

2022NOC 初中组 C++ 决赛

一、单项选择题 (20 分/题，共 5 题)

1 时间复杂度为 $O(n \log n)$ 的排序算法是 ()

A. 冒泡排序 B. 归并排序 C. 计数排序 D. 选择排序

2 后缀表达式 "3 2 5 12 + * +" 的值是 ()。

A. 23 B. 25 C. 37 D. 65

3

```
1 int fun(int x)
2 {
3     if (x <= 1) return 1;
4     if (x >= 5) return x*fun(x-2);
5     return x*fun(x-1);
6 }
```

有如上函数定义，则调用 `fun(6)` 得到的返回结果为 ()

A. 720 B. 180 C. 144 D. 48

4 小于等于 30000 的正整数中，与 30000 互质的正整数有 () 个

A. 8000 B. 8500 C. 6000 D. 9000

5 插入排序算法的伪代码如下。

输入：数组 A，元素下标为 $1 \sim n$ 。

输出：按非递减顺序排序的 A。

插入排序算法：

1. for $i = 2$ to n

2. $key = A[i]$

3. $j = i - 1$

4. while $j > 0$ and $A[j] > key$

5. $A[j + 1] = A[j]$

6. $j = j - 1$

7. $A[j + 1] = key$

对 n 个数用以上排序算法进行排序，第 5 行语句最少执行 () 次，最多执行 () 次。

A. $n, n(n-1)/2$ B. $n, n(n+1)/2$ C. $0, n(n+1)/2$ D. $0, n(n-1)/2$

二、程序设计题（100分/题，共3题）

6 商场导购

【题目描述】

作为 H 国知名商人，东东在 D 城的市中心开了一家高人气商场，商场提供极其贴心的五星级导购服务。需要导购服务的顾客可以提前一天预约使用的时间段。目前有 N 位顾客预约了第二天导购服务，其中第 i 位顾客预约的时间段为 A_i 到 B_i ，注意导购服务包括了 A_i 和 B_i 两个时间段。很明显一位导购在同一个时间段只能服务一位客户。为了节约人工成本，东东希望使用最少的导购满足全部顾客的要求。请你帮他求出第二天最少需要多少位导购。

【输入格式】

输入有两行，第一行为两个正整数 N，表示预约导购服务的顾客数量。第 2 到 N+1 行，每行两个整数 A_i ， B_i ，表示第 i 位顾客预约的时间段。

【输出格式】

输出一个整数，表示最少需要多少导购，可以满足全部顾客。

【输入样例 1】

```
5
1 10
2 5
5 8
3 6
6 10
```

【输出样例 1】

```
4
```

【样例 1 说明】 由于导购服务包括边界的时间段，所以 $[2, 5]$ 和 $[5, 8]$ 必须让不同的导购负责，那么只有 $[2, 5]$ 和 $[6, 10]$ 可以请同一个导购，其他都需要单独请导购。

【输入样例 2】

```
10
24 29
11 14
5 10
26 32
4 6
27 31
39 39
39 44
18 21
18 18
```

【输出样例 2】

【数据范围】

对于 30% 的数据: $1 \leq N \leq 10$;

对于 60% 的数据: $1 \leq N \leq 100$, $1 \leq A_i \leq B_i \leq 100$;

对于 100% 的数据: $1 \leq N \leq 10000$, $1 \leq A_i \leq B_i \leq 5000$ 。

7 谜

【题目描述】

二战期间，德军使用一种名为“Enigma”的密码机。这个密码机最终被盟军破解的关键之一，是来自于使用它的人的习惯。德军每天发送的电报中，会有很多重复的词语，比如问候语、结尾的落款、每天的天气等等。盟军的破解人员把这些重复出现的词语称作“crib”。

知道了“crib”，还要知道“crib”在密文中的什么位置。由于“Enigma”是通过电路加密，所以一个字母加密后绝对不会仍然是自身，这个特性可以帮助我们定位“crib”的位置。

例如盟军截获了一条密文“XEQFTWRWEVTRES”，并且我们猜测原文中可能含有“WETTER”这个词。因为字母绝对不会被加密成自身，所以当我们把“WETTER”这个词和密文去比对，就可以找到这个词可能的位置。

在下面这个位置时，“WETTER”和密文对应的每个位置的字母都不相同，所以这条密文的原文中可能含有“WETTER”。

```
WETTER
XEQFTWRZEXGRES
```

一条密文中可能含有不止一种“crib”，同一种“crib”也可能出现多次，“crib”所占据的位置不能有重叠。给出所有可能的“crib”和一条密文，你要求出这条密文中可能包含的“crib”的最多个数。

【输入格式】

第 1 行，1 个正整数 n ，表示有 n 种“crib”。

接下来 n 行，每行一个字符串，表示 1 种“crib”。

最后 1 行，1 个字符串，表示密文。

【输出格式】

输出 1 个整数，表示密文中可能包含的“crib”的最多个数。

【输入样例 1】

```
2
WETTER
XORT
XEQFTWROEXGRES
```

【输出样例 1】

```
2
```

【输入样例 2】

```
2
CRYPTIC
OVULIST
```

CCNDDZIYXDSMTOS

【输出样例 2】

2

【输入样例 3】

3

RUBRA

DOING

GRATED

XMQWVBJAMMRGCOXQPFIZOBMDUYOXJNGTTNRJOHLD

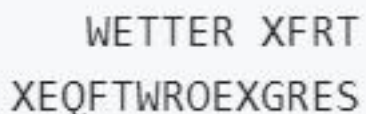
【输出样例 3】

8

【说明提示】

样例 1 说明：

如下安排两个 crib 的位置，最多包含 2 个 crib。



WETTER XFRT
XEQFTWROEXGRES

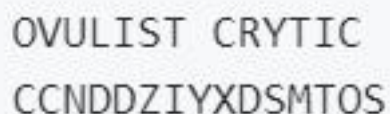
crib 也可能在如下两个位置，仍然是包含 2 个 crib。



XORT XORT
XEQFTWROEXGRES

样例 2 说明：

如下安排两个 crib 的位置，最多包含 2 个 crib。



OVULIST CRYTIC
CCNDDZIYXDSMTOS

【数据范围】

对 10%数据：n=1

另有 10%数据满足：所有“crib”的长度都等于 1。

对 100%数据：1 ≤ n ≤ 50；“crib”互不相同。“crib”的长度不超过 50，密文的长度不超过 10⁵，输入中所有字符串只含大写英文字母。

8 炫耀成绩

【题目描述】

猴帅老师所教班级刚刚进行了一次测试，猴帅老师邀请编程组的黑客老师一起判卷，其实是想秀一秀自己学生成绩。试卷从上到下编号 $1 \sim n$ ，猴帅老师对自己学生了如指掌，一下就掌握了他们的实际分数。猴帅老师为了显得不这么刻意，会主动说第 i 个试卷到第 j 个试卷自己判，其中 $1 < i \leq j < n$ ，即侯帅老师一定不会选择第 1 个试卷和第 n 个试卷，其余交给黑客老师。请帮助猴帅老师分析下，在保证猴帅老师与黑客老师都判卷至少 1 张的情况下，如何选择试卷能让黑客老师所判试卷的平均分最高。最高的平均分为多少？

【输入格式】

共 2 行

第 1 行，1 个整数 n ，代表试卷数量。

第 2 行， n 个整数 a_1, a_2, \dots, a_n ，代表每张试卷的实际分数。

【输出格式】

共 1 行，一个实数，表示最高的平均分，保留三位小数（四舍五入）。

【输入样例 1】

5

8 9 12 5 9

【输出样例 1】

9.500

【样例 1 说明】

猴帅老师把第 4 张试卷拿走自己批阅。剩下 4 张的平均分是 9.500

【输入样例 2】

6

2 7 3 1 3 5

【输出样例 2】

4.667

【数据说明】

对于 30% 数据： $n \leq 1000$;

对于 100% 数据： $10 \leq n \leq 200000$, $0 \leq a_i \leq 10000$, $1 < i \leq j < n$ 。

答案

一、单项选择题 (20 分/题，共 5 题)

1 B

2 C

3 C

4 A

5 D

二、程序设计题 (100 分/题，共 3 题)

6

```
#include<cstdio>
#include<iostream>
#include<algorithm>
#include<queue>
using namespace std;
const int N = 100005;
struct cows{ int l, r; }c[N];
bool cmp(cows a, cows b)
{
    return a.l < b.l;
}
int n, t[N];
int main()
{
    scanf("%d", &n);
    for(int i = 1; i <= n; i++)
        scanf("%d%d", &c[i].l, &c[i].r);
    sort(c+1, c+n+1, cmp);
    int cnt = 0, j = 1;
    for(int i = 1; i <= n; i++)
    {
        bool bl = 0;
        for(; j < c[i].l; j++) //查找有时间的导购
            if(t[j] > 0)
            {
                t[j]--;
                bl = 1;
                break;
            }
        if(bl == 0) cnt++; //如果没有找到，增加一个新导购
        t[c[i].r]++;
    }
    printf("%d\n", cnt);
}
```

```

        return 0;
    }
}
7
#include<cstdio>
#include<cstring>
#include<iostream>
#include<cmath>
#include<string>
#include<algorithm>
#include<vector>
#include<queue>
#include<stack>
#include<map>
using namespace std;
typedef long long LL;
const int MR=1e5+10;

bool check(const string &x,const string &y){
    if(x.size()!=y.size()) return false;
    int l=x.size();
    for(int i=0;i<l;i++){
        if(x[i]!=y[i]) return false;
    }
    return true;
}
int n,dp[MR],cL[51];
string s,crib[51];
int main(){
    cin>>n;
    for(int i=1;i<=n;i++) {
        cin>>crib[i];
        cL[i]=crib[i].size();
    }
    cin>>s;
    int len=s.size();
    for(int i=1;i<=len;i++){
        dp[i]=dp[i-1];
        for(int j=1;j<=n;j++){
            if(i>=cL[j] && check(s.substr(i-cL[j],cL[j]),crib[j])){
                dp[i]=max(dp[i-cL[j]]+1,dp[i]);
            }
        }
    }
}
cout<<dp[len]<<endl;

```



```

    return 0;
}
8
#include <cstdio>
#include <cstring>
#include <algorithm>
#include <cmath>
#include <iostream>

using namespace std;

const int NR = 2e5 + 10;

int a[NR];
int n;
long long sum;

bool check(int x)
{
    long long t = 0, ans = 9e18;
    for (int i = 2; i <= n - 1; i++)
    {
        if (t < 0) t += a[i] - x;
        else t = a[i] - x;
        ans = min(ans, t);
    }
    return ans <= sum - 1ll * n * x;
}

int main()
{
    // freopen("0.in", "r", stdin);
    // freopen("0.out", "w", stdout);

    cin >> n;

    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        a[i] *= 100000;
        sum += a[i];
    }

    // VVVVVVVVVVVVXXXXXXXXXX

```

```
int l = 0, r = 1e9, ans = 1e9;
while (l <= r)
{
    int m = (l + r) / 2;
    if (check(m)) ans = m, l = m + 1;
    else r = m - 1;
}
printf("%.3f", ans / 100000.0);
return 0;
}
```