**HƯỚNG DẪN CHẤM MÔN TIN HỌC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Tên chương trình** | **Dữ liệu vào** | **Kết quả ra** | **Điểm** |
| **Câu 1** | Demuoc.\* | Từ bàn phím | Ghi ra màn hình | 4,0 |
| **Câu 2** | Xuathien.\* | Từ bàn phím | Ghi ra màn hình | 4,0 |
| **Câu 3** | Lucky.\* | Từ bàn phím | Ghi ra màn hình | 4,0 |
| **Câu 4** | Chonso.\* | Từ bàn phím | Ghi ra màn hình | 4,0 |
| **Câu 5** | Buocnhay.\* | Từ bàn phím | Ghi ra màn hình | 4,0 |

**Câu 1. Đếm số ước của**

- Subtask 1: Học sinh chỉ cần tính Sau đó đếm số ước bằng duyệt toàn bộ các số nhỏ hơn, thuật toán như sau:

int demuoc(int n)

{

int s=0;

for(int i=1; i<=n; i++)

if(n%i==0)

s++;

return s;

}

- Subtask 2: Với N lớn hơn, hs cần cải tiến thuật toán đếm ước, như sau:

int demuoc2(int n)

int demuoc2(int n) {

int s=0;

for(int i=1; i<=sqrt(n); i++) {

if(n%i==0)

s=s+2;

if(i\*i==n)

s=s-1;

}

return s;

}

-Subtask 3: Với N lớn hơn thì không tính được, mà hs cần phải sử dụng kĩ thuật phân tích các số ra tích thừa số nguyên tố sau đó đếm số ước.

vector<int>dem;

void phantich(int n) {

for(int i=2; i<=sqrt(n); i++)

while(n%i==0) {

dem[i]++;

n=n/i;

}

if(n>1)

dem[n]++;

}

void solve(int n) {

dem.clear();

dem.resize(n+5);

for(int i=1; i<=n; i++)

phantich(i);

res=1;

for(int i=2; i<=n; i++)

res=res\*(dem[i]+1);

cout<<res;

}

**Câu 2. Chữ số xuất hiện nhiều nhất**

Bài này hs có 2 cách tiếp cận: Một là, sắp xếp các kí tự, rồi sau đó sử dụng kĩ thuật hai con trỏ để thực hiện việc tìm số xuất hiện nhiều nhất. Hai là, sử dụng kĩ thuật đếm.

Chương trình như sau:

#include<bits/stdc++.h>

using namespace std;

string s;

char kq;

int cur,dem[265];

int main() {

cin>>s;

for(char c:s) {

if(c>='0' && c<='9') {

dem[c]++;

cur=max(cur,dem[c]);

}

}

for(int i='0'; i<='9'; i++)

if(dem[i]==cur)

kq=i;

cout<<kq;

}

**Câu 3: Số may mắn**

- Subtask 1: Hs kiểm tra toàn bộ các số trong đoạn [A,B] nếu số nào chỉ chứa số 6, 8 thì tăng biến đếm kết quả lên.

Chương trình con kiểm tra số may mắn như sau:

bool check(int n) {

while(n>0) {

if(n%10!=6 && n%10!=8)

return 0;

n=n/10;

}

return 1;

}

- Subtask 2: Khi giới hạn lớn, ta bắt buộc phải sinh ra dãy các số may mắn, sau đó thì kiểm tra xem có bao nhiêu số thuộc đoạn [A,B].

vector<int> lucky,x,y;

x.push\_back(6);

x.push\_back(8);

void make\_lucky(vector<int> &x, vector<int> &y) {

y.clear();

for(int v:x) {

y.push\_back(v\*10+6);

lucky.push\_back(v\*10+6);

y.push\_back(v\*10+8);

lucky.push\_back(v\*10+8);

}

swap(x,y);

}

**Câu 4. Chọn số**

- Subtask 1: Dữ liệu cho vào toàn các số lẻ. Do đó tổng của chúng là lẻ, nên chúng ta chỉ xoá một số là đủ tạo ra dãy tổng chẵn. Đáp số là: N

- Subtask 2: Xét mọi cách xoá cặp chỉ số

- Subtask 3: Do số chẵn lẻ chỉ phụ thuộc vào chữ số cuối cùng, nên có lớn cũng không ảnh hưởng, sử dụng kiểu string để đọc dữ liệu vào.

- Subtask 4: Với

Gọi dem1 là số số lẻ trong dãy, dem2 là số số chẵn trong dãy.

Xét hai trường hợp sau:

\*) Nếu tổng dãy số là chẵn: Ta có thể xoá một hai số chẵn, hoặc 2 số lẻ khác nhau bất kì để thu được tổng chẵn. Do vậy kết quả là:

\*) Nếu tổng dãy số là lẻ: Ta có thể xoá một số lẻ, hoặc 1 số chẵn và 1 số lẻ để thu được tổng chẵn. Do vậy đáp số là:

Chương trình tham khảo:

#include<bits/stdc++.h>

#define int long long

using namespace std;

string s;

int n,tong,dem1,dem2,a[100005];

main() {

cin>>n;

for(int i=1; i<=n; i++) {

cin>>s;

a[i]=s[s.size()-1]-'0';

if(a[i]%2)

dem1++;

else

dem2++;

tong=tong+a[i];

}

if(tong%2)

cout<<dem1+dem1\*dem2;

else

cout<<dem1\*(dem1-1)/2+dem2\*(dem2+1)/2;

}

**Câu 5: Bước nhảy**

- Subtask 1: Xét mọi cặp chỉ số

Với mỗi cặp thoả mãn điều kiện ta cập nhật lại kết quả

- Subtask 2: Kết hợp cấu trúc dữ liệu, cách giải quyết như sau: Gọi T[i] là phần tử nhỏ nhất từ a1đến ai. Vậy T là dãy giảm dần.-Duyệt tất cả các chỉ số j từ 1 đến N. Với mỗi j ta tìm kiếm nhị phân trên trên đoạn chỉ số [1, j-1] của dãy T phần tử lớn nhất thỏa mãn ≤ a[j]-p.-Lưu ý: Ở đây dãy T là dãy giảm dần.

Độ phức tạp thuật toán:

Chương trình tham khảo:

#include<bits/stdc++.h>

using namespace std;

int n,res,p,a[100005],T[100005];

int bs(int L, int R, int x) {

while(L<R) {

int mid=(L+R)/2;

if(T[mid]<=x-p)

R=mid;

else

L=mid+1;

}

return L;

}

int main() {

cin>>n>>p;

T[0]=INT\_MAX;

for(int i=1; i<=n; i++) {

cin>>a[i];

T[i]=min(T[i-1],a[i]);

}

for(int i=1; i<=n; i++) {

res=max(res,i-bs(1,i,a[i]));

}

cout<<res;

}

/\*

6 3

4 3 7 2 6 4

\*/

**-------------HẾT-------------**