

BÀI 1. TRÒ CHƠI

* Thuật toán đề xuất:

Rất đơn giản là tìm $\max(a,b)$ hai lần và cộng chúng lại với nhau.

Chủ đề: *rẽ nhánh đơn giản*

* Chương trình minh họa bằng ngôn ngữ lập trình C++:

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

using namespace std;
int a, b, res;
int main()
{
    freopen("game.inp", "r", stdin);
    freopen("game.out", "w", stdout);
    cin >> a >> b;
    if (a > b) res = a, a--;
    else res = b, b--;
    res += max(a,b);
    cout << res;
    return 0;
}
```

* Chương trình minh họa bằng ngôn ngữ lập trình Python:

```
file = open("game.inp", "r")
a, b = map(int, file.readline().split())
file.close()
res = max(a,b)
if (a > b): a -= 1
else: b -= 1
res += max(a,b)
file = open("game.out", "w")
file.write(str(res))
file.close()
```

BÀI 2. ĐẾM SỐ

* Thuật toán đề xuất:

Một thuật toán rất tự nhiên mà chúng ta nghĩ đến đầu tiên là duyệt X từ A đến B và kiểm tra điều kiện của X để tìm kết quả.

Rõ ràng cách làm trên với độ phức tạp là $O(B - A)$ không đạt hiệu quả cao khi mà $1 \leq A, B \leq 10^{18}$

Cách giải tối ưu như sau:

Giải quyết bài toán ngược: đếm giá trị $X \in [A, B]$ và $(X : C \text{ hoặc } X : D)$. Khi đó kết quả sẽ bằng $(B - A + 1) -$ giá trị đếm vừa tìm được ở trên.

Gọi $DivCount(P, X, Y)$ là hàm đếm số lượng số nguyên dương nhỏ hơn hoặc bằng P và chia hết cho X hoặc Y .

Khi đó kết quả sẽ là: $(B - A + 1) - (DivCount(B, C, D) - DivCount(A, C, D))$

Chủ đề: số học

*** Chương trình minh họa bằng ngôn ngữ lập trình C++:**

```
#include <bits/stdc++.h>
#define ll long long
using namespace std;
ll A, B, C, D;

ll DivCount(ll P, ll X, ll Y)
{
    ll c1 = P/X;
    ll c2 = P/Y;
    ll c3 = P/((X*Y)/__gcd(X,Y));
    return c1 + c2 - c3;
}

int main()
{
    freopen("cntnum.inp", "r", stdin);
    freopen("cntnum.out", "w", stdout);
    cin >> A >> B >> C >> D;
    cout << (B - A + 1) - (DivCount(B, C, D) - DivCount(A, C, D));
    return 0;
}
```

*** Chương trình minh họa bằng ngôn ngữ lập trình Python:**

```
import math

def DivCount(P, X, Y):
    c1 = P//X
    c2 = P//Y
    c3 = P//((X*Y)//math.gcd(X,Y))
    return c1 + c2 - c3

file = open("cntnum.inp", "r")
A, B, C, D = map(int, file.readline().split())
file.close()
file = open("cntnum.out", "w")
```

```
file.write(str(B-A+1-(DivCount(B,C,D)-DivCount(A-1,C,D))))  
file.close()
```