

青少年软件编程（C 语言）等级考试试卷（三级）

分数：100 题数：5

一、编程题(共 5 题，共 100 分)

1. 中位字符

题目描述

给定一个字符串 S，输出该字符串的中位字符。

中位字符指的是字符串正中间的字符。当字符串长度为奇数时，中位字符是唯一的；当长度为偶数时，中位字符是中间的两个字符。

输入格式

一个字符串 S。

输出格式

一个字符，表示 S 的中间字符。

输入样例#1

```
C++
```

输出样例#1

```
+
```

输入样例#2

```
a
```

输出样例#2

a

输入样例#3

abc

输出样例#3

b

输入样例#4

abcd

输出样例#4

bc

说明提示

【数据范围】

- $1 \leq |S| \leq 10^6$;
- $|S|$ 表示字符串的长度 , 且 S 不含任何不可见字符。

限制

时间限制 : 1000ms

内存限制 : 256MiB

试题编号 : 20250906-3-01

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 <string>
int main()
{
    std::string s;
    std::cin >> s;
    int n = s.size();
    if(n % 2 == 0) std::cout << s[n/2-1];
    std::cout << s[n/2];
}
```

考生得分：20

是否评分：已评分

评价描述：

2. 数山峰

题目描述

某地质勘探队正在研究一条山脉的 N 个连续山峰，
山峰编号为 1 到 N 。每个山峰的高度状态用字符串 S
表示： $\#$ 表示高峰， $\cdot$ 表示低谷。

地质学家需要统计存在多少山峰编号 i ($1 \leq i \leq N - 2$) 满足以下特殊地形特征：

- 山峰 i 是高峰；
- 山峰 $i + 1$ 是低谷；
- 山峰 $i + 2$ 是高峰。

即寻找所有满足“高峰-低谷-高峰”模式的连续三个山峰
组的起始位置数量。

输入格式

- 第一行：整数 N （山峰数量）；
- 第二行：长度为 N 的字符串 S （山峰高度状态）。

输出格式

满足条件的山峰编号 i 的个数。

输入样例^{#1}

```
6
#.##.##
```

输出样例^{#1}

```
2
```

输入样例^{#2}

```
1
```

#

输出样例#2

0

输入样例#3

9

##.##.##

输出样例#3

3

说明提示

【数据范围】

- $1 \leq N \leq 2 \times 10^5$;
- S 仅包含字符 # 和 .。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-3-02

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<iostream>
```

```
#include<string>
using namespace std;

int main(){
    int n;
    cin >> n;
    string s;
    cin >> s;
    int cnt=0;
    for(int i=1;i+1<n;i++){
        if(s[i-1] == '#' and s[i] == '.' and s[i+1] == '#'){
            cnt++;
        }
    }
    cout << cnt;
}
```

考生得分：20

是否评分：已评分

评价描述：

3. 移动距离

题目描述

森林里住着 262626 只小动物，它们分别对应字母 A 到 Z。这些小动物的家沿着一条直线排列，相邻两家之间的距离为 1。

它们的排列顺序记作一个字符串 S。

现在，你要按顺序去拜访 A,B,C,...,Z。每次拜访时，你需要从当前动物的家移动到目标小动物的家。

起初，你直接在动物 A 所处的位置。请计算到拜访完 Z 为止，需要移动的距离之和。

输入格式

共一行，输入一串字符串 S 。

输出格式

共一行，输出一个整数，表示从 A 出发到拜访完 Z 的最小总移动距离。

输入样例^{#1}

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

输出样例^{#1}

```
25
```

输入样例^{#2}

```
VENFLQURTCWOAXMGJYZDKSBHP
```

输出样例^{#2}

```
231
```

说明提示

【数据范围】

S 长度为 26。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-3-03

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 <iostream>

int main()
{
    int pos[30];
    char c;
    int dis=0;
    for(int i = 0 ; i < 26; ++i){
        std::cin >> c;
        pos[c - 'A'] = i;
    }
    for(int i = 1 ; i < 26; ++i){
        dis += abs(pos[i]-pos[i-1]);
    }
    std::cout << dis;
    return 0;
}
```

考生得分：20

是否评分：已评分

评价描述：

4. 扑克牌

题目描述

给定一个字符串表示扑克牌的牌面顺序，两名玩家轮流抽取卡牌。每次抽取后，立即比较双方当次抽取的牌面大小，并输出比较结果。

比较规则：

- 牌面大小顺序：

3 < 4 < 5 < 6 < 7 < 8 < 9 < T < J < Q < K < A < 2

- 若牌面相同，则判定为平局

输入格式

一个字符串，表示扑克牌的顺序

输出格式

一个字符串，依次表示每次比较的结果：

- > 表示玩家 1 的牌更大
- < 表示玩家 2 的牌更大
- = 表示平局

输入样例^{#1}

```
8632T5669A
```

输出样例^{#1}

```
><>=<
```

输入样例^{#2}

```
J2TT34K5
```

输出样例#2

<=<>

说明提示

【数据范围】

- 单张牌面字符属于集合
{3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A, 2}
- 输入字符串长度一定是偶数，且不超过 2×10^6

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-3-04

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>
#include <string>
int clc(char x){
    if(x == 'T') return 10;
    if(x == 'J') return 11;
    if(x == 'Q') return 12;
    if(x == 'K') return 13;
    if(x == 'A') return 14;
    if(x == '2') return 15;
    return x - '0';
}
int main()
{
```

```
char a,b;
while(std::cin >> a >> b){
    if(a == b) std::cout << "=";
    else{
        int x = clc(a);
        int y = clc(b);
        if(x > y) std::cout << ">";
        else std::cout << "<";
    }
}
```

考生得分：0

是否评分：已评分

评价描述：

5. 最近的数

题目描述

给定一个整数 X 和一个长度为 N 的整数序列

$p_1, p_2, \dots, p_N$ 。

请找出不在该序列中的整数（不一定为正数）中，与 X 的差的绝对值最小的那个整数。若存在多个满足条件的整数，则输出其中最小的一个。

输入格式

- 第一行：两个整数 X, N ；
- 第二行： N 个整数表示 $p_1, p_2, \dots, p_N$ 。

注意：当 $N = 0$ 时，第二行为空行。

输出格式

输出满足条件的整数。

输入样例^{#1}

```
6 5
4 7 10 6 5
```

输出样例^{#1}

```
8
```

输入样例^{#2}

```
6 5
4 7 8 6 5
```

输出样例^{#2}

```
3
```

说明提示

【数据范围】

- $0 \leq N < 100$;
- $1 \leq X, p_i < 100$ 。

限制

时间限制：1000ms

内存限制：256MiB

试题编号：20250906-3-05

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 <algorithm>
#include <cstdio>
const int inf = 0x3f3f3f3f;

int main() {

    int x, n;
    std::cin >> x >> n;
    int a[1001] = {0};
    int re;
    for (int i = 0; i < n; i++) {
        std::cin >> re;
        a[re]++;
    }

    int ans, res, ans1 = inf, res1;
    for (int i = x; i <= 101; i++) {
        if (a[i] == 0) {
            ans = i - x;
            res = i;
            break;
        }
    }
    for (int i = x - 1; i >= 0; i--) {
        if (a[i] == 0) {
            ans1 = std::min(ans1, x - i);
            res1 = i;
            break;
        }
    }
    if (ans < ans1) std::cout << res << '\n';
    else std::cout << res1 << '\n';
    return 0;
}
```

考生得分：0

是否评分：已评分

评价描述：