*Code c++ hay và cần thiết*

Trang format code: <http://www.planetb.ca/syntax-highlight-word>

Mục lục

[***1,Quay lui nhi phân*** 1](#_Toc16000379)

[***2,Liệt kê hoán vị bằng quay lui*** 2](#_Toc16000380)

[***3,liệt kê tổ hợp chập k của n bằng quay lui*** 3](#_Toc16000381)

[***4,Sinh nhị phân*** 3](#_Toc16000382)

[***5,Sinh Tổ hợp chập k của n*** 4](#_Toc16000383)

[***6,Liệt kê hoán vi sinh*** 5](#_Toc16000384)

[***7,Bài tập về ước chung lớn nhất GCD(a, x, y).*** 6](#_Toc16000385)

[***8,Chia Tổng các số từ 1 đến n()*** 6](#_Toc16000386)

[***9,Bội chung nhỏ nhất của 3 số*** 7](#_Toc16000387)

## ***1,Quay lui nhi phân***

|  |  |
| --- | --- |
| 1. #include<iostream> 3. **using** **namespace** std; 4. **int** n,k,x[200]; 6. **void** output(){ 7. **for**(**int** u=0; u<n; ++u) 8. cout << x[u]; 9. cout << endl; 10. } 12. **void** backTrack(**int** i){ 13. **for**(**int** j=0; j<2; ++j){ 14. x[i] = j; 15. **if** (i==n-1 ) output(); 16. **else** backTrack(i+1); 17. } 18. } 20. **int** main(){ 21. **int** bo;cin>>bo; 22. **while**(bo--){ 23. cin>>n; 25. backTrack(0); 27. } 29. **return** 0; 30. } | In:  -Số bộ test  -Mỗi bộ test nhập vào n  Out:  -Các xấu nhi phân có độ dài n;  J chạy từ 0->1;  Gán lần lượt j vào x[i];  In ra khi I =n-1; |

## ***2,Liệt kê hoán vị bằng quay lui***

|  |  |
| --- | --- |
| 1. //liet ke hoan vi bang quay lui 2. #include<iostream> 4. **using** **namespace** std; 5. **int** n,k,x[200]; 6. **bool** chuaxet[200]={};  //mac dinh chuaxet[i]=false 7. **void** out(){ 8. **for**(**int** i=1;i<=n;++i){ 9. cout<<x[i]<<" "; 10. } 11. cout<<endl; 12. } 13. **void** hoanvi(**int** i){ 14. **for**(**int** j=1;j<=n;++j){ 15. **if**(chuaxet[j]==**false**){ 16. x[i]=j; 17. chuaxet[j]=**true**; 18. **if**(i==n) out(); 19. **else** hoanvi(i+1); 20. chuaxet[j]=**false**; 21. } 22. } 23. } 24. **int** main(){ 25. **int** bo;cin>>bo; 26. **while**(bo--){ 27. cin>>n; 28. //for(int i=1;i<=n;++i) chuaxet[i]=true; 29. hoanvi(1); 30. } 32. **return** 0; 33. } | Mảng chuaxet dùng để đánh dấu phần tử đã xuât hiện trong mang x hay chưa.  Nhưng mặc định tất cả các phần tử của chuaxet là false nên cần cài đặt cho mang này giá trị ban đầu là true  Chú ý :nếu để điều kiện là i==k thì in ra ,ta cũng không được tổ hợp chập k của n vì có lặp. |

## ***3,liệt kê tổ hợp chập k của n bằng quay lui***

|  |  |
| --- | --- |
| 1. //liet ke to hop chap k cua n 2. #include<iostream> 4. **using** **namespace** std; 5. **int** n,k,x[200]; 7. **void** out(){ 8. **for**(**int** i=1;i<=k;++i){ 9. cout<<x[i]<<" "; 11. } 12. cout<<endl; 13. } 14. **void** tohop(**int** i){ 15. **for**(**int** j=x[i-1]+1;j<=n-k+i;j++){ 16. x[i]=j; 17. **if**(i==k) out(); 18. **else** tohop(i+1); 19. } 20. } 22. **int** main(){ 23. **int** bo;cin>>bo; 24. **while**(bo--){ 25. cin>>n>>k; 26. x[0]=0;, 27. tohop(1); 28. } 30. **return** 0; 31. } | Hàm quay lui tohop();  Duyệt j từ x[i-1] +1,đến n-k+I :  Vd n=5,k=3 i=1 ->j=0+1->3;  Mỗi vị trí của x[i] :có phần tử đầu tiên lấy ở tổ hợp đầu tiên ,phần tử cuối lấy ở cấu hình cuối |

## ***4,Sinh nhị phân***

|  |  |
| --- | --- |
| 1. //sinh nhi phan 2. #include<iostream> 4. **using** **namespace** std; 5. **int** x[200]={},n; 6. **bool** ok=**true**; 7. **void** in(){ 8. cin>>n; 9. **for**(**int** i=0;i<=n;++i) x[i]=0; 10. } 11. **void** out(){ 12. **for**(**int** i=1;i<=n;++i) 13. cout<<x[i]<<" "; 14. cout<<endl; 15. } 16. **void** sinh(){ 17. **int** i=n; 18. **while**(i>0 && x[i]==1) { 19. x[i]=0; 20. i--; 21. } 22. **if**(x[i]==0 && i>0) x[i]=1; 23. **else** ok=**false**; 24. } 25. **int** main(){ 26. in(); 27. **while**(ok){ 28. out(); 29. sinh(); 30. } 32. **return** 0; 33. } | Duyệt từ cuối mảng ,nếu xi =1 thì gán x[i]=0 cho đến khi thấy x[i]=0 hoặc hết chuỗi thì dừng lại;  Nếu i>0 thì gán x[i]=1 kết thúc hàm  Nếu khác gán ok=false tức là cấu hình cuối cùng |

## ***5,Sinh Tổ hợp chập k của n***

|  |  |
| --- | --- |
| 1. //sinh to hop chap k cua n 2. #include<iostream> 4. **using** **namespace** std; 5. **int** n,k,x[200]={}; 6. **bool** ok =**true**; 7. **void** init(){ 8. cin>>n; 9. cin>>k; 10. **for**(**int** i=1;i<=k;++i){ 11. x[i]=i; 12. } 13. } 14. **void** out(){ 15. **for**(**int** i=1;i<=k;++i){ 16. cout<<x[i]<<" "; 17. } 18. cout<<endl; 19. } 20. **void** tohop(){ 21. **int** i=k; 22. **while**(i>0 && x[i]==n-k+i) i--; 23. **if**(i>0) { 24. x[i]=x[i]+1; 25. **for**(**int** j=i+1;j<=k;j++) x[j]=x[i]+j-i; 26. } 27. **else** ok=**false**; 28. } 29. **int** main(){ 30. init()  ; 31. **while**(ok){ 32. out(); 33. tohop(); 34. } 35. **return** 0; 36. } |  |

## ***6,Liệt kê hoán vi sinh***

|  |  |
| --- | --- |
| 1. //sinh hoan vi ke tiep 3. #include<iostream> 5. **using** **namespace** std; 6. **int** n,k,x[200]={}; 7. **bool** ok=**true**; 8. **void** ini(){ 9. cin>>n; 10. **for**(**int** i=1;i<n;i++){ 11. x[i]=i; 12. } 13. } 14. **void** out(){ 15. **for**(**int** i=1;i<=n;++i){ 16. cout<<x[i]<<" "; 17. } 18. cout<<endl; 19. } 20. **void** shoanvi(){ 21. **int** j=n-1; 22. **while**(j>0 && x[j]>x[j+1]) j--; 23. **if**(j>0){ 24. **int** k=n; 25. **while**(x[j]>x[k]) k--; 26. swap(x[j],x[k]); 27. **int** r=j+1,s=n; 28. **while**(r<s){ 29. swap(x[r],x[s]); 30. r++;s--; 31. } 32. } 33. **else** ok =**false**; 34. } 36. **int** main(){ 37. ini(); 38. **while**(ok){ 39. out(); 40. shoanvi(); 41. } 42. **return** 0; 43. } | Liệt kê hoán vị bằng phương pháp sinh :  Tìm j là vị trí không đúng |

## ***7,Bài tập về ước chung lớn nhất GCD(a, x, y).***

Cho ba số a, x, y. Nhiệm vụ của bạn là tìm ước số chung lớn nhất của hai số P và Q,  
trong đó P lặp lại x lần số a và Q lặp lại y lần số a. Ví dụ a =2, x = 3, y =2 thì P=222, Q=22.  
Input:  
• Dòng đầu tiên đưa vào T là số lượng bộ test.  
• T dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ ba số a, x, y phân biệt  
nhau bởi một vài khoảng trống.  
• Các số T, a, x, y thỏa mãn ràng buộc: 1≤T≤100; 1≤a, x, y≤1018;  
Output:  
• Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2 2 2 3 123 5 2 | 2  123 |

|  |  |
| --- | --- |
| 1. // GCD(a, x, y): x, y la so lan lap lai a 2. #include<bits/stdc++.h> 3. **typedef** **long** **long** **int** ll; 4. **using** **namespace** std; 5. // Return the Greatest common Divisor of two numbers. 6. ll gcd(ll a, ll b) { 7. **if** (a == 0) 8. **return** b; 9. **return** gcd(b%a, a); 10. } 11. **void** findgcd(ll a, ll x, ll y) { 12. // tim GCD(x,y) 13. ll g = gcd(x,y); 14. // in a nlan. 15. **for** (**int** i = 0; i < g; i++) 16. cout <<a; 17. cout<<endl; 18. } 20. // Driven Program 21. **int** main() { 22. ll T, a, x, y;cin>>T; 23. **while**(T--){ 24. cin>>a>>x>>y; 25. findgcd(a,x,y); 26. } 27. } | Tổng từ 1 đến n=n\*(n+1)/2;  Hàm \_\_gcd(),có trong thư viện algorithm  Hàm gcd này cx kha nhanh  Có thể dùng toán tử 3 ngôi  Để viết lại hàm |

## ***8,Chia Tổng các số từ 1 đến n()***

**GCD(n, m)**. Cho hai số tự nhiên n, m. Nhiệm vụ của bạn là xác định xem có thể chia các số từ 1 đến n thành hai tập sao cho giá trị tuyệt đối của tổng hai tập là m và tổng các phần tử của cả hai tập là các số đồng nguyên tố (co-prime : nguyên tố cùng nhau) hay không? Ví dụ n =5, m = 7 ta có kết quả là Yes vì ta chia thành 2 tập {1, 2, 3, 5} và 4 có giá trị tuyệt đối của tổng hai tập là 7 và là các số nguyên tố cùng nhau. Với n=6, m=3 ta có câu trả lời là No vì ta có thể tìm ra hai tập {1, 2, 4, 5} và{3, 6} có trị tuyệt đối của tổng là 3 tuy nhiên cặp 12=1+2+4+5 và 9=3 + 6 không là đồng nguyên tố.  
Input:  
• Dòng đầu tiên đưa vào T là số lượng bộ test.  
• T dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là bộ hai số n, m phân biệt  
nhau bởi một vài khoảng trống.  
• Các số T, n, m, thỏa mãn ràng buộc: 1≤T≤100; 1≤n,m≤1012;  
Output:  
• Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input | Output |
| 2 5 7 6 3 | Yes No |

|  |  |
| --- | --- |
| 1. #include<iostream> 2. #include<cmath> 3. #include<algorithm> 4. **using** **namespace** std; 6. **int** main(){ 7. **int** bo; 8. cin>>bo; 9. **while**(bo--){ 10. **int** n , m ; 11. cin>>n>>m ; 12. **long** **long** tong=n\*(n+1)/2; 13. **long** **long** sum1=(tong+m)/2; 14. **long** **long** sum2=tong-sum1; 15. **int** check=\_\_gcd(sum1,sum2); 16. **if**(check== 1) cout<<"YES"<<endl; 17. **else** cout<<"NO"; 18. } 20. **return** 0; 21. } | Số nguyên tố cùng nhau là số có ước chung lớn nhất bằng 1;  Đề bài nên sửa có nhiệu hai tổng bằng m; |

## ***9,Bội chung nhỏ nhất của 3 số***

**Smallest Number**. Cho bốn số nguyên dương X, Y, Z và N. Hãy tìm số nguyên dương nhỏ nhất có N chữ số chia hết đồng thời cho X, Y, Z. Ví dụ với X = 2, Y = 3, Z = 5, N = 4 ta tìm được số  
nguyên dương nhỏ nhất có 4 chữ số là 1020 chia hết cho cả 2, 3, 5.  
Input:  
• Dòng đầu tiên đưa vào số lượng test T (T≤100).  
• Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là bộ bốn số X, Y, Z, N. Các số X,  
Y, Z, N thỏa mãn ràng buộc dưới đây:  
1≤ X, Y, Z ≤105; N≤18.  
Output:  
• Đưa ra theo từng dòng kết quả mỗi test là số nguyên nhỏ nhất có N chữ số chia hết  
đồng thời cho X, Y, Z. Trong trường hợp không có số nguyên N chữ số thỏa mãn yêu  
cầu bài toán đưa ra giá trị -1.

|  |  |
| --- | --- |
| Input | Output |
| 3 2 3 5 4  4 5 6 3 3 5 7 2 | 1020 120  -1 |

|  |  |
| --- | --- |
| 1. #include<iostream> 2. #include<math.h> 3. **typedef** **long** **long** **int** ll; 5. **using** **namespace** std; 6. //ucln khong dung de quy 7. ll gcd(ll x,ll y){ 8. **while**(x!=0){ 9. ll r=y%x; 10. y=x; 11. x=r; 12. } 13. **return** y; 14. } 15. // tim boi cnn : 16. ll bcnn3(**int** x,**int** y,**int** z){ 17. ll p=gcd(x,y); 18. ll t=(x\*y)/p; 19. ll c=gcd(t,z); 20. **return** (t\*z)/c; 21. } 23. **int** main(){ 24. **int** bo ;cin>>bo; 25. **while**(bo--){ 26. **long** **long** x,y,z,n; 27. cin>>x>>y>>z>>n; 28. ll min=pow(10,n-1); 29. ll term=min\*10-1; 30. ll k=bcnn3(x,y,z); 31. //cout<<min<<endl; 32. ll r=min%k; 33. **if**(r==0); 34. **else** { 35. min=min+(k-r); 36. **if**(min>term)  min=-1; 37. } 38. cout<<min<<endl; 39. } 40. **return** 0; 41. } | Lấy số nhỏ nhất có n chữ số (min)chia cho bội chung nhỏ nhất của 3 số (k)để tìm số dư;  Lấy min +k-dư ta được số cần tìm; |