

# 一维数组1

---

## 人员

---

刘峰烁、李沛都、罗启宸、吴青汉、王奕皓、周苇杰、陶汇笙、张昱霖、郭栩睿、锁喜栋 到课  
温郝冬、康佳 请假

## 作业

---

noi 1.6 06, noi 1.5 26

## 课堂表现

---

所有同学上课听讲都非常认真，实现代码也都很好，对所有人提出表扬！！

## 课堂内容

---

noi 1.5 28:分离整数的各个数位

```
#include <iostream>

using namespace std;

int main()
{
    int n; cin >> n;
    while (n > 0) {
        cout << n%10 << " ";
        n /= 10;
    }
    return 0;
}
```

noi 1.4 15:最大数输出

```
// 方式一
#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;

int main()
{
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    if (a>=b&&a>=c) {
        printf("%d",a);
    }
    else if (b>=c) {
        printf("%d",b);
    }
    else {
        printf("%d",c);
    }
    return 0;
}
```

```
// 方式二
#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;

int main()
{
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    int t = a;
    if (b > t) {
        t = b;
    }
    if (c > t) {
        t = c;
    }
    cout << t << endl;
    return 0;
}
```

noi 1.5 06:整数序列的元素最大跨度值

```

#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;

int main()
{
    int n, x;
    cin >> n;
    int maxt = 0; // 想求最大值，一开始定义为最小
    int mint = 1000; // 想求最小值，一开始定义为最大
    for (int i = 1; i <= n; i++) {
        cin >> x;
        if (x > maxt) {
            maxt = x;
        }
        if (x < mint) {
            mint = x;
        }
    }
    cout << maxt - mint << endl; // 输出 最大值 - 最小值
    return 0;
}

```

## noi 1.6 06:校门外的树

可以开一个数组，开始把 $a[0] \sim a[L]$ 全部设为1，代表 $a[0] \sim a[L]$ 这段区间中每个位置都有1棵树  
 之后建 $m$ 个地铁，每个地铁可以用一段 $l \sim r$ 表示  
 $m$ 个地铁可以表示为  $l_1 \sim r_1, l_2 \sim r_2, l_3 \sim r_3, \dots, l_m \sim r_m$   
 对于每一段地铁 $l \sim r$ ，可以使用循环把每个位置的 $a$ 值设为0，即代表这个位置的树被砍掉了  
 最后，使用计数器统计 $a[0] \sim a[L]$ 中有几个位置的 $a$ 值为1即可。  
 $a$ 值为1的地方就是还有树的地方。

```
#include <bits/stdc++.h>

using namespace std;

int w[10000+5];

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 0; i <= n; i++) {
        w[i] = 1;
    }
    for (int i = 1; i <= m; i++) {
        int l, r; cin >> l >> r;
        for (int j = l; j <= r; j++) {
            w[j] = 0;
        }
    }
    int sum = 0;
    for (int i = 0; i <= n; i++) {
        if (w[i] == 1) {
            sum++;
        }
    }
    cout << sum << endl;
    return 0;
}
```