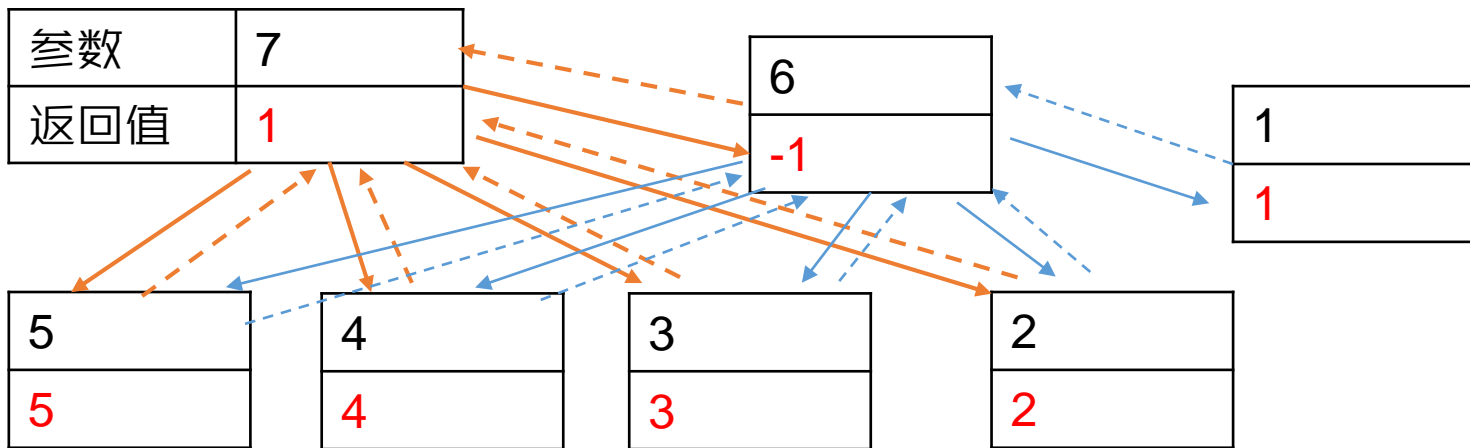
```
#include<iostream>
using namespace std;
const int NUM=5;
int r(int n) {
    int i;
    if (n<=NUM)
        return n;
    for (i=1; i<=NUM; i++)
        if (r(n-i)<0)
            return i;
    return -1;
}
int main() {
    int n;
    cin>>n;
    cout<<r(n)<<endl;
    return 0;
}
```

输入： 7

输出：

输入： 16

输出：



参数	返回值
16	4
调用	
r(15)	3
r(14)	2
r(13)	1
r(12)	-1

参数	返回值
15	3
调用	
r(14)	2
r(13)	1
r(12)	-1

参数	返回值
14	2
调用	
r(13)	1
r(12)	-1

参数	返回值
13	1
调用	
r(12)	-1

参数	返回值
11	5
调用	
r(10)	4
r(9)	3
r(8)	2
r(7)	1
r(6)	-1

参数	返回值
12	-1
调用	
r(11)	5
r(10)	4
r(9)	3
r(8)	2
r(7)	1

参数	返回值
6	-1
5	5
4	4
3	3
2	2
1	1

找规律

201203

```
#include <iostream>
using namespace std;
int n,i,j,a[100][100];
int solve(int x,int y) {
    int u,v;
    if(x==n) return a[x][y];
    u=solve(x+1,y);
    v=solve(x+1,y+1);
    if(u>v) return a[x][y]+u;
    else return a[x][y]+v;
}
int main() {
    cin>>n;
    for(i=1;i<=n;i++)
        for(j=1;j<=i;j++) cin>>a[i][j];
    cout<<solve(1,1)<<endl;
    return 0;
}
```

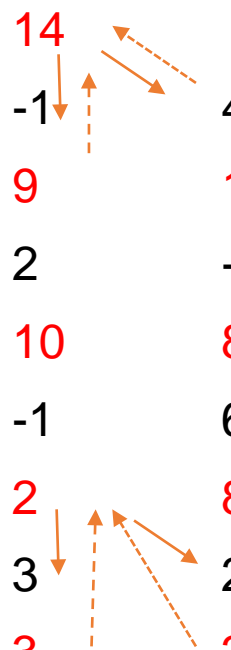
输入:

5
2
-1 4
2 -1 -2
-1 6 4 0
3 2 -1 5 8

输出:

201203

x\y	1	2	3	4	5
1	2				
返回值	14				
2	-1	4			
返回值	9	12			
3	2	-1	-2		
返回值	10	8	7		
4	-1	6	4	0	
返回值	2	8	9	8	
5	3	2	-1	5	8
返回值	3	2	-1	5	8



作业

阅读程序写结果严禁手工录入程序，运行计算结果。要自己模拟比赛环境，利用草稿纸手算结果。

请将作业打印（如有条件）或在电脑上使用文本文件填写，老师上课前1天发答案，家长按照答案批改督促。

每到题目都要严格按照讲课要求绘制递归调用过程。

请在老师发答案前独立完成！