

青少年软件编程（C 语言）等级考试试卷（四级）

分数：100 题数：4

一、编程题(共 4 题，共 100 分)

1. 替换

题目描述

有一个包含 N 个正整数 $A_1, A_2, \dots, A_N$ 的数列 A 。接下来将进行 Q 次操作：

在第 i 次操作中，将数列中所有值为 B_i 的元素替换为 C_i 。

请针对每次操作（ $1 \leq i \leq Q$ ），输出操作后数列 A 的所有元素之和 S_i 。

输入格式

- 第一行，一个整数 N ；
- 第二行， N 个整数 $A_1, A_2, \dots, A_N$ ；
- 第三行，一个整数 Q 。
- 接下来 Q 行，每行两个整数 B_i, C_i 。

输出格式

输出 Q 行，每行一个整数 S_i ，表示每次操作后的数列总和。

注意： S_i 可能超出 32 位整数范围。

输入样例#1

```
4
1 2 3 4
3
1 2
2 4
3 4
```

输出样例#1

```
11
15
16
```

输入样例#2

```
5
1 2 3 4 5
5
1 2
2 3
3 4
4 5
5 1
```

输出样例#2

```
16
18
21
25
5
```

说明提示

【数据范围】

- $1 \leq N, Q, A_i, B_i, C_i \leq 10^5, B_i \neq C_i$ 。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-4-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

参考代码：

```
#include <iostream>
#include <cstdio>
using LL = long long;

const int MAX_SIZE = 500000;
LL a[MAX_SIZE];
LL sum = 0;

int main() {
    LL n;
    std::cin >> n;
    for (LL i = 0; i < n; i++) {
        LL k;
        std::cin >> k;
        sum += k;
        a[k]++;
    }
    LL q;
    std::cin >> q;
    for (LL i = 0; i < q; i++) {
        LL b, c;
        std::cin >> b >> c;
        LL x = a[b];
        sum = sum - x * b + x * c;
        a[b] = 0;
        a[c] = a[c] + x;
        std::cout << sum << '\n';
    }
    return 0;
}
```

考生得分：0

是否评分：已评分

评价描述：

2. 无法整除的数

题目描述

给定一个长度为 N 的数列 A 。

请计算满足以下条件的下标 i ($1 \leq i \leq N$) 的数量：

对于任意 j ($1 \leq j \leq N$ 且 $i \neq j$)，元素 A_i 不能被 A_j 整除。

输入格式

- 第一行，一个整数 N ；
- 第二行， N 个整数表示 $A_1, A_2, \dots, A_N$ 。

输出格式

输出一个整数，表示满足条件的下标数量。

输入样例#1

```
5
2 3 9 11 19
```

输出样例#1

```
4
```

输入样例#2

```
10
11 18 45 28 8 19 89 86 2 4
```

输出样例#2

```
5
```

说明提示

【数据范围】

- 所有输入均为整数；
- $1 \leq N \leq 2 \times 10^5$ ；
- $1 \leq A_i \leq 10^6$ 。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-4-02

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e6 + 10;
int n,a[N];
bool cnt[N];
signed main()
{
    scanf("%d",&n);
    for(int i = 1; i <= n; i++) scanf("%d",&a[i]);
    sort(a + 1,a + n + 1);
    int ans = 0;
    for(int i = 1; i <= n; i++)
    {
        if(!cnt[a[i]])
        {
            if(a[i] != a[i + 1]) ans++;
            for(int j = a[i]; j <= 1e6; j += a[i]) cnt[j] = true;
        }
    }
}
```

```
    }  
    }  
    printf("%d",ans);  
}
```

考生得分：0

是否评分：已评分

评价描述：

3. 咖啡机

题目描述

一台自动咖啡机按以下规则接单：制作一杯咖啡需要 c 秒，只有当前订单制作完成后，才能处理下一个订单。如果在制作期间，接到新订单，则做忽略处理。

有 n 位顾客下单，其中第 i 位顾客的下单时间为第 t_i 秒，保证所有下单时间均不重复。这台咖啡机最多能完成多少杯订单？

输入格式

- 第一行，两个整数表示 n, c ；
- 第二行， n 个整数表示 $t_1, t_2, t_3, \dots, t_n$ 。

输出格式

这台咖啡机最多能完成多少杯订单。

输入样例#1

```
6 5  
1 3 12 10 8 7
```

输出样例#1

```
3
```

输入样例#2

```
3 2
0 2 4
```

输出样例#2

```
3
```

输入样例#3

```
10 3
0 3 4 9 15 12 6 17 19 20
```

输出样例#3

```
7
```

说明提示

【数据范围】

- $1 \leq n \leq 100$;
- $1 \leq c \leq 1000$;
- $0 \leq t_i \leq 1000$ 。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-4-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e6 + 10;
int n,a[N];
bool cnt[N];
signed main()
{
    scanf("%d",&n);
    for(int i = 1; i <= n; i++) scanf("%d",&a[i]);
    sort(a + 1,a + n + 1);
    int ans = 0;
    for(int i = 1; i <= n; i++)
    {
        if(!cnt[a[i]])
        {
            if(a[i] != a[i + 1]) ans++;
            for(int j = a[i]; j <= 1e6; j += a[i]) cnt[j] = true;
        }
    }
    printf("%d",ans);
}
```

考生得分：0

是否评分：已评分

评价描述：

4. 字符数对

题目描述

给定一个由字符 `o` 和 `x` 组成的长度为 N 的字符串 S 。

请计算满足以下所有条件的整数对 (l, r) 的数量：

- $1 \leq l \leq r \leq N$ ；
- 在字符串 S 的子串 $S[l..r]$ (从第 l 个字符到第 r 个字符) 中，同时包含 `o` 和 `x` 两种字符。

输入格式

- 第一行，一个整数 N ；
- 第二行，一个字符串 S 。

输出格式

输出满足条件的整数对的数量。

输入样例^{#1}

```
4
OXXO
```

输出样例^{#1}

```
5
```

输入样例^{#2}

```
7
XOXOXX
```

输出样例^{#2}

```
19
```

说明提示

- $1 \leq N \leq 10^6$;
- S 仅由字符 o 和 x 组成。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-4-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

参考代码：

```
#include <iostream>
```

```
int main() {
    int64_t N;
    std::string S;
    std::cin >> N >> S;
    S.push_back('!');
    int64_t result = N * N;
    char last_c = S[0];
    int64_t last_id = 0;
    for (int64_t i = 1; i <= N; i++) {
        if (S[i] == last_c) continue;
        result -= (i - last_id) * (i - last_id);
        last_id = i;
        last_c = S[i];
    }
    std::cout << result / 2 << std::endl;
}
```

考生得分：0

是否评分：已评分

评价描述：