

TỔNG QUAN BÀI THI

Bài	Tệp chương trình	Dữ liệu vào	Dữ liệu ra	Điểm
1	BAI01.*	BAI01.INP	BAI01.OUT	4,0
2	BAI02.*	BAI02.INP	BAI02.OUT	4,0
3	BAI03.*	BAI03.INP	BAI03.OUT	4,0
4	BAI04.*	BAI04.INP	BAI04.OUT	4,0
5	BAI05.*	BAI05.INP	BAI05.OUT	4,0

Thí sinh có thể sử dụng ngôn ngữ lập trình Pascal hoặc C++ để giải các bài toán. Khi đó dấu * trong Tệp chương trình sẽ là *pas* hoặc *cpp*.

Bài 1 (4,0 điểm) Số tam giác.

Một số được gọi là “số tam giác” nếu ta có thể biểu diễn nó dưới dạng lưới hình tam giác gồm các điểm sao cho các điểm tạo thành một tam giác đều, tức là hàng đầu tiên có một điểm, hàng thứ hai có hai điểm, hàng thứ ba có ba điểm,...

Các số tam giác bắt đầu là 1, 3 (1+2), 6 (1+2+3), 10 (1+2+3+4),.... Xem hình minh họa (“số tam giác”

là số được phân tích thành tổng các số từ 1 tới một số k tương ứng).

Cho biết số nguyên dương N , hãy cho biết N có phải là số tam giác hay không?

Dữ liệu vào: Đọc dữ liệu vào từ tệp **BAI01.INP** là một số nguyên dương N ($1 \leq N \leq 10^6$).

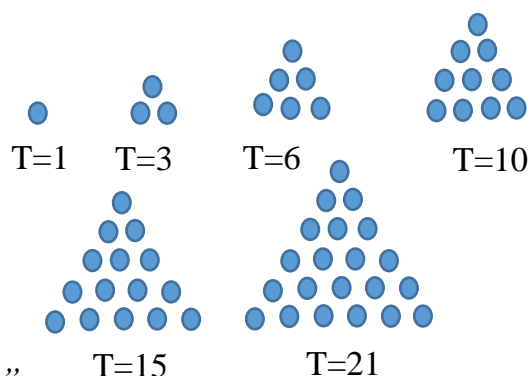
Dữ liệu ra: Ghi kết quả ra tệp **BAI01.OUT** là số 1 nếu N là số tam giác, là số 0 nếu N không phải là số tam giác.

Ví dụ:

BAI01.INP	BAI01.OUT
3	1
4	0
6	1

Bài 2 (4,0 điểm) Trò chơi.

Vì tình hình dịch Covid-19 đang diễn biến phức tạp, Tít phải học trực tuyến tại nhà. Tít cũng là người chấp hành rất tốt quy định về phòng chống dịch nên không ra ngoài đi chơi. Tít đã nghĩ ra một trò chơi với bộ sưu tập gồm n viên đá của mình. Đó là xếp n viên đá lên



một bảng hình chữ nhật chia thành lưới ô vuông đơn vị, sao cho mỗi ô có không quá một viên đá.

Ví dụ: Với $n = 100$, Tít có thể xếp chúng vào bảng kích thước 1×100 , 2×50 , 4×25 , 5×20 và 10×10 .

Yêu cầu: Xác định kích thước của bảng có chu vi nhỏ nhất mà Tít có thể thực hiện được trò chơi (nếu có nhiều kích thước thỏa mãn chu vi nhỏ nhất, Tít sẽ chọn bảng hình vuông).

Dữ liệu vào: Đọc dữ liệu vào từ tệp **BAI02.INP** một số tự nhiên $n < 10^{12}$.

Dữ liệu ra: Ghi kết quả ra tệp **BAI02.OUT** ghi ra hai số cách nhau một dấu cách là độ dài hai cạnh của bảng tìm được (nếu kích thước hai cạnh khác nhau ghi số bé trước, số lớn sau).

Ví dụ:

BAI02.INP	BAI02.OUT	Giải thích
100	10 10	100 có thể sử dụng bảng 1×100 , 2×50 , 4×25 , 5×20 và 10×10 vậy bảng có kích thước 10×10 sẽ có chu vi nhỏ nhất.
24	5 5	24 có thể sử dụng bảng 1×24 , 2×12 , 3×8 , 4×6 và 5×5 . Vậy bảng có kích thước 4×6 và 5×5 là bảng có chu vi nhỏ nhất. Tuy nhiên vì Tít thích bảng hình vuông nên sẽ chọn bảng 5×5 .

Bài 3 (4,0 điểm) Sân điền kinh.

Trong một sân điền kinh, đầu sân là vạch sơn được đánh số 0 và người ta kẻ thêm n vạch sơn để thi đấu các cự ly khác nhau. Vạch sơn thứ 1 cách đầu sân a_1 mét, vạch sơn thứ 2 cách đầu sân a_2 mét, ... vạch sơn thứ n cách đầu sân a_n mét ($a_1 < a_2 < \dots < a_n$). Ban tổ chức cần tìm hai vạch sơn để thi đấu cự ly m mét.

Yêu cầu: Cho trước khoảng cách từ đầu sân đến n vạch sơn. Tìm vạch sơn thứ x và vạch sơn thứ y để thi đấu cự ly m mét (thỏa mãn $a_y - a_x = m$).

Dữ liệu vào: Đọc dữ liệu vào từ tệp **BAI03.INP**.

- Dòng thứ nhất hai số nguyên dương n và m ($1 < n \leq 10^5$, $1 < m \leq 10^6$).

- Dòng thứ hai n số nguyên dương ($0 < a_1 < a_2 < \dots < a_n$, $a_1 \leq 10^7$).

Các số nguyên trên cùng một dòng cách nhau bởi một ký tự trắng.

Dữ liệu ra: Ghi kết quả ra tệp **BAI03.OUT**.

- Số nguyên x , y tìm được. Nếu có nhiều đáp án thì ghi đáp án gần đầu sân nhất, nếu không có đáp án thì ghi -1.

Ví dụ:

BAI03.INP	BAI03.OUT	Giải thích
4 6 1 5 7 11	1 3	Có 2 kết quả là $a_1 a_3$ ($7-1=6$) và $a_2 a_4$ ($11-5=6$) lấy kết quả 1 3 vì gần vạch đầu sân

Ràng buộc:

- **Subtask 1 (75%):** $1 < n \leq 10^3$.

- **Subtask 2 (25%):** $10^3 < n \leq 10^5$.

Bài 4 (4,0 điểm) Số lớn nhất.

Tít có một người em rất nghịch ngợm. Một hôm em của Tít ngồi nghịch và viết ra 1 dãy rất dài gồm các chữ cái và chữ số. Tít liền nghĩ ra một bài toán để đố người em nghịch ngợm của mình. Đó là tách dãy trên thành tập các số với mỗi số là một đoạn liên tiếp các chữ số sau đó tìm số lớn nhất trong tập các số vừa tìm được. Bạn hãy lập trình để kiểm tra xem kết quả em của Tít đưa ra có đúng không.

Ví dụ: 123abd67dadf89 sẽ tách được thành 3 số là 123, 67 và 89 như vậy số lớn nhất là 123.

Dữ liệu vào: Đọc dữ liệu vào từ tệp **BAI04.INP** một dòng duy nhất là đoạn kí tự gồm chữ cái và chữ số (Test luôn tồn tại ít nhất 1 số, $s \leq 10^5$).

Dữ liệu ra: Ghi kết quả ra tệp **BAI04.OUT** một số duy nhất là số lớn nhất trong dãy số trên.

Ví dụ:

BAI04.INP	BAI04.OUT	Giải thích
123abd67dadf89	123	Có 3 số được tách ra là 123, 67 và 89. Vậy số lớn nhất là 123.

Ràng buộc:

- **Subtask 1 (70%):** Số lớn nhất $\leq 10^{18}$.
- **Subtask 2 (30%):** Số lớn nhất $\leq 10^{100000}$.

Bài 5 (4,0 điểm) Đi chơi.

Sau bao ngày học hành vất vả và dịch Covid-19 đã được kiểm soát, Tít cùng M người bạn của mình lên kế hoạch đi chơi. Nhà của Tít và các bạn của Tít nằm trên cùng 1 con đường, các nhà được đánh vị trí từ 1 đến N, mỗi nhà cách nhau 1 mét. Nhà của Tít ở vị trí 1 và địa điểm vui chơi ở vị trí N. Nhà M người bạn của Tít ở các vị trí a_1, a_2, \dots, a_M . Ngoài ra trên tuyến đường còn có P trạm xe buýt tại các vị trí b_1, b_2, \dots, b_P .

Từ nhà mình, Tít lần lượt đi đến nhà của các bạn mình theo kế hoạch. Tít có thể đi bằng taxi hoặc xe buýt. Với taxi, Tít có thể bắt từ bất kì vị trí nào, giá của taxi là T đồng/mét. Với xe buýt, Tít chỉ có thể bắt từ trạm này và đi đến một trạm khác, giá của xe buýt là B đồng/lượt không phân biệt khoảng cách. Do còn phải để dành tiền để đi chơi, Tít không thể lãng phí quá nhiều tiền cho việc đi lại. Bạn hãy giúp Tít tìm cách đi đón tất cả các bạn và đến điểm vui chơi với số tiền phải trả là ít nhất nhé!

Yêu cầu: Cho biết số nhà trên đường, các nhà phải đến đón, số trạm xe buýt và số tiền đi xe taxi, xe buýt, bạn hãy tìm cách đi cho Tít sao cho đến thăm đúng thứ tự các nhà và đến vị trí N với số tiền ít nhất.

Dữ liệu vào: Đọc dữ liệu vào từ tệp **BAI05.INP** có dạng:

- Dòng thứ nhất chứa các số nguyên N, M, P, T, B: Là vị trí điểm vui chơi, các nhà phải đón, số trạm xe buýt và số tiền đi taxi, xe buýt ($1 \leq N \leq 10^9 \mid 0 \leq M, P \leq 10^5 \mid 1 \leq T, B \leq 10^4$).

- Dòng thứ hai chứa M số nguyên là thứ tự các nhà phải đến, số thứ a_i là vị trí của nhà thứ i ($1 \leq a_i \leq N$). Dữ liệu cho đảm bảo không có 2 nhà trùng vị trí.
- Dòng cuối cùng chứa P số nguyên là vị trí các trạm xe buýt theo thứ tự tăng dần, số thứ b_i là vị trí của trạm thứ i , mặc định có trạm ở vị trí 1 và N ($1 \leq b_i \leq N$).

Dữ liệu ra: Ghi kết quả ra tệp **BAI05.OUT** có gồm 1 số nguyên duy nhất là số tiền ít nhất phải trả.

Ví dụ:

BAI05.INP	BAI05.OUT	Giải thích
10 2 2 1000 2000 5 8 4 7	8000	- Đầu tiên Tít đi xe buýt từ 1 đến 4. - Sau đó đi taxi từ 4 đến 5, 5 đến 8 và 8 đến 10. Tổng số tiền là $2000+1000+3000+2000=8000$ đồng.

Ràng buộc:

- **Subtask 1 (40%):** $N, M, P \leq 20$.
- **Subtask 2 (40%):** $N, M, P \leq 100$.
- **Subtask 3 (20%):** $N \leq 10^9$; $0 \leq M, P \leq 10^5$.

-----**HẾT**-----

Họ và tên thí sinh:Số báo danh.....

Chữ ký giám thị 1:Chữ ký giám thị 2:

*Thí sinh không sử dụng tài liệu
Cán bộ coi thi không giải thích gì thêm*

I, Nguyên tắc chăm:

Chấm vòng 1 hoàn toàn bằng phần mềm chấm tự động Themis. Sau khi chấm xong vòng 1, tổ trưởng thống nhất cách chấm code và xin ý kiến trưởng ban chấm thi nếu thấy kết quả điểm thi của học sinh thấp khó xét giải thì vòng 2 kết hợp chấm code của học sinh theo nguyên tắc điểm chấm code + điểm chấm Themis (Bài nào đã lấy điểm chấm Themis thì không chấm code) không làm thay đổi thứ tự xếp hạng của học sinh.

II, Quy trình chấm:

Bước 1: Chấm bài học sinh từ CD bằng phần mềm Themis.

Bước 2: Xem lại những bài đúng 1 phần, chấm ý tưởng thuật toán với số điểm không quá 70% số điểm tối đa của từng bài.

Bước 3: Xem lại những bài 0 điểm, chấm ý tưởng thuật toán, số điểm không quá 50% số điểm tối đa của từng bài.

Bước 4: Điểm bài thi = tổng điểm của từng bài sau khi đã chấm bằng Themis và chấm code (nếu có).

Chú ý: Với những bài sai sót nhỏ: sai tên tệp, khai báo thiếu mảng, quá mảng nếu sửa lỗi này chấm lại bài thí sinh bằng Themis được điểm tối đa thì trừ 0.5 điểm/ bài.

Nội dung	Điểm
<p>Bài 1 Số tam giác – 4 điểm- chấm code tối đa 2.5 điểm.</p> <p>Ý tưởng giải thuật: Có nhiều cách giải, Học sinh có thể trừ n đi lần lượt các số từ 1 tới khi không trừ được nữa ($n \leq 0$). Nếu $n=0$ thì n là số tam giác, còn không thì n không phải số tam giác.</p> <p>Code tham khảo</p> <pre>#include <bits/stdc++.h> using namespace std; int n,i=1; int main() { ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); freopen("BAI01.INP","r",stdin);</pre>	<p>0.25</p> <p>0.5</p>

<pre> freopen("BAI01.OUT","w",stdout); cin>>n; while(n>0) { n-=i; i++; } if(n==0) cout<<1; else cout<<0; } </pre>	0.25 0.5 0.5 0.5
<p>Bài 2 Trò chơi – 4 điểm- chấm code tối đa 2.5 điểm.</p> <p>Ý tưởng giải thuật: Học sinh xác định độ dài hai cạnh bằng cách lấy căn bậc hai của n là xét 3 trường hợp có thể xảy ra.</p> <p>Code tham khảo</p> <pre> #include <bits/stdc++.h> using namespace std; long long n,k; int main() { ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); freopen("BAI02.INP","r",stdin); freopen("BAI02.OUT","w",stdout); cin>>n; k=sqrt(n); if(k*k>=n) cout<<k<<" "<<k; else if((k+1)*k>=n) cout<<k<<" "<<k+1; else cout<<k+1<<" "<<k+1; } </pre>	0.25 0.5 0.25 0.5 0.5 0.5

Câu 3 Sân điền kinh (4 điểm) – chấm code tối đa 2.75 điểm.

Ý tưởng giải thuật: Có nhiều cách để giải quyết bài toán này.

+ Cách 1: Duyệt trâu, sử dụng 2 vòng for, để duyệt với mỗi $a[i]$ kiểm tra xem có tồn tại $a[j]$ (j chạy từ $i+1$) thỏa mãn điều kiện $a[j]-a[i]=m$; trường hợp xấu nhất độ phức tạp là $O(n^2)$.

+ Cách 2: Vì đây là dãy tăng nên có thể sử dụng chập nhị phân để tìm $a[j]$ ứng với $a[i]$ thỏa mãn $a[i]+a[j]=m$;

+ Cách 3: Vì $a[i] \leq 10^7$ nên có thể dùng mảng đánh dấu, với mỗi $a[i]$ vừa nhập ta kiểm tra xem có tồn tại số $a[i]-m$ chưa, nếu có duyệt lại mảng để tìm chỉ số của số $a[i]-m$, in ra 2 chỉ số và kết thúc (chú ý trường hợp $a[i]-m < 0$).

Code tham khảo

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int n,m,cnt[1000005],a[1000005];
```

```
int main()
```

```
{
```

```
    ios_base::sync_with_stdio(0);
```

```
    cin.tie(0);
```

```
    cout.tie(0);
```

```
    freopen("BAI03.INP","r",stdin);
```

```
    freopen("BAI03.OUT","w",stdout);
```

```
    cin>>n>>m;
```

```
    cnt[0]=1;
```

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

```
    {
```

```
        cin>>a[i];
```

```
        if(a[i]-m>0&&cnt[a[i]-m]==1)
```

```
        {
```

```
            int j=0;
```

```
            while(a[j]+m!=a[i])
```

```
                j++;
```

```
            cout<<j<<" "<<i;
```

```
            return 0;
```

```
        }
```

```
        cnt[a[i]]++;
```

```
    }
```

```
    cout<<-1;
```

```
}
```

0.25

0.5

0.5

0.5

0.5

0.5

Bài 4 Số lớn nhất (4,0 điểm) – chấm code tối đa 2.75 điểm.	
Ý tưởng giải thuật: 70% học sinh viết chương trình tách số bình thường sau đó tìm số lớn nhất từ các số đã tách ra. 100% điểm: học sinh định nghĩa hàm tìm Max của 2 số lớn, chú ý loại bỏ chữ số 0 thừa ở đầu số trước khi so sánh.	
Code tham khảo: <pre>#include <bits/stdc++.h> using namespace std; string Max(string s1, string s2) { if(s1.size() > s2.size()) return s1; else return s2; if(s1 >= s2) return s1; else return s2; } string s, s1, s2; int main() { ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); freopen("BAI04.INP","r",stdin); freopen("BAI04.OUT","w",stdout); cin >> s; for(int i = 0; i < s.size(); i++) if(s[i] >= '0' && s[i] <= '9') s1 += s[i]; else { while(s1[0] == '0') s1.erase(0, 1); s2 = Max(s2, s1); s1 = ""; } while(s1[0] == '0') s1.erase(0, 1); cout << Max(s2, s1); }</pre>	
	0.5
	0.25
	0.5
	0.5
	0.5
	0.5

Bài 5 Đi chơi (4,0 điểm) – chấm code tối đa 2.75 điểm.

Ý tưởng giải thuật:

- Với nhà thứ i cần đón nhất, ta tìm và lưu lại trạm xe buýt gần nhà đó nhất. (tìm kiếm nhị phân).
- Từ nhà thứ i đi đến nhà thứ $i+1$ có 2 khả năng xảy ra:
 - + Đi luôn bằng taxi
 - + Kết hợp đi taxi với xe buýt.
- Vì thế ta số tiền đi từ nhà cần đón thứ i đến nhà cần đón thứ $i+1$ là giá trị min trong hai cách đi trên.

Code tham khảo

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
#define maxn 100010
```

```
int main()
```

```
{
```

```
    freopen("BAI05.INP", "r", stdin);
```

```
    freopen("BAI05.OUT", "w", stdout);
```

```
    int n, m, p, t, b, home[maxn], bus[maxn], near[maxn], l, r, mid, id, i;
```

```
    long long res = 0;
```

```
    cin >> n >> m >> p >> t >> b;
```

```
    for(i = 1; i <= m; i++)
```

```
        cin >> home[i];
```

```
    for(i = 1; i <= p; i++)
```

```
        cin >> bus[i];
```

```
    home[0] = 1;
```

```
    home[m + 1] = n;
```

```
    bus[0] = 1;
```

```
    bus[p + 1] = n;
```

```
    for(i = 1; i <= m; i++)
```

```
    {
```

```
        l = 0;
```

```
        r = p + 1;
```

```
        id = 0;
```

```
        while (l <= r)
```

```
        {
```

```
            mid = (l + r) / 2;
```

```
            if ( abs(home[i] - bus[mid]) < abs(home[i] - bus[id]) )
```

```
                id = mid;
```

```
            if ( bus[mid] > home[i] )
```

```
                r = mid - 1;
```

```
            else if (bus[mid] < home[i] )
```

```
                l = mid + 1;
```

```
            else
```

```
                break;
```

0.25

0.25

0.25

0.25

1.0

<pre> } near[i] = bus[id]; } near[0] = 1; near[m + 1] = n; for(i = 0; i <= m; i++) { long long f1, f2, f3; f1 = (long long) abs(near[i] - home[i]) * t + b; f2 = (long long) abs(near[i + 1] - home[i + 1]) * t; f3 = (long long) abs(home[i + 1] - home[i]) * t; res = res + min(f1 + f2, f3); } cout << res; return 0; } </pre>	<p>0.5</p> <p>0.25</p>
--	------------------------

-----Hết-----