

## A> 空游的草坪

---

直接模拟

```
#include <stdio.h>
#define MAXN 1234

int a[MAXN];

int main() {
    int n, m;
    scanf("%d%d", &n, &m);
    int i;
    for (i = 1; i <= n; i++)
        scanf("%d", &a[i]);
    while (m--) {
        int l, r;
        scanf("%d%d", &l, &r);
        int j, temp;
        // 对区间内草坪高度进行交换
        for (i = l, j = r; i < j; i++, j--) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    for (i = 1; i <= n; i++)
        printf("%d ", a[i]);
    return 0;
}
```

## B> 多项式求值II

---

读入的同时求导，最后一口气算出答案。

```
#include <stdio.h>
#include <math.h>
#define MAXN 1234

int c[MAXN], e[MAXN];

int main() {
    int n;
    scanf("%d", &n);
    int i;
    for (i = 0; i < n; i++) {
        scanf("%d*x^%d", &c[i], &e[i]);
    }
}
```

```
        c[i] *= e[i];    // 求导
        e[i]--;
    }
    double v;
    scanf("%lf", &v);
    double res = 0;
    for (i = 0; i < n; i++)
        res += c[i] * pow(v, e[i]); // 计算结果
    printf("%.1f", res);
    return 0;
}
```

## C> 真心话大冒险

---

计算每个人的真心话分别需要多少轮才能传递到n，取最大值输出即可

```
#include <stdio.h>
#define MAXN 1024

int a[MAXN];

int max(int a, int b) {
    return a > b ? a : b;
}

int make(int k, int n) { // 计算秘密从k传递到n需要几轮
    if (k == n) // 如果秘密已经传递到n，返回0
        return 0;
    // 否则，先传递一轮，再计算剩下需要几轮
    return make(a[k], n) + 1;
}

int main() {
    int n;
    scanf("%d", &n);
    int i;
    for (i = 1; i <= n; i++)
        scanf("%d", &a[i]);
    int res = 0;
    // 寻找需要最多轮数才能传递到n的真心话
    for (i = 1; i <= n; i++)
        res = max(res, make(i, n));
    printf("%d\n", res);
    return 0;
}
```

## D> 数分

---

先算出结果，再根据输入改变格式化字符串的内容，最后使用定制 的格式化字符串配合printf进行输出

```
#include <stdio.h>

int main() {
    int i;
    long long fac = 1;
    double res = 0;
    // 经过误差估计发现15之后的项对小数点后前9位无影响
    for (i = 1; i <= 15; i++) {
        fac *= i;
        res += 1.0 / fac;
    }
    int x;
    scanf("%d", &x);
    // 通过改变format字符串来控制输出格式
    char format[] = "%.0f\n";
    format[2] = x + '0';
    printf(format, res);
    return 0;
}
```

## E> 学长的简陋计算器

---

switch即可，注意判断wrong。

这里通过改变格式化字符串的值来控制输出格式。

```
#include <stdio.h>
#include <math.h>
#define eps 1e-8

int main() {
    double x, y;
    char op;
    scanf("%lf%c%lf", &x, &op, &y);
    char good[] = "%06.2f";
    char bad[] = "wrong\n";
    int flag = 1;
    double res;
    switch (op) {
        case '+':
            res = x + y;
            break;
        case '-':
            res = x - y;
            break;
        case '*':
            res = x * y;
```

```
        break;
    case '/':
        if (fabs(y) < eps)
            flag = 0;
        res = x / y;
        break;
    case '^':
        res = pow(x, y);
        break;
    default:
        flag = 0;
}
printf(flag ? good : bad, res);
return 0;
}
```

## F> 矩阵乘法I

根据矩阵乘法定义进行运算即可。

```
#include <stdio.h>
#define MAXN 123

int a[MAXN][MAXN];
int b[MAXN][MAXN];
int c[MAXN][MAXN];

int main() {
    int n;
    scanf("%d", &n);
    int i, j;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &a[i][j]);
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &b[i][j]);
    int k;
    for (i = 0; i < n; i++)    // 计算矩阵乘法
        for (k = 0; k < n; k++) {
            int r = a[i][k];
            for (j = 0; j < n; j++)
                c[i][j] += r * b[k][j];
        }
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++)
            printf("%d ", c[i][j]);
        putchar('\n');
    }
}
```

```
    return 0;  
}
```