

趣题测试



一只小猴买了若干个桃子。第一天他刚好吃了这些桃子的一半，又贪嘴多吃了一个；第二天他也刚好吃了剩余桃子的一半，贪嘴多吃了一个；第三天他又刚好吃了剩下的桃子的一半，并贪嘴多吃了一个。第四天起来一看，发现桃子只剩下一个了。请问小猴买了几个桃子？

```
cout << (((1 + 1) * 2 + 1) * 2 + 1) * 2 << endl;
```


信息学奥赛的评测任务是单位时间内均匀增加的。8 台评测机 30 分钟可以刚好把评测队列中的程序评测完毕，10 台评测机 6 分钟可以刚好把评测队列中的程序评测完毕，请问几台评测机可以在 10 分钟时刚好把评测队列中的程序评测完毕？

// (1) 设 原有评测任务 x 个， 每分钟均匀增加的任务为 y 个， 评测机每分钟处理的任务数量为 z 个

// (2) 由已知条件得到: $x + 30y = 8z * 30$

// (3) 由已知条件得到: $x + 6y = 10z * 6$

// (4) 解方程得到: $x = 15z, y = 7.5z$

// (5) 待求方程式: $x + 10y = ?z * 10$

// (6) 代入已知条件: $15z + 7.5z * 10 = ?z * 10$

// (7) 消去 z 待求 $? = 9$

`cout << (15 + 7.5 * 10) / 10 << endl;`

根据咕咕网校的预测，当课程定价为 110 元时，会有 10 人报名。如果课程价格每降低 1 元，就会多 1 名报名者（反之亦然）。如果希望总共能收到 3500 元学费的话，那么应该定价多少呢？已知本题有两个答案符合要求，则取定价较小的那一个。如果这个答案不是整数，则需四舍五入精确到整数。


```
// 设 x 为定价
// 根据已知条件得知:  $x * (10 + (110 - x)) =$ 
3500
// 变换方程得到:  $x * x - 120 * x + 3500 = 0$ 
// 根据二元一次方程求根公式:
double delta = sqrt(-120 * -120 - 4 * 1 *
3500);
cout << (-1 * -120 - delta) / (2 * 1) << endl;
```


小 A 有两块球形橡皮泥，一个半径是 4，一个半径是 10。他想把这两块橡皮泥揉在一起，然后塑造成一个正方体，请问这个正方体的棱长是多少？如果结果不是整数，则舍去小数点之后的数字。取 $\pi=3.141593$ 。

```
float pi = 3.141593;  
cout << int(pow(4.0 / 3 * pi * 4 * 4 * 4 + 4.0 /  
3 * pi * 10 * 10 * 10, 1.0/3)) << endl;
```


vip004

【问题描述】

国王将金币作为工资, 发放给忠诚的骑士。第一天, 骑士收到一枚金币; 之后两天(第二天和第三天), 每天收到两枚金币; 之后三天(第四、五、六天), 每天收到三枚金币; 之后四天(第七、八、九、十天), 每天收到四枚金币, ……这种工资发放模式会一直这样延续下去。当连续 N 天每天收到 N 枚金币后, 骑士会在之后的连续 $N+1$ 天里, 每天收到 $N+1$ 枚金币。

请编程计算在前 K 天里, $1 \leq K \leq 10000$, 骑士一共获得了多少枚金币。

【输入格式】

一行一个正整数 K , 表示发放金币的天数。

【输出格式】

一行一个正整数, 表示骑士收到的金币数。

【样例输入 1】

6

【样例输出 1】

14

【样例输入 2】

1000

【样例输出 2】

29820

【样例 1 说明】

骑士第一天收到一枚金币; 第二天和第三天, 每天收到两枚金币; 第四、五、六天, 每天收到三枚金币。因此一共收到 $1+2+2+3+3+3=14$ 枚金币。





```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     int day, coin = 0, cnt = 0;
6     bool flagStop = false;
7     cin >> day;
8     for (int i = 1; i <= day; i++) {
9         for (int j = 1; j <= i; j++) {
10             coin += i;
11             cnt++;
12             if (cnt >= day) {
13                 flagStop = true;
14                 break;
15             }
16         }
17         if (flagStop) break;
18     }
19     cout << coin << endl;
20     return 0;
21 }
```


vip005

【问题描述】

在社交媒体上,经常会看到针对某一个观点同意与否的民意调查以及结果。例如,对某一观点表示支持的有 1498 人,反对的有 902 人,那么赞同与反对的比例可以简单地记为 1498 : 902。

不过,如果把调查结果就以这种方式呈现出来,大多数人肯定不会满意。因为这个比例的数值太大,难以一眼看出它们的关系。对于上面这个例子,如果把比例记为 5 : 3,虽然与真实结果有一定的误差,但依然能够较为准确地反映调查结果,同时也显得比较直观。

现给出支持人数 A , 反对人数 B , 以及一个上限 L , 请将 A 比 B 化简为 A' 比 B' , 要求在 A' 和 B' 均不大于 L 且 A' 和 B' 互质(两个整数的最大公约数是 1)的前提下, $A' / B' \geq A / B$ 且 $A' / B' - A / B$ 的值尽可能小。

【输入格式】

一行三个正整数 A 、 B 、 L , 每两个正整数之间用一个空格隔开, 分别表示支持人数、反对人数以及上限。其中, $1 \leq A \leq 1000000$, $1 \leq B \leq 1000000$, $1 \leq L \leq 100$, $A/B \leq L$ 。

【输出格式】

一行两个正整数 A' 和 B' , 中间用一个空格隔开, 表示化简后的比例。

【样例输入】

1498 902 10

【样例输出】

5 3




```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int inf = pow(2, 31) - 1;
4  int main() {
5      freopen("ratio.in", "r", stdin);
6      freopen("ratio.out", "w", stdout);
7      int A, B, L, AA, BB;
8      cin >> A >> B >> L;
9      AA = 0;
10     BB = 1;
11     long double ratio = inf;
12     for (int i = 1; i <= L; i++) {
13         for (int j = 1; j <= L; j++) {
14             if ((1.0 * i / j <= L) && (1.0 * i / j >= 1.0 * A / B)
15             && ratio > 1.0 * i / j - 1.0 * A / B) {
16                 ratio = 1.0 * i / j - 1.0 * A / B;
17
18                 //朴素算法 求最大公约数
19                 /*
20                 int minAB = min(i, j);
21                 bool flagAB = true;
22                 for (int k = 2; k < minAB; k++) {
23                     if (i % k == 0 && j % k == 0) {
24                         flagAB = false;
25                         break;
26                     }
27                 }
28                 if (flagAB) {
29                     AA = i;
30                     BB = j;
31                 }
32                 */

```

//辗转相除法 求最大公约数

```

int n = i, m = j, t = 0;
while (m > 0) {
    t = n % m;
    n = m;
    m = t;
}
if (n == 1) {
    AA = i;
    BB = j;
}

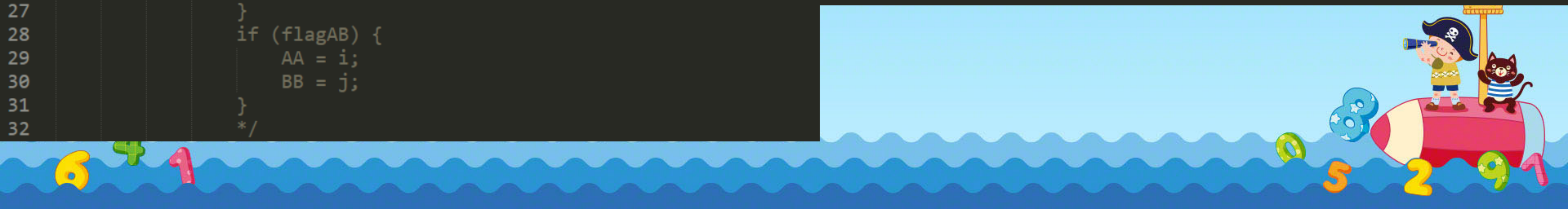
```

//cout << AA << " " << BB << " " << ratio << endl

```

}
cout << AA << " " << BB << endl;
return 0;
}

```



vip024

【问题描述】

某次科研调查时得到了 n 个自然数, 每个数均不超过 1500000000。已知不相同的数不超过 10000 个, 现在需要统计这些自然数各自出现的次数, 并按照自然数从小到大的顺序输出统计结果。

【输入格式】

第 1 行一个正整数 n , 表示自然数的个数。

第 2~ $n+1$ 行, 每行一个自然数。

【输出格式】

包含 m 行, m 为 n 个自然数中不相同数的个数。请按照自然数从小到大的顺序输出, 每行输出两个正整数, 分别是自然数和该数出现的次数, 之间用一个空格隔开。

【输入样例】

8
2
4
2
4
5
100

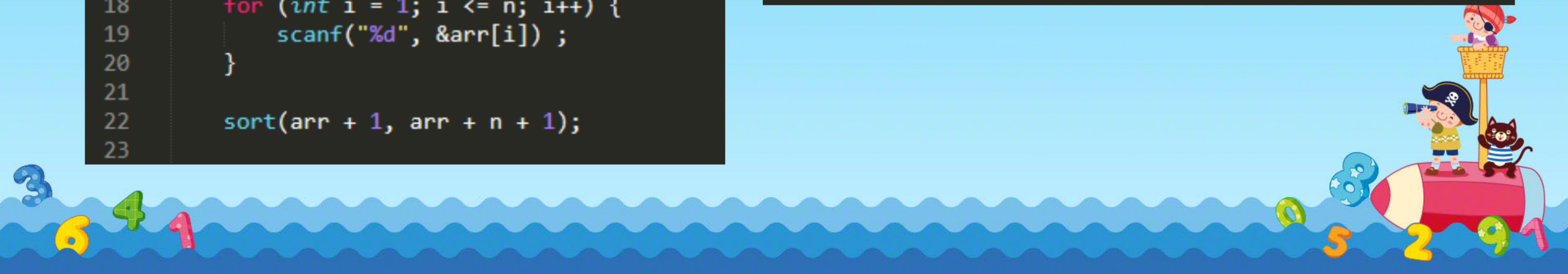
2
100
【输出样例】
2 3
4 2
5 1
100 2

【数据规模】

对于 40% 的数据满足: $1 \leq n \leq 1000$;
对于 80% 的数据满足: $1 \leq n \leq 50000$;
对于 100% 的数据满足: $1 \leq n \leq 200000$ 。



```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define MAXSIZE 200000
5
6  int arr[MAXSIZE + 1];
7
8  int main() {
9      freopen("count.in", "r", stdin);
10     freopen("count.out", "w", stdout);
11
12     int n;
13     cin >> n;
14
15     memset(arr, 0, sizeof(arr));
16     int var;
17
18     for (int i = 1; i <= n; i++) {
19         scanf("%d", &arr[i]) ;
20     }
21
22     sort(arr + 1, arr + n + 1);
23
```



```
24     arr[n + 1] = -1;
25     int count = 1;
26     for (int i = 2; i <= n + 1; i++) {
27         if (arr[i - 1] == arr[i]) {
28             count++;
29         } else {
30             printf("%d %d\n", arr[i - 1], count);
31             count = 1;
32         }
33     }
34
35     return 0;
36 }
37
```



```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define MAXSIZE 150000
4
5 char arr[MAXSIZE + 1];
6 int main() {
7     freopen("count.in", "r", stdin);
8     freopen("count.out", "w", stdout);
9     memset(arr, 0, sizeof(arr));
10    int n, num;
11    cin >> n;
12    for (int i = 1; i <= n; ++i) {
13        cin >> num;
14        arr[num] += 1;
15    }
16    for (int i = 1; i <= MAXSIZE; ++i) {
17        if (arr[i]) {
18            cout << i << " " << int(arr[i]) << endl;
19        }
20    }
21    return 0;
22 }
```





商品排序。(bsort,1s,256MB)



vip028

【问题描述】

某商场的仓库中有 n 件商品,每件商品的价格在 $0\sim 1000$ 之间(价格为 0 的商品为赠品)。现在商场经理要求将这 n 件商品按价格由低到高排序。请编程输出 n 件商品排序后的情况。

【输入格式】

第一行一个正整数 n ,表示有 n 件商品, $1\leq n\leq 100000$ 。
接下来的 n 行,每行一个整数,表示第 i 件商品的价格。

【输出格式】

n 行,每行输出一个整数。

【输入样例】

5

1

8

1

2

2

【输出样例】

1

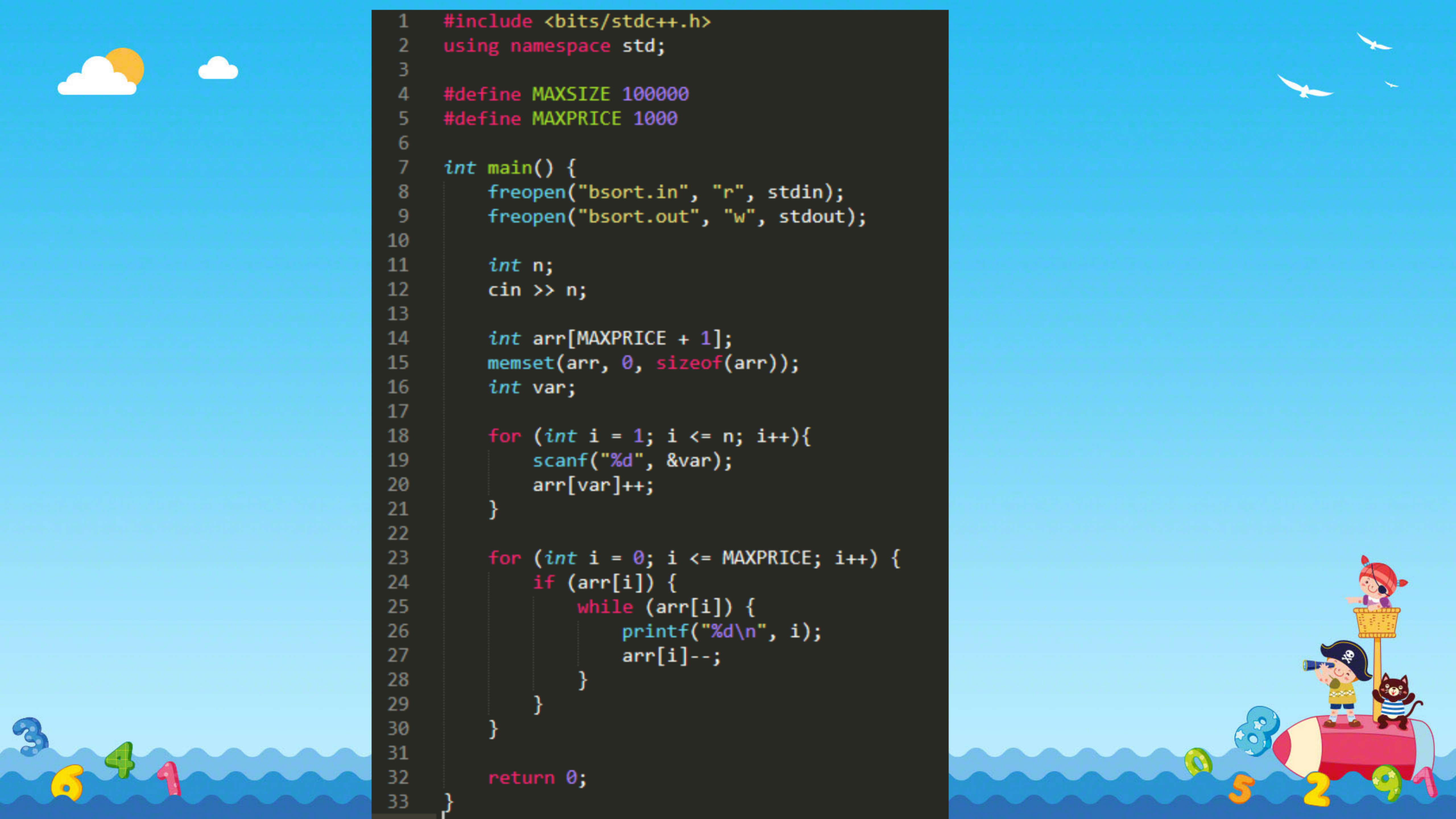
1

2

2

8





```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define MAXSIZE 100000
5  #define MAXPRICE 1000
6
7  int main() {
8      freopen("bsort.in", "r", stdin);
9      freopen("bsort.out", "w", stdout);
10
11     int n;
12     cin >> n;
13
14     int arr[MAXPRICE + 1];
15     memset(arr, 0, sizeof(arr));
16     int var;
17
18     for (int i = 1; i <= n; i++){
19         scanf("%d", &var);
20         arr[var]++;
21     }
22
23     for (int i = 0; i <= MAXPRICE; i++) {
24         if (arr[i]) {
25             while (arr[i]) {
26                 printf("%d\n", i);
27                 arr[i]--;
28             }
29         }
30     }
31
32     return 0;
33 }
```


vip091

例 2 表达式求值。(NOIP2013 普及组复赛,expr,1s,256MB)

【问题描述】

给定一个只包含加法和乘法的算术表达式,请编程计算表达式的值。

【输入格式】

输入仅有一行,为需要计算的表达式。表达式中只包含数字、加法运算符“+”和乘法运算符“*”,且没有括号,所有参与运算的数字均为 $0 \sim 2^{31}-1$ 之间的整数。输入数据保证这一行只有 0~9、+、* 这 12 种字符。

【输出格式】

输出只有一行,包含一个整数,表示这个表达式的值。注意:当答案长度多于 4 位时,请只输出最后 4 位,前导 0 不输出。

【输入样例 1】

1+1*3+4

【输出样例 1】

8

【输入样例 2】

1+1234567890*1

【输出样例 2】

7891

【输入样例 3】

1+1000000003*1

【输出样例 3】

4

【输入输出样例说明】

样例 1 计算的结果为 8,直接输出 8。

样例 2 计算的结果为 1234567891,输出后 4 位,即 7891。

样例 3 计算的结果为 1000000004,输出后 4 位,即 4。

【数据范围】

对于 30% 的数据, $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 100 。

对于 80% 的数据, $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 1000 。

对于 100% 的数据, $0 \leq$ 表达式中加法运算符和乘法运算符的总数 ≤ 100000 。



$$3 - [9 + (4 - 1) * 8]$$

诉求1：去除括号

诉求2：让运算顺序从左到右

