**MỤC LỤC**

[**PHẦN I: BÀI TẬP CƠ BẢN** 5](#_Toc105068305)

[**Bài 1.** **A add b (aaddb.\*)** 5](#_Toc105068306)

[**Bài 2.** **Trước và sau (tvas.\*)** 5](#_Toc105068307)

[**Bài 3.** **Basic01 (basic01.\*)** 6](#_Toc105068308)

[**Bài 4.** **Basic02 (basic02.\*)** 6](#_Toc105068309)

[**Bài 5.** **Bài tập 228 (baitap228.\*)** 7](#_Toc105068310)

[**Bài 6.** **Bài tập 229 (baitap229.\*)** 7](#_Toc105068311)

[**Bài 7.** **Basic03 (basic03.\*)** 8](#_Toc105068312)

[**Bài 8.** **Chia kẹo (chiakeo01.\*)** 8](#_Toc105068313)

[**Bài 9.** **Cho kẹo (chokeo.\*)** 9](#_Toc105068314)

[**Bài 10.** **Bài tập 231 (baitap231.\*)** 10](#_Toc105068315)

[**Bài 11.** **Số bàn học (sobh.\*)** 10](#_Toc105068316)

[**Bài 12.** **Bài tập 234 (baitap234.\*)** 11](#_Toc105068317)

[**Bài 13.** **Chữ số đầu tiên (csldt.\*)** 11](#_Toc105068318)

[**Bài 14.** **Căn bậc hai (intsqrt.\*)** 12](#_Toc105068319)

[**Bài 15.** **Chia kẹo (scandy.\*)** 12](#_Toc105068320)

[**Bài 16.** **Phương trình bậc 3 (ptb3.\*)** 13](#_Toc105068321)

[**Bài 17.** **Trang trí hàng cây (trangtri.\*)** 13](#_Toc105068322)

[**Bài 18.** **Gặp nhau (gapnhau.\*)** 14](#_Toc105068323)

[**Bài 19.** **Tìm số (getap.\*)** 15](#_Toc105068324)

[**PHẦN II: CÂU LỆNH IF** 16](#_Toc105068325)

[**Bài 20.** **Học bỏng (hb.\*)** 16](#_Toc105068326)

[**Bài 21.** **Dấu của tổng (NPZ.\*)** 16](#_Toc105068327)

[**Bài 22.** **Tích hai số lớn nhất (tich2m.\*)** 17](#_Toc105068328)

[**Bài 23.** **Giống nhau (giongnhau.\*)** 17](#_Toc105068329)

[**Bài 24.** **Hết tiết học (hthoc.\*)** 18](#_Toc105068330)

[**Bài 25.** **Số chính phương (socp.\*)** 18](#_Toc105068331)

[**Bài 26.** **Bầu cử (vote.\*)** 19](#_Toc105068332)

[**Bài 27.** **Chia kẹo 03 (chiakeo03.\*)** 19](#_Toc105068333)

[**Bài 28.** **Năm nhuận (namnhuan.\*)** 20](#_Toc105068334)

[**Bài 29.** **Bài tập 315 (baitap315.\*)** 20](#_Toc105068335)

[**Bài 30.** **Bài tập 321 (baitap321.\*)** 21](#_Toc105068336)

[**Bài 31.** **Bài tập 323 (baitap323.\*)** 22](#_Toc105068337)

[**Bài 32.** **Bài tập 327 (baitap327.\*)** 23](#_Toc105068338)

[**Bài 33.** **Phân số lớn nhất (maxps.\*)** 23](#_Toc105068339)

[**Bài 34.** **Táo và Cam (tac.\*)** 24](#_Toc105068340)

[**Bài 35.** **Mua thức ăn (pets.\*)** 24](#_Toc105068341)

[**Bài 36.** **Alibaba (alibaba.\*)** 25](#_Toc105068342)

[**Bài 37.** **Đồng xu (dongxu.\*)** 26](#_Toc105068343)

[**Bài 38.** **Đèn Led (led.\*)** 27](#_Toc105068344)

[**Bài 39.** **Trợ giúp (help.\*)** 28](#_Toc105068345)

[**PHẦN III: CẤU TRÚC LẶP** 29](#_Toc105068346)

[**Bài 40.** **Dãy số (dayso.\*)** 29](#_Toc105068347)

[**Bài 41.** **Tính tổng (sumxy.\*)** 29](#_Toc105068348)

[**Bài 42.** **Số chính phương V2 (socp\_v2.\*)** 30](#_Toc105068349)

[**Bài 43.** **Ước số chung (uschung.\*)** 30](#_Toc105068350)

[**Bài 44.** **Tổng các ước của một số (tonguoc.\*)** 31](#_Toc105068351)

[**Bài 45.** **Ước chung lớn nhất (ucln.\*)** 31](#_Toc105068352)

[**Bài 46.** **Ước số tốt nhất (bestdiv.\*)** 32](#_Toc105068353)

[**Bài 47.** **Số hoàn hảo (sohh.\*)** 33](#_Toc105068354)

[**Bài 48.** **Tổng và số lượng chữ số (tongslcs.\*)** 33](#_Toc105068355)

[**Bài 49.** **Số nút (sonut.\*)** 34](#_Toc105068356)

[**Bài 50.** **Phép cộng không nhớ (cong.\*)** 34](#_Toc105068357)

[**Bài 51.** **Phân tích thành thừa số nguyên tố (ptsnt.\*)** 35](#_Toc105068358)

[**Bài 52.** **Chia 2, trừ 1 (d2s1.\*)** 36](#_Toc105068359)

[**Bài 53.** **Thu hoạch khoai tây (PTT1.\*)** 36](#_Toc105068360)

[**Bài 54.** **Tính tổng (Total.\*)** 37](#_Toc105068361)

[**Bài 55.** **Tinh chất ma thuật (tcmt.\*)** 39](#_Toc105068362)

[**Bài 56.** **Đồng tiền vàng và tam giác (cointr.\*)** 40](#_Toc105068363)

[**Bài 57.** **Three Boxs (threebox.\*)** 40](#_Toc105068364)

[**PHẦN IV: MẢNG MỘT CHIỀU** 42](#_Toc105068365)

[**Bài 58.** **In ngược dãy số (arr01.\*)** 42](#_Toc105068366)

[**Bài 59.** **Số lượng và tổng dãy số (arr02.\*)** 42](#_Toc105068367)

[**Bài 60.** **Dãy số B (seqb.\*)** 43](#_Toc105068368)

[**Bài 61.** **Arr\_tbc (arrtbc.\*)** 44](#_Toc105068369)

[**Bài 62.** **Tìm số K (findk.\*)** 45](#_Toc105068370)

[**Bài 63.** **Min max array (MinMaxArray.\*)** 45](#_Toc105068371)

[**Bài 64.** **Vòng tròn số (vongso.\*)** 46](#_Toc105068372)

[**Bài 65.** **Tổng lớn nhất hai số (sum2n.\*)** 47](#_Toc105068373)

[**Bài 66.** **Cùng dấu (cungdau.\*)** 48](#_Toc105068374)

[**Bài 67.** **Cực đại địa phương (cddp.\*)** 49](#_Toc105068375)

[**Bài 68.** **Tổng của đoạn số (SumP.\*)** 49](#_Toc105068376)

[**Bài 69.** **Bộ 3 số (b3so.\*)** 50](#_Toc105068377)

[**Bài 70.** **Cặp số có tích lớn nhất (max2num.\*)** 51](#_Toc105068378)

[**Bài 71.** **Array Manipulation (ArrM.\*)** 51](#_Toc105068379)

[**Bài 72.** **Ăn kẹo (ecandy.\*)** 52](#_Toc105068380)

[**Bài 73.** **Fast Food (fafo.\*)** 53](#_Toc105068381)

[**Bài 74.** **Số 10 (number10.\*)** 54](#_Toc105068382)

[**PHẦN V: STRING** 56](#_Toc105068383)

[**Bài 75.** **So sánh tên hai người (str01.\*)** 56](#_Toc105068384)

[**Bài 76.** **Ký tự đầu tiên và cuối cùng (str02.\*)** 56](#_Toc105068385)

[**Bài 77.** **Xâu đảo ngược (str03.\*)** 57](#_Toc105068386)

[**Bài 78.** **Xóa ký tự trắng trong xâu (str04.\*)** 57](#_Toc105068387)

[**Bài 79.** **Xóa ký tự trắng dư thừa (str05.\*)** 58](#_Toc105068388)

[**Bài 80.** **Xóa chữ số trong xâu (str06.\*)** 58](#_Toc105068389)

[**Bài 81.** **Số từ trong xâu (str07.\*)** 59](#_Toc105068390)

[**Bài 82.** **Xâu đối xứng (str08.\*)** 60](#_Toc105068391)

[**Bài 83.** **Số lượng ký tự (str09.\*)** 61](#_Toc105068392)

[**Bài 84.** **In hoa in thường (str10.\*)** 61](#_Toc105068393)

[**Bài 85.** **Thay thế chuỗi (str11.\*)** 62](#_Toc105068394)

[**Bài 86.** **So sánh hai xâu (str12.\*)** 63](#_Toc105068395)

[**Bài 87.** **Xâu con (substr.\*)** 64](#_Toc105068396)

[**Bài 88.** **Super digit (superdigit.\*)** 65](#_Toc105068397)

[**Bài 89.** **Khoảng cách giữa hai số (kc2cs.\*)** 66](#_Toc105068398)

[**PHẦN V: TOÁN** 67](#_Toc105068399)

[**Bài 90.** **Restaurant (rest.\*)** 67](#_Toc105068400)

[**Bài 91.** **Summing the series (sns.\*)** 67](#_Toc105068401)

[**Bài 92.** **Tổng của đoạn thứ k (sumofk.\*)** 68](#_Toc105068402)

[**Bài 93.** **Connecting towns (coto.\*)** 69](#_Toc105068403)

[**Bài 94.** **Tam giác vuông (tgvuong.\*)** 70](#_Toc105068404)

[**Bài 95.** **Số nguyên tố đặc biệt (sntdb.\*)** 71](#_Toc105068405)

[**Bài 96.** **Kiểm tra số fibonacci (isfibo.\*)** 72](#_Toc105068406)

[**Bài 97.** **Số chính phương chẵn (mahl.\*)** 73](#_Toc105068407)

[**Bài 98.** **Reverse Game (rega.\*)** 74](#_Toc105068408)

[**Bài 99.** **Hai điểm xa nhất – easy (2point.\*)** 75](#_Toc105068409)

[**Bài 100.** **Chọn một số (tcone.\*)** 76](#_Toc105068410)

[**Bài 101.** **Tổng cặp số (pairsum.\*)** 78](#_Toc105068411)

# **PHẦN I: BÀI TẬP CƠ BẢN**

## **A add b (aaddb.\*)**

Cho 2 số nguyên  và

**Yêu cầu:** Tính

**Dữ liệu vào:** Hai số  và

**Giới hạn:**

**Kết quả:** giá trị của

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **output** |
| 10 5 | 15 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long a,b;  int main(){  cin>>a>>b;  cout<<a+b;  return 0;  } |

## **Trước và sau (tvas.\*)**

Liên hợp quốc thống kê và đánh số từng người một theo thứ tự xuất hiện của người đó trên trái đất. Mỗi người có một số thứ tự riêng không trùng với người khác. Bạn có số thứ tự là . Hãy xác định số thứ tự của người liền trước bạn và của người liền sau bạn.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:**

+ Dòng đầu ghi số thứ tự của người liền trước.

+ Dòng thứ hai ghi số thứ tự của người liền sau.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 120988 | 120987  120989 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  long long n ;  cin>>n;  cout<<n-1<<endl<<n+1<<endl;  return 0;  } |

## **Basic01 (basic01.\*)**

Một người đi xe đạp với tốc độ 10km/h và một người đi xe máy với tốc độ 30km/h cùng xuất phát từ một vị trí và đi cùng một hướng. Lập trình tính và đưa ra màn hình khoảng cách của hai người sau  giờ.

**Dữ liệu vào:** Số

**Giới hạn:**   nguyên dương và có giá trị không vượt quá 15

**Kết quả:** Một số nguyên duy nhất là kết quả của bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9 | 180 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int t;  cin>>t;  int ans=20\*t;  cout<<ans;  return 0;  } |

## **Basic02 (basic02.\*)**

Hiền gọi điện trao đổi bài với Minh. Cứ mỗi phút dùng điện thoại trả  đồng. Cuộc trao đổi kéo dài  phút. Hãy lập trình tính số tiền phải trả cho cuộc điện thoại này.

**Dữ liệu vào:** Hai số nguyên dương  và

**Giới hạn:**

**Kết quả:** Một số nguyên duy nhất là kết quả của bài toán

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 3 | 24 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a,t;  int main()  {  cin>>a>>t;  int ans=a\*t;  cout<<ans;  return 0;  } |

## **Bài tập 228 (baitap228.\*)**

Cho hai số thực , tính và đưa ra kết quả:

+ Số là trung bình cộng các bình phương của hai số

+ Số là trung bình cộng các trị tuyệt đối của hai số

**Dữ liệu vào:** ghi 2 số thực

**Giới hạn:** Hai số có trị tuyệt đối không vượt quá 1000 và có tối đa 2 chữ số thập phân

**Kết quả:**

+ Dòng 1 ghi số

+ Dòng 2 ghi số

Các số được ghi với độ chính xác 6 chữ số thập phân

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 13.30 0.48 | 88.560200  6.890000 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  double a, b, tbc, ttd;  int main(){  cin>>a>>b;  tbc = (a\*a + b\*b)/2;  ttd = fabs((a + b)/2);  cout<<setprecision(6)<<fixed<<tbc<<" ";  cout<<setprecision(6)<<fixed<<ttd;  } |

## **Bài tập 229 (baitap229.\*)**

Cho hai số nguyên . Hãy tính tổng, hiệu, tích, thương (chia nguyên) của hai số đó.

**Dữ liệu vào:** Hai số nguyên ;

**Kết quả:** ghi theo dạng của mỗi phép tính trên một dòng lần lượt là:

+ Dòng 1: Phép tính

+ Dòng 2: Phép tính

+ Dòng 3: Phép tính

+ Dòng 4: Phép tính

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 15 4 | 15 + 4 = 19  15 – 4 = 11  15 \* 4 = 60  15 div 4 = 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int a,b;  cin>>a>>b;  cout<<a<<" + "<<b<<" = "<<a+b<<endl;  cout<<a<<" - "<<b<<" = "<<a-b<<endl;  cout<<a<<" \* "<<b<<" = "<<a\*b<<endl;  cout<<a<<" div "<<b<<" = "<<a/b<<endl;  return 0;  } |

## **Basic03 (basic03.\*)**

Theo quy định của nhà trường, mỗi trường hợp không đeo thẻ học sinh sẽ bị phạt trừ 3 điểm thi đua của lớp, mỗi trường hợp nói chuyện trong lớp bị trừ 2 điểm và mỗi trường hợp đi muộn bị trừ 5 điểm. Sổ đầu bài ghi nhận trong tháng có  trường hợp không đeo thẻ,  trường hợp nói chuyện trong lớp và  trường hợp đi muộn.

**Yêu cầu:** Hãy tính tổng số điểm bị trừ

**Dữ liệu vào:** Một dòng ghi lần lượt các số

**Giới hạn:**

**Kết quả:** Một số nguyên duy nhất là tổng số điểm bị trừ

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 6 6 | 51 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int t, n, m;  cin >> t >> n >> m;  cout << (t\*3+n\*2+m\*5);  return 0;  } |

## **Chia kẹo (chiakeo01.\*)**

Lớp mẫu giáo có cháu. Ban phụ huynh mang đến một túi kẹo nhờ cô giáo phát cho các cháu. Trong túi có chiếc kẹo. Với lứa tuổi mầm non, điều quan trọng nhất là công bằng, vì vậy mỗi cháu sẽ nhận được số kẹo là như nhau.

Hãy xác định mỗi cháu được nhiều nhất là bao nhiêu chiếc kẹo và số lượng kẹo còn thừa trong túi là bao nhiêu nếu mỗi cháu nhận được số kẹo tối đa có thể

**Dữ liệu vào:** Hai số nguyên

**Giới hạn:**

**Kết quả:**

+ Dòng đầu cho biết số lượng kẹo nhiều nhất mỗi cháu được nhận

+ Dòng thứ hai cho biết số lượng kẹo còn thừa

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 26 132 | 5  2 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n,m;  int main(){  cin>>n>>m;  cout<<m/n<<endl<<m%n;  return 0;  } |

## **Cho kẹo (chokeo.\*)**

Ngày hôm nay Tí đi xem phim cùng Tèo. Như thường lệ, Tí mang theo gói kẹo cam và gói kẹo chanh, mỗi gói đều có cái kẹo. Trên đường đi Tí ăn hết cái kẹo cam và cái kẹo chanh. Lúc đến rạp chiếu phim Tí chia đôi số kẹo mỗi loại thành hai phần rồi cho Tèo một phần sao cho độ chênh lệch số kẹo trong mỗi phần của Tí và Tèo là ít nhất. Nếu có chênh lệch thì Tí sẽ lấy phần nhiều hơn.

**Yêu cầu:** Hãy cho biết số kẹo mỗi loại còn lại của Tí là bao nhiêu sau khi đã cho Tèo.

**Dữ liệu vào:** Gồm 5 số nguyên dương được ghi trên một dòng và giữa các số cách nhau một dấu cách. Các số trong tệp có giá trị không vượt quá 100.

**Kết quả:** Ghi hai số nguyên trên một dòng theo thứ tự là số kẹo cam và kẹo chanh của Tí sau khi đã chia cho Tèo. Giữa hai số cách nhau một dấu cách.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3 4 10 7 | 5 3 |

**Chương trình tham khảo**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int a,b,k,x,y;  cin>>a>>b>>k>>x>>y;  cout<<(a\*k-x+1)/2<<" "<<(b\*k-y+1)/2;  return 0;  } |

## **Bài tập 231 (baitap231.\*)**

|  |  |
| --- | --- |
| Cho số  Lập trình tính diện tích hình vành khuyên có bán kính ngoài là và bán kính trong là (xem hình) | A picture containing text, clock, clipart  Description automatically generated |

**Dữ liệu vào:** Hai số thực

**Giới hạn:**

**Kết quả:** Diện tích hình vành khuyên tìm được với độ chính xác 4 chữ số thập phân

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1.63 1.55 | 0.7992 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  double pi=3.1415926535897, r1,r2;  cin>>r1>>r2;  cout<<setprecision(4)<<fixed<<pi\*(r1\*r1-r2\*r2);  return 0;  } |

## **Số bàn học (sobh.\*)**

Trường mới đẹp và rộng hơn trường cũ, số phòng học cũng nhiều hơn 3 so với trước. Nhà trường dự định tuyển thêm học sinh cho 3 lớn mới với số lượng học sinh mỗi lớp tương ứng là . Dĩ nhiên, cần mua bàn ghế mới cho 3 lớp này. Mỗi bộ bàn ghế học có không quá 2 học sinh ngồi.

Hãy xác định số lượng bộ bàn ghế tối thiểu cần mua

**Dữ liệu vào:** Nhập từ bàn phím ba số nguyên

**Kết quả:** Số lượng bộ bàn ghế tối thiểu cần mua

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 20 21 22 | 32 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int a,b,c;  cin>>a>>b>>c;  cout<<(a+1)/2+(b+1)/2+(c+1)/2;  return 0;  } |

## **Bài tập 234 (baitap234.\*)**

Cho số nguyên có đúng 4 chữ số, hãy sử dụng các phép toán ***Chia lấy nguyên*** và ***Chia lấy dư*** để tính tổng các chữ số của .

**Dữ liệu vào:** Số nguyên dương

**Giới hạn:** có 4 chữ số

**Kết quả:** tổng các chữ số của

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **output** |
| 1234 | 10 |  | 2222 | 8 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n,s=0;  cin>>n;  s+=n%10; n/=10;  s+=n%10; n/=10;  s+=n%10; n/=10;  s+=n%10;  cout<<s;  return 0;  } |

## **Chữ số đầu tiên (csldt.\*)**

Cho số thực , hãy cho biết chữ số đầu tiên của số sau dấu chấm thập phân.

**Dữ liệu vào:** Số

**Kết quả:** Chữ số tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12.64 | 6 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  double x;  int z;  cin>>x;  long long y=x\*10;  z=y%10;  cout<<abs(z);  return 0;  } |

## **Căn bậc hai (intsqrt.\*)**

Cho số nguyên dương

**Yêu cầu:** Hãy tính căn bậc hai của

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Một số nguyên duy nhất cho biết phần nguyên của

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 100 | 10 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  long long n;  cin >>n;  int t=sqrt(n);  cout <<t;  return 0;  } |

## **Chia kẹo (scandy.\*)**

Po có kẹo cam, Pi có kẹo táo. Hai bạn chia sẻ kẹo với nhau bằng cách chia số kẹo mình có thành 2 phần sao cho số lượng kẹo trong hai phần lệch nhau ít nhất rồi đưa cho người kia 1 phần và giữ lại cho mình phần nhiều hơn (nếu hai phần không bằng nhau).

**Yêu cầu:** Hãy cho biết sau khi chia kẹo thì mỗi bạn có bao nhiêu kẹo mỗi loại?

**Dữ liệu vào:** Ghi hai số nguyên lần lượt là và

**Kết quả:**

+ Dòng đầu ghi hai số lần lượt cho biết số kẹo cam và kẹo táo của Po sau khi chia

+ Dòng đầu ghi hai số lần lượt cho biết số kẹo cam và kẹo táo của Pi sau khi chia

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 10 | 3 5  2 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int n,m; cin>>n>>m;  cout<<(n+1)/2<<" "<<m/2<<endl;  cout<<n/2<<" "<<(m+1)/2<<endl;  return 0;  } |

## **Phương trình bậc 3 (ptb3.\*)**

Xét phương trình bậc 3 . Gọi là 3 nghiệm của phương trình này.

Theo định lý Viet:

+

+

+

Với các giá trị nguyên nhập vào từ bàn phím hãy đưa ra hệ số để phương trình nhận các số đã nhập là các nghiệm.

**Dữ liệu vào:**

+ Ba số nguyên lần lượt là

**Kết quả:**

+ Ghi lần lượt các số nguyên tìm được

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 5 3 | -10 31 -30 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  long long x1, x2, x3;  cin >> x1 >> x2 >>x3;  cout << -(x1+x2+x3) <<" "<< x1\*x2+x2\*x3+x1\*x3 <<" "<< x1\*x2\*x3;  return 0;  } |

## **Trang trí hàng cây (trangtri.\*)**

Dọc theo một con đường thẳng người ta trồng một hàng gồm cây xanh có khoảng cách đều nhau. Để trang trí cho con đường vào ban đêm, người ta gắn lên hàng cây một số bóng đèn theo quy tắc xen kẽ, cứ hai cây liền kề nhau thì một cây được gắn đèn, một cây không gắn đèn. Biết rằng để gắn một bóng đèn lên một cây thì cần chi phí với số tiền là đồng.

**Yêu cầu:** Hãy tính tổng chi phí để gắn được nhiều bóng đèn nhất cho hàng cây.

**Dữ liệu vào:** chỉ gồm một dòng ghi hai số nguyên và cách nhau một dấu cách.

**Giới hạn:**

+

+

**Kết quả:** Ghi một số nguyên là kết quả tìm được của bài toán.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 10 | 30 |

**Hướng dẫn:**

Để gắn được nhiều bóng đèn nhất lên hàng cây thì ta gắn lên các cây có vị trí 1, 3, 5, 7,….nghĩa là các cây có vị trị lẻ. Như vậy số lượng bóng đèn được sử dụng là

Pascal:

C++:

Kết quả bài toán là:

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long n,x;  int main()  {  cin>>n>>x;  cout<<((n-1)/2+1)\*x;  return 0;  } |

## **Gặp nhau (gapnhau.\*)**

Một tàu đánh cá có ngư dân bị tai nạn cần cấp cứu đã gọi điện về cơ sở y tế ở đảo gần nhất cách đó hải lý. Để có thể sớm sơ cứu, tàu đánh cá đổi hướng, đi thẳng về phía đảo với vận tốc hải lý/giờ, đồng thời từ đảo người ta cũng cho một tàu cứu nạn nhằm vào hướng tàu cá đi với vận tốc hải lý/giờ.

Xác định sau bao giờ hai tàu gặp nhau?

**Dữ liệu vào:** Nhập từ bàn phím các số nguyên

**Giới hạn:**

**Kết quả:** Một số thực cho biết kết quả của bài toán với độ chính xác 5 chữ số thập phân

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 65 12 23 | 1.85714 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int d,v1,v2;  cin>>d>>v1>>v2;  cout<<setprecision(5)<<fixed<<(double) d/(v1+v2);  return 0;  } |

## **Tìm số (getap.\*)**

Cho ba số nguyên dương

**Yêu cầu:** Hãy tìm nguyên dương sao cho

**Dữ liệu vào:**

+ Một dòng ghi lần lượt 3 số nguyên dương

Dữ liệu luôn đảm bảo tìm được .

**Kết quả:**

+ Số nguyên .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 2 9 | 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int a,b,c;  cin>>a>>b>>c;  cout<<(a+c)/(2\*b);  return 0;  } |

# **PHẦN II: CÂU LỆNH IF**

## **Học bỏng (hb.\*)**

Bạn vừa tham gia một cuộc thi lập trình do thầy LHP tổ chức và được thứ hạng . Bạn biết rằng thứ hạng từ 1 đến 10 sẽ nhận được 10 điểm, thứ hạng từ 11 đến 20 sẽ nhận được 9 điểm, các thứ hạng còn lại không nhận được điểm.

Với thứ hạng , hãy cho biết bạn nhận được bao nhiêu điểm.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Ghi số điểm mà bạn nhận được hoặc ghi -1 nến bạn không nhận được điểm.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 | 10 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int r;  cin>>r;  if (r<=10) cout<<10;  else if(r<=20) cout<<9;  else cout<<-1;  return 0;  } |

## **Dấu của tổng (NPZ.\*)**

Cho hai số thực . Hãy cho biết là số âm hay số dương hay bằng 0.

**Dữ liệu vào:** Hai số .

**Kết quả:** Ghi **Positive** nếu , ghi **Negative** nếu , ghi **Zero** nếu .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 -10 | Negative |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  double a, b;  cin>>a>>b;  if (a+b>0) cout<<"Positive";  if (a+b<0) cout<<"Negative";  if (a+b==0) cout<<"Zero";  return 0;  } |

## **Tích hai số lớn nhất (tich2m.\*)**

Cho ba số nguyên . Hãy cho biết tích lớn nhất của hai số trong ba số .

**Dữ liệu vào:** Ba số nguyên

**Giới hạn:**

**Kết quả:** Tích lớn nhất của 2 số

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 5 2 | 15 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int a, b, c;  cin>>a>>b>>c;  int t1=a\*b, t2=a\*c, t3=b\*c;  int maxx=t1;  if(t2>maxx) maxx=t2;  if(t3>maxx) maxx=t3;  cout<<maxx;  return 0;  } |

## **Giống nhau (giongnhau.\*)**

Có ba chiếc bánh với giá lần lượt là . Hãy cho biết có bao nhiêu chiếc bánh có giá giống nhau.

**Dữ liệu vào:** ba số nguyên dương (

**Kết quả:** Một số nguyên duy nhất cho biết số lượng chiếc bánh giống nhau.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 2 3 | 2 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int p1, p2, p3;  cin>>p1>>p2>>p3;  if(p1==p2 and p1==p3) cout<<3; else  if(p1==p2 or p1==p3 or p2==p3) cout<<2;  if(p1!=p2 and p1!=p3 and p2!=p3) cout<<0;  return 0;  } |

## **Hết tiết học (hthoc.\*)**

Giờ học ở trường bắt đầu từ 7h00 sáng, mỗi tiết học mất 45 phút, sau tiết 1 và 3 được nghỉ 10 phút, sau tiết 2 và 4 được nghỉ 5 phút. Hãy cho biết giờ kết thúc tiết .

**Dữ liệu vào:** Số nguyên dương .

**Giới hạn:**

**Kết quả:** hai số nguyên lần lượt cho biết giờ và phút kết thúc tiết .

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Ghi chú** |
| 2 | 8 40 | Tiết học thứ 2 kết thúc lúc 8 giờ 40 phút |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int k; cin>>k;  if (k==1) cout<<"7 45";  if (k==2) cout<<"8 40";  if (k==3) cout<<"9 30";  if (k==4) cout<<"10 25";  if (k==5) cout<<"11 15";  } |

## **Số chính phương (socp.\*)**

Số tự nhiên  được gọi là số chính phương nếu  là số tự nhiên.

**Yêu cầu:** Hãy cho biết số tự nhiên  có phải là số chính phương không?

**Dữ liệu vào:** Số  (

**Kết quả:** Ghi 1 nếu  là số chính phương, ngược lại ghi 0

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 9 | 1 |  | 8 | 0 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  long long n;  cin>>n;  long long m=sqrt(n);  cout<< (m\*m==n);  return 0;  } |

## **Bầu cử (vote.\*)**

Cuộc bầu cử mà ba ứng cử viên tham gia đã kết thúc, người thứ nhất nhận được phiếu bầu, người thứ 2 nhận được phiếu bầu, người thứ 3 nhận được phiếu bầu.

**Yêu cầu:** Hãy cho biết người thứ ba cần thêm ít nhất bao nhiêu phiếu bầu nữa để dành chiến thắng?

**Dữ liệu vào:** Ba số lần lượt cho biết số phiếu bầu nhận được của 3 người.

**Kết quả:** Một số nguyên duy nhất cho biết kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 5 1 | 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  long long a,b,c;  cin>>a>>b>>c;  int m=a;  if(m<b) m=b;  if (c>a and c>b) cout<<"0";  else cout<<m+1-c;  return 0;  } |

## **Chia kẹo 03 (chiakeo03.\*)**

Nam có 3 gói kẹo, gói thứ nhất có cái kẹo, gói thứ hai có cái kẹo, gói thứ ba có cái kẹo. Nam muốn chia 3 gói kẹo này cho hai em bé sao cho số kẹo hai em nhận được chênh lệch nhau là ít nhất. Biết rằng không được mở gói kẹo để chia.

Hãy cho biết số kẹo chênh lệch của hai em bé sau khi chia kẹo.

**Dữ liệu vào:** Ba số cho biết số lượng kẹo lần lượt trong 3 gói.

**Kết quả:** Một số nguyên duy nhất cho biết kết quả bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 4 5 | 0 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  long long a, b, c;  int main() {  cin>>a>>b>>c;int tg;  if (a > b) {tg = a; a = b; b = tg;}  if (b > c) {tg = b; b = c; c = tg;}  cout<<abs(c - a - b);  return 0;  } |

## **Năm nhuận (namnhuan.\*)**

Năm nhuận là những năm chia hết cho 400 hoặc là những năm chia hết cho 4 nhưng không chia hết cho 100. Những năm nhuận tháng 2 có 29 ngày. Đặc biệt, những năm chia hết cho 3328 được đề xuất là năm nhuận kép, tháng 2 có 30 ngày!

**Yêu cầu:** Với số nguyên dương , hãy cho biết là năm không nhuận hay năm nhuận hay là năm nhuận kép.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Ghi nếu là năm không nhuận; ghi nếu là năm nhuận; ghi nếu là năm nhuận kép.

**Ví dụ:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 1900 | No |  | 6656 | Multi leap |  | 2020 | Leap |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int n;  cin>>n;  if(n%3328==0) cout<<"Multi leap";  else if (n%400==0 or (n%4==0 and n%100!=0)) cout<<"Leap";  else cout<<" No";  return 0;  } |

## **Bài tập 315 (baitap315.\*)**

Cho 2 số nguyên khác nhau . Hãy thay số nhỏ bằng hiệu của số lớn và số nhỏ, thay số lớn bằng tổng của số lớn và số nhỏ.

**Dữ liệu vào:** Hai số nguyên (

**Kết quả:** Hai giá trị mới của

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1376 4786 | 3410 6162 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n, m;  cin>>n>>m;  if(n<m) { int tg=n; n=m-n; m=m+tg; }  else { int tg=m; m=n-m; n=n+tg; }  cout<<n<<" "<<m;  return 0;  } |

## **Bài tập 321 (baitap321.\*)**

Cho tọa độ hai đỉnh trái trên và phải dưới của một tứ giác lồi có các cạnh song song với các trục tọa độ và có tọa độ các đỉnh là nguyên. Kiểm tra xem tứ giác này có phải là hình vuông hay không?

**Dữ liệu vào:**

+ Dòng đầu tiên ghi 2 số là tọa độ của đỉnh trái trên

+ Dòng thứ hai ghi hai số () là tọa độ của đỉnh phải dưới

**Kết quả:**

+ Nếu tứ giác đã cho là hình vuông thì

* Dòng đầu ghi “HINH VUONG”
* Dòng thứ hai ghi diện tích hình tròn nội tiếp của tứ giác

+ Nếu tứ giác đã cho là hình chữ nhật thì:

* Dòng đầu ghi “HINH CHU NHAT”
* Dòng thứ hai ghi diện tích hình tròn ngoại tiếp của tứ giác

Các số thực được ghi với độ chính xác 4 chữ số thập phân

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 18 48 -25 | HINH VUONG  1452.2012 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  float x,y,u,v,a,b,c,d;  cin>>x>>y>>u>>v;  if (abs(u-x) == abs(v-y))  {  cout<<"HINH VUONG"<<'\n';  a=abs(u-x)/2;  cout<<setprecision(4)<<fixed<<a\*a\*3.1415926535897;  }  else  {  cout<<"HINH CHU NHAT"<<'\n';  b=abs(u-x);c=abs(v-y);  d=sqrt(b\*b+c\*c)/2;  cout<<setprecision(4)<<fixed<<d\*d\*3.1415926535897;  }  return 0;    } |

## **Bài tập 323 (baitap323.\*)**

Giải bóng đá phong trào của khối 11 có ba đội tham gia: . Giải được tổ chức theo thể thức thi đấu vòng tròn tính điểm: gặp , gặp , gặp . Điểm mỗi trận đươc tính theo luật của FIFA: Đội thắng được 3 điểm, đổi thua được 0 điểm, nếu hòa thì mỗi đội được 1 điểm. Kết quả trận gặp có tỉ số là – Đội ghi được bàn thắng và thủng lưới lần, kết quả trận gặp có tỉ số – Đội ghi được bàn thắng và bị thủng lưới lần, kết quả trận gặp có tỉ số – Đội ghi được bàn và bị thủng lưới lần.

Cho biết các số , hãy tính và đưa ra số điểm của mỗi đội

**Dữ liệu vào:** Các số (

**Kết quả:** Ghi ba số lần lượt là số điểm của ba đội

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9 7 7 5 4 8 | 6 0 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int p,q,r,s,u,v;  cin>>p>>q>>r>>s>>u>>v;  int da,db,dc;  da=0; db=0; dc=0;    if(p>q) da=da+3;  else if(p==q) {da++; db++;}  else db=db+3;    if(r>s) da=da+3;  else if (r==s) {da++; dc++;}  else dc=dc+3;  if(u>v) db=db+3;  else if(u==v) {db++; dc++;}  else dc=dc+3;    cout<<da<<" "<<db<<" "<<dc;  return 0;  } |

## **Bài tập 327 (baitap327.\*)**

Kết quả bầu cán sự học tập môn Tin học cho biết bạn An được phiếu bầu, bạn Vinh được phiếu và bạn Quang được phiếu. Ai cao phiếu nhất sẽ được làm cán sự. Nếu có hai bạn đạt cùng số phiếu cao nhất hoặc cả 3 bạn đạt cùng số phiếu thì phải bầu lại?

**Dữ liệu vào:** ba số nguyên dương (

**Kết quả:** Ghi “BAU LAI” nếu có hai bạn đạt cùng số phiếu cao nhất hoặc cả 3 bạn đạt cùng số phiếu, ngược lại ghi tên của bạn được làm cán sự.

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 1 2 2 | BAU LAI |  | 2 4 1 | VINH |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int p,q,r;  cin>>p>>q>>r;  if(p>q and p>r) cout<<"AN"; else  if(q>p and q>r) cout<<"VINH"; else  if(r>q and r>p) cout<<"QUANG"; else  cout<<"BAU LAI";  return 0;  } |

## **Phân số lớn nhất (maxps.\*)**

Cho 4 số nguyên dương .

**Yêu cầu:** Hãy chọn 2 trong 4 số để 2 số đã chọn tạo thành phân số có giá trị lớn nhất.

**Dữ liệu vào:** Nhập từ bàn phím lần lượt 4 số (

**Kết quả:** In giá trị 2 số đã chọn dưới dạng phân số tử/mẫu (tử mẫu)

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 2 4 9 | 9/2 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long a,b,c,d;  int main (){  cin>>a>>b>>c>>d;  long long kq1=max(max(max(a,b),c),d);  long long kq2=min(min(min(a,b),c),d);  cout<<kq1<<"/"<<kq2;  return 0;  } |

## **Táo và Cam (tac.\*)**

Biết được Jerry thích ăn táo và cam nên Tom nhờ mẹ mua một giỏ táo và một giỏ cam để tặng Jerry. Giỏ táo có quả còn giỏ cam có quả. Tom thích sự cân bằng nên muốn số lượng quả cam và quả táo chênh lệch nhau ít nhất có thể, vì vậy Tom sẽ mua thêm một ít quả cam hoặc táo. Mỗi quả (cam hoặc táo) có giá 1 đồng và Tom chỉ có thể mua tối đa quả vì Tom chỉ có đồng tiền.

**Yêu cầu:** Hãy cho biết số lượng chênh lệch ít nhất của quả cam và quả táo mà Tom mang đến tặng Jerry?

**Dữ liệu vào:** Ba số nguyên

**Kết quả:** Một số nguyên duy nhất là kết quả bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 4 1 | 0 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int n,m, k;  cin>>n>>m>>k;  int c;  if(n>m) c=n-m;  if(n<=m) c=m-n;  if(k>=c) cout<<0;  if(k<c) cout<<c-k;  } |

## **Mua thức ăn (pets.\*)**

Po nuôi con chó và con mèo nhưng trong tủ đồ ăn cho thú cưng chỉ có gói thức ăn cho chó, gói thức ăn cho mèo và gói thức ăn tổng hợp (dành cho cả chó và mèo). Hãy cho biết Po cần mua thêm ít nhất bao nhiêu gói thức ăn tổng hợp để chó và mèo của Po có đủ thức ăn. Biết răng mỗi con chó hoặc mèo ăn một gói thức ăn.

**Dữ liệu vào:** Các số ghi lần lượt trên một dòng, giá trị các số không vượt quá 100.

**Kết quả:** Một số nguyên dương cho biết kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 5 3 2 1 | 3 |

**Hướng dẫn:**

Đặt

Nếu thì chó có đủ thức ăn, nếu thì chó không có đủ thức ăn nên cần có thêm gói thức ăn.

Nếu thì mèo có đủ thức ăn, nếu thì mèo không có đủ thức ăn nên cần có thêm gói thức ăn.

Như vậy cần thêm gói thức ăn cho cả chó và mèo, lúc này ta sẽ dùng đến gói thức ăn tổng hợp: Nếu thì đủ thức ăn cho cả chó và mèo nên không cần mua thêm, ngược lại nếu thì cần mua thêm gói thức ăn

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int main(){  int n,m,a,b,c,x=0,y=0,k;  cin>>n>>m>>a>>b>>c;  if (a<n) x=n-a;  if (b<m) y=m-b;  k=x+y;  if (x+y>c ) cout<<k-c;  else cout<<0;  return 0;  } |

## **Alibaba (alibaba.\*)**

Alibaba nói rằng: anh ấy có đồng tiền gồm ba loại là vàng, bạc và đồng, trong đó số đồng tiền vàng ít hơn số đồng tiền bạc là và số đồng tiền bạc cũng ít hơn số đồng tiền đồng là .

**Yêu cầu**: Bạn hãy lập trình xác định xem Alibaba nói có đúng không? Nếu anh ấy nói đúng thì anh ấy đang có bao nhiêu đồng tiền vàng, bạc và đồng?

**Dữ liệu vào**: Hai số nguyên dương và theo đúng thứ tự.

**Kết quả**: Ba số nguyên tương ứng là số đồng tiền vàng, bạc và đồng mà Alibaba đang có. Giữa các số cách nhau đúng một dấu cách. Nếu bài toán không có lời giải (Alibaba nói sai) thì chỉ in ra số .

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 12 2 | 2 4 6 |  | 20 3 | -1 |

**Hướng dẫn:**

Gọi lần lượt là số đồng tiền vàng, bạc, đồng mà Alibaba có. Suy ra

Theo giả thiết của đề bài:

Thế vào phương trình ta được:

Để thỏa mãn điều kiện cần kiểm tra hay không

Nếu thì:

Cần phải để ý ở điều kiện , vì nên

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a,b,c,n,d;  int main(){  cin>>n>>d;  if(n%3==0) {  b=n/3;  a=b-d;  c=n-(a+b);  if(a<=0 or c<=0) cout<<-1;  else cout<<a<<" "<<b<<" "<<c;  }  else cout<<"-1";  return 0;  } |

## **Đồng xu (dongxu.\*)**

Nam có rất nhiều tờ tiền loại 1 xu và loại 2 xu. Anh ta vào cửa hàng tạp hóa để mua một gói kẹo với giá xu. Vì Nam yêu thích loại 1 xu và 2 xu như nhau nên muốn số tờ tiền của hai loại này dùng để trả chênh lệch nhau là ít nhất.

Hãy cho biết Nam cần trả bao nhiêu tờ tiền loại 1 xu và loại 2 xu để mua gói kẹo.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Một dòng ghi 2 số lần lượt cho biết số tờ tiền loại 1 xu và loại 2 xu Nam trả để mua gói kẹo.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 | 4 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n;  cin>>n;  if (n%3==0) cout<<n/3<<" "<<n/3;  if (n%3==1) cout<<n/3+1<<" "<<n/3;  if (n%3==2) cout<<n/3<<" "<<n/3+1;  } |

## **Đèn Led (led.\*)**

Alice vừa hoàn thành một hợp đồng lắp bảng hiển thị số bằng đèn LED. Mỗi chữ số được hiển thị trong một khung chữ nhật với 7 ống đèn LED. Bằng cách bật các ống đèn LED thích hợp ta có thể hiển thị chữ số bất kỳ. Số đèn sáng càng nhiều thị việc hiển thị chữ số đó càng tốn năng lượng. Ví dụ hiển thị chữ số 9 sẽ tốn năng lượng hơn hiển thị chữ số 7.

Icon

Description automatically generated

Sau khi bàn giao sản phẩm trong tay Alice còn thừa lại một số khá nhiều các khung hiểu thị số và một cục pin nguồn. Dung lượng pin cho phép bật sáng ống đèn LED. Ailce muốn dùng pin bật đúng ống đèn để hiển thị một số và số hiển thị được phải có tổng chữ số lớn nhất.

**Yêu cầu:** Hãy xác định tổng lớn nhất của các chữ số có thể bật sáng.

**Dữ liệu vào:**

+ Số nguyên dương

**Kết quả:**

+ Một số nguyên cho biết tổng chữ số lớn nhất tìm được

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7 | 11 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long n, ans;  int main(){  cin >> n;  long ans = (n/3)\*7;  n%=3;  if (n==1){  ans -= 7;  ans += 4;  }  else  if (n == 2) ans += 1;  cout << ans << "\n";  return 0;  } |

## **Trợ giúp (help.\*)**

Mẹ đang dở tay trong bếp vì vậy đề nghị Alice chạy ra siêu thị mua thêm một mặt hàng, đồng thời qua bưu điện nhận bưu phẩm.

Alice không bao giờ từ chối giúp mẹ nhưng muốn đi thật nhanh để về lập trình tiếp bài toán đang làm.

Đường đi từ nhà tới siêu thị có độ dài là , từ nhà tới bưu điên có độ dài là , từ bưu điện tới siêu thị có độ dài là . Với tay không tốc độ di chuyển của Alice là , khi mang hàng mua được hoặc bưu phẩm Alice đi với tốc độ , còn khi mang trên tay cả 2 thứ thì đi với tốc độ

Để hoàn thành nhiệm vụ được giao Alice có thể đi và mang về nhà cùng lúc cả 2 thứ hoặc lần lượt mạng từng thứ một về nhà.

Với thói quen của người lập trình Alice nhẩm tính các thực hiện nhanh nhất nhiệm vụ được giao. Hãy xác định thời gian ngắn nhất thực hiện nhiệm vụ.

**Dữ liệu vào:** Lần lượt 6 số thực

**Kết quả:** Đưa ra một số thực cho biết thời gian ít nhất Alice hoàn thành nhiệm vụ với độ chính xác 5 chữ số thập phân.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 3 4 7 6 5 | 1.49524 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  cin.tie(nullptr)->sync\_with\_stdio(false);  double a,b,c,x,y,z;  cin >> a >> b >> c >> x >> y >> z;  double res=100000001;  double tmp1=min(b,a+c);  double tmp2=min(a,b+c);  a=tmp1;  b=tmp2;  res=(a/x+a/y+b/x+b/y);  res=min(res,a/x+c/y+b/z);  cout << res;  } |

# **PHẦN III: CẤU TRÚC LẶP**

## **Dãy số (dayso.\*)**

Cho hai số nguyên và .

**Yêu cầu:** Hãy đưa ra các số nguyên nằm trên đoạn chia hết cho 2 hoặc chia cho 3 dư 1.

**Dữ liệu vào:** Hai số nguyên và ;

**Giới hạn**: ;

**Kết quả:** Đưa ra màn hình các số tìm được theo trình tự tăng dần của giá trị, các số cách nhau một dấu cách. Dữ liệu vào luôn đảm bảo có kết quả

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 15 | 6 7 8 10 12 13 14 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int a, b;  cin>>a>>b;  for(int i=a; i<=b;i++)  if(i%2==0 or i%3==1) cout<<i<<" ";  return 0;  } |

## **Tính tổng (sumxy.\*)**

Cho hai số nguyên . Hãy tính tổng

**Dữ liệu vào:** Hai số nguyên

**Kết quả:** Tổng tìm được

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 | 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int x,y;  cin>>x>>y;  long long s=0;  for (long long i=x;i<=y;i++) s=s+i\*i;  cout<<s;  return 0;  } |

## **Số chính phương V2 (socp\_v2.\*)**

Số tự nhiên được gọi là số chính phương khi căn bậc 2 của cũng là một số tự nhiên. Ví dụ là số chính phương vì là số tự nhiên.

**Yêu cầu:** Tìm các số chính phương trong đoạn từ đến .

**Dữ liệu vào:** Nhập từ bàn phím 2 số tự nhiên và .

**Kết quả:** Xuất ra màn hình tất cả các số chính phương tìm được theo thứ tự tăng dần. Dữ liệu đảm bảo luôn tìm được ít nhất 1 số chính phương.

**Ví dụ:**

|  |  |
| --- | --- |
| **Dữ liệu vào** | **Kết quả** |
| 1 20 | 1 4 9 16 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int a,b;  cin>>a>>b;  for (int i=a;i<=b;i++)  {  int m=(int)sqrt(i);  if (m\*m==i) cout<<i<<" ";  }  return 0;  } |

## **Ước số chung (uschung.\*)**

Ước số chung của hai số nguyên dương và là số nguyên mà cả và đều chia hết cho nó.Trong bài toán này chỉ xét đến các ước số chung nguyên dương của và *n*.

**Yêu cầu**: Hãy lập trình tìm các ước số chung nguyên dương của hai số và cho trước.

**Dữ liệu vào**: Hai số nguyên dương và .

**Kết quả**: In ra các ước số chung nguyên dương của và trên cùng một dòng theo thứ tự tăng dần, các số được in cách nhau một dấu cách.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  52 | 1 2 4 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int n,m;  cin>>n>>m;  for(int i=1;i<=min(n,m);i++)  if(n%i==0 and m%i==0) cout<<i<<" ";  } |

## **Tổng các ước của một số (tonguoc.\*)**

Cho số nguyên dương . Hãy tính tổng các ước của số nguyên dương

**Dữ liệu vào:** Số nguyên dương

**Giới hạn:**

**Kết quả:** Một số nguyên duy nhất là tổng các ước của

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 | 12 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  long long n, s=0;  cin >>n;  for(long long i=1;i\*i<=n;i++)  {  if(n%i==0) s=s+i+n/i;  if(i\*i==n) s-=i;  }  cout <<s;  return 0;  } |

## **Ước chung lớn nhất (ucln.\*)**

Cho 2 số nguyên dương  và . Hãy tìm ước chung lớn nhất của chúng

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương

+ Dòng thứ 2 ghi số nguyên dương

**Kết quả:** Một số nguyên duy nhất là ước chung lớn nhất của hai số  và

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  15 | 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n, m,uc;  cin>>n>>m;  for(int i=1;i<=min(n,m);i++)  if(n%i==0 and m%i==0) uc=i;  cout<<uc;  return 0;  } |

## **Ước số tốt nhất (bestdiv.\*)**

Kristen yêu thích trò chơi so sánh các con số. Cô ấy định nghĩa rằng với hai số nguyên dương nếu số nào có tổng các chữ số lớn hơn thì số đó tốt hơn, nếu hai số có tổng các chữ số bằng nhau thì số nhỏ hơn sẽ tốt hơn.

**Yêu cầu:**Cho số nguyên dương nn hãy tìm ước số tốt nhất của .

Ví dụ với  thì  có các ước số  trong đó ước số tốt nhất là 6 vì 6 có tổng các chữ số lớn nhất.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Một số nguyên duy nhất là kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12 | 6 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int x,n,s;  int a[100000];  int maxn=0,maxi;  int main()  {  cin>>n;  for(int i=1;i<=n;i++)  {  if(n % i == 0) {  x=i;  s=0;  while (x!=0){  s=s+x%10;  x=x/10;  }  a[i]=s;  }  }  for(int i=1; i<=n;i++)  if (maxn<a[i])  {  maxn=a[i];  maxi=i;  }  cout<<maxi<<endl;  } |

## **Số hoàn hảo (sohh.\*)**

Số hoàn thiện (hay còn gọi là số hoàn chỉnh, số hoàn hảo hoặc số hoàn thành) là một số nguyên dương mà tổng các ước nguyên dương chính thức của nó (số nguyên dương bị nó chia hết ngoại trừ nó) bằng chính nó.

Cho số nguyên dương . Hãy cho biết  có phải là số hoàn thiện hay không?

**Dữ liệu vào:** Số nguyên dương  (

**Kết quả:** Ghi số 1 nếu  là số hoàn thiện, ngược lại ghi 0

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 | 1 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  long long n,s=0;  cin>>n;  for(long long i=1;i\*i<=n;i++) {  if(n%i==0) s=s+i+n/i;  if(i\*i==n) s-=i;  }  if(s-n==n) cout<<1; else cout<<0;  return 0;  } |

## **Tổng và số lượng chữ số (tongslcs.\*)**

Cho số nguyên dương . Hãy tính tổng các chữ số và số lượng các chữ số của .

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** + Dòng đầu tiên ghi tổng các chữ số của

+ Dòng thứ 2 ghi số lượng chữ số của

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 123 | 6  3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  long long n; cin>>n;  int s=0,d=0;  while( n>0 ) {  s=s+n%10;  n=n/10; d++;  }  cout<<s<<endl<<d;  return 0;  } |

## **Số nút (sonut.\*)**

Người ta gọi số Nút của một số nguyên dương là một số nguyên có một chữ số và có giá trị từ 1 đến 9. Để tính số nút của số , người ta tính tổng các chữ số của , nếu kết quả là một số lớn hơn 9 thì tiếp tục lặp lại thao tác tính tổng các chữ số của nó cho đến khi kết quả cuối cùng tìm được là một số có giá trị nhỏ hơn hoặc bằng 9. *Ví dụ:* Số 76 có tổng các chữ số là 13, số 13 lại có tổng các chữ số là 4 cho nên số 76 và số 13 có số Nút là 4.

**Yêu cầu**: Cho trước một nguyên dương . Bạn hãy lập trình tính số nút của số .

**Dữ liệu vào**: Số nguyên .

**Kết quả**: Ghi ra số nút của số .

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 76 | 4 |  | 8 | 8 |

**Chương trình tham khảo**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n;  cin>>n;  while(n>9) {  int a=n, s=0;  while(a>0){ s+=a%10; a/=10;}  n=s;  }  cout<<n;  return 0;  } |

## **Phép cộng không nhớ (cong.\*)**

Bài tập về nhà của Tý trong môn Toán là phép tính có dạng:

Trong đó và đã biết và là số cần tìm. Rất nhanh chóng Tý tìm được . Tuy nhiên Tý lại nghĩ đến kết quả sẽ thế nào nếu phép tính là **phép cộng không nhớ**, nghĩa là với 2 chữ số và thì kết quả của sẽ bị bỏ đi chữ số hàng chục (nếu có).

Ví dụ:

**Yêu cầu:** Với hai số tự nhiên và có cùng số lượng chữ số, hãy tìm sao cho với phép cộng không nhớ.

**Dữ liệu vào:** Hai số và cách nhau bằng 1 dấu cách .

**Kết quả:** Số nguyên .

**Ví dụ:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 612 401 | 13 |  | 896 426 | 212 |  | 111 999 | 0 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int n,m,s=0;;  cin>>n>>m;  while(m>0)  {  s=s\*10+(n%10+m%10)%10;  n=n/10;  m=m/10;  }  int t=0;  while(s>0)  {  t=t\*10+s%10;  s=s/10;  }  cout<<t;  return 0;  } |

## **Phân tích thành thừa số nguyên tố (ptsnt.\*)**

Cho số nguyên dương . Hãy phân tích  thành tích các số nguyên tố.

**Dữ liệu vào:** Số nguyên dương

**Kết quả:**

Dãy gồm  số nguyên tố   sao cho  và

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 100 | 2 2 5 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long n;  int main(){  cin>>n;  int i=2;  while(i<=n) {  while(n%i==0) { cout<<i<<" "; n=n/i; }  i++;  }  return 0;  } |

## **Chia 2, trừ 1 (d2s1.\*)**

Fed có một số nguyên . Nếu là số chẵn thì Fed sẽ thực hiện phép gán ; ngược lại nếu là số lẻ thì Fed sẽ thực hiện phép gán ; Feb sẽ thực hiện các phép gán này cho đến khi thì dừng. Ví dụ với

+ Lần 1: vì chẵn nên gán ;

+ Lần 2: vì lẻ nên gán ;

+ Lần 3: vì chẵn nên gán ;

Như vậy vậy kết quả là sau 3 lần gán thì

Với số nguyên cho trước, hãy cho biết Fed thực hiện bao nhiêu phép gán để đưa về 1?

**Dữ liệu vào:** Số nguyên dương

**Kết quả:** Một số nguyên cho biết kết quả bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 | 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int main(){  long long x,y=0;  cin>>x;  while(x!=1){  if(x%2==0) x=x/2;  else x--;  y++;  }  cout<<y;  return 0;  } |

## **Thu hoạch khoai tây (PTT1.\*)**

Feb có ba ruộng khoai tây, anh ta đã thu hoạch được củ khoai tây ở ruộng thứ nhất và củ khoai tây ở ruộng thứ hai. Anh ta vẫn chưa thu hoạch ở ruộng thứ ba. Feb là người mê tín và anh ta tin rằng nếu tổng số củ khoai tây thu hoạch được ở cả ba ruộng là một số nguyên tố thì anh ta sẽ gặp được nhiều may mắn. Tuy nhiên Feb muốn nghỉ ngơi sớm nên muốn số lượng khoai tây thu hoạch ở ruộng thứ ba là ít nhất.

Hãy cho biết Fed cần thu hoạch thêm ít nhất bao nhiêu củ khoai tây ở ruộng thứ ba để tổng số lượng khoai tây thu hoạch được ở cả ba ruộng là một số nguyên tố.

Lưu ý rằng phải có ít nhất 1 củ khoai tây được thu hoạch ở ruộng thứ ba.

**Dữ liệu vào:** Hai số trên một dòng.

**Kết quả:** Một số nguyên duy nhất là kết quả bài toán.

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 1 3 | 1 |  | 3 4 | 4 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int a,b,t;  cin>>a>>b;  int k=1;  for (int i=1;k!=0;i++)  {  k=0;  t=a+b+i;  for (int j=2;j<=sqrt(t);j++)  if (t%j==0) k++;  if (k==0) cout<<i;  }  return 0;  } |

## **Tính tổng (Total.\*)**

Sau buổi học về số học, Minh đã biết cách tính tổng của số tự nhiên liên tiếp. Ở nhà, Minh tiếp tục làm các bài tập về tính tổng của các số tự nhiên liên tiếp. Minh thắc mắc, liệu với số tự nhiên thì có thể phân tích thành tổng các số tự nhiên liên tiếp hay không?. Ví dụ với (có thể phân tích: 9 = 9; 9 = 5+4; 9 = 2 + 3 + 4) thì có 3 cách phân tích thành tổng các số tự nhiên liên tiếp.

**Yêu cầu:** Viết chương trình giúp Minh tìm số cách phân tích số nhiên thành tổng các số tự nhiên liên tiếp.

**Dữ liệu vào:**

Số nguyên dương .

**Kết quả:**

Một số duy nhất là số cách phân tích tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9 | 3 |

**Hướng dẫn:**

**Cách 1:** Xét hết toàn bộ dãy số liên tiếp từ đến j

Vd:

1,1; 1,2; 1,3; 1,4; 1,5

2,2; 2,3; 2,4; 2,5

3,3; 3,4; 3,5

4,4; 4,5

5,5

Với mỗi tính tổng các số ; nếu thì

Độ phức tạp thuật toán

for i in range(1,k+1):

for j in range(i,k+1):

s= //công thức cấp số cộng

**Cách 2:** Biến đổi toán học.

Giả sử số là tổng của số liên tiếp, trong đó là số nhỏ nhất

Kết quả là một phương trình có 2 ẩn do vậy ta sẽ duyệt để tìm . Vì là số nguyên dương nên

Độ phức tạp:

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int main(){  long long n,res=0;  cin >> n;  long long n1=n\*2;  for(long long i = 1 ;i\*(i+2)/2 < n; i++){  double a;  if((n-(i\*(i+1)/2))%(i+1)==0)  res++;    }  cout<<res+1;  } |

## **Tinh chất ma thuật (tcmt.\*)**

Bạn đang chế tạo một loại thuốc thần kỳ có tác dụng làm khuôn mặt trở nên xinh đẹp. Công thức loại thuộc này là ma thuật và nước. Mỗi thao tác bạn cần thêm 1 ml tính chất ma thuật hoặc 1 ml tinh chất nước. Hãy cho biết bạn ít nhất bao nhiêu thao tác để tạo được loại thuốc thần kỳ.

**Dữ liệu vào:** Số nguyên

**Kết quả:** Số thao tác ít nhất để tạo được thuốc thần kỳ.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 25 | 4 |

**Hướng dẫn:**

Gọi là thao tác để thêm 1 ml ma thuật và là thao tác để thêm 1 ml nước

Cách 1: Xét toàn bộ giá trị của để tìm được nhỏ nhất và có đúng ma thuật, nước

Cách 2: Nhận thấy để đạt ma thuật và nước thì ta cần thao tác nhỏ 1 ml ma thuật và ml thao tác nhỏ 1 ml nước. Yêu cần đề bài cần tìm số thao tác nhỏ nhất nên ta sẽ tối giản phân số

**Chương trình tham khảo:**

|  |  |
| --- | --- |
| **Cách 1:** | **Cách 2** |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int x,mt,i,j;  cin>>x;  if(x==100 or x==0) cout<<1;  else  {  mt=100;  for(i=1; i<100; i++)  for(j=99; j>0;j--)  if(((double)i/j==(double)x/(100-x)) and (j+i<mt))  mt=i+j;  cout<<mt;  }  return 0;  } | #include <bits/stdc++.h>  using namespace std;  int main()  {  int x;  cin>>x;  int y=100-x;  int g=\_\_gcd(x,y);  cout<<x/g+y/g;  return 0;  } |

## **Đồng tiền vàng và tam giác (cointr.\*)**

|  |  |
| --- | --- |
| Tom có đồng tiền vàng, cậu ta thường lấy các đồng tiền vàng này để xếp thành hình tam giác theo quy tắc: Dòng đầu tiên dùng 1 đồng tiền, dòng thứ 2 dùng 2 đồng tiền, dòng thứ 3 dùng 3 đồng tiền, dòng thứ dùng đồng tiền… | Diagram  Description automatically generated |

**Yêu cầu:** Hãy cho biết với đồng tiền vàng, Tom có thể xếp được tam giác nhiều nhất bao nhiêu dòng?

**Dữ liệu vào:** Dòng đầu tiên ghi số nguyên

**Kết quả:** Một số nguyên cho biết kết quả của bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7 | 3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int n,i=1;  cin>>n;  if (n==0) cout<<0;  while (n>=i) {n=n-i;i++;};  cout<<i-1;  return 0;  } |

## **Three Boxs (threebox.\*)**

Chef có 3 hộp với kích thước . Anh ấy cho các hộp vào những chiếc túi có kích thước là . Tìm số túi nhỏ nhất Chef cần để tất cả các hộp đều thuộc một túi nào đó. Một túi có thể chứa nhiều hơn một hộp nếu tổng kích thước của các hộp không vượt quá kích thước của túi.

**Dữ liệu vào:**

+ Dòng đầu tiên chứa số nguyên cho biết số test.

+ dòng tiếp theo, mỗi dòng chứa 4 số nguyên thể hiện kích thước của các hộp và các túi.

**Kết quả:** Với mỗi test in ra một số nguyên cho biết số túi ít nhất mà Chef cần sử dụng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  2 3 5 10  1 2 3 5  3 3 4 4 | 1  2  3 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int t;  cin>>t;  while(t--)  {  int a,b,c,d;  cin>>a>>b>>c>>d;  if(a>b) swap(a,b);  if(b>c) swap(b,c);  if(a>b) swap(a,b);  if(a+b+c<=d) cout<<1; else  if(a+b<=d) cout<<1+c/d+(c%d>0); else  cout<<a/d+b/d+c/d+(a%d>0)+(b%d>0)+(c%d>0);  cout<<endl;  }  return 0;  } |

# **PHẦN IV: MẢNG MỘT CHIỀU**

## **In ngược dãy số (arr01.\*)**

Cho số nguyên dương  và dãy số nguyên

Hãy in dãy số  theo thứ tự ngược lại

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số

**Kết quả:** Dãy số  theo thứ tự ngược lại

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  1 2 3 4 5 6 | 6 5 4 3 2 1 |

**Chương trình tham khảo**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int n,a[10001];  int main(){  cin>>n;  for (int i=1 ; i<=n ; i++) cin>>a[i];  for (int i=n ; i>=1 ; i--) cout<<a[i]<<" ";  return 0;  } |

## **Số lượng và tổng dãy số (arr02.\*)**

Cho số nguyên dương  và dãy số nguyên

Hãy cho biết số lượng và tổng các số trong dãy có giá trị lớn hơn

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số

**Kết quả:**

+ Dòng đầu tiên ghi một số nguyên cho biết số lượng tìm được.

+ Dòng thứ hai ghi một số nguyên cho biết tổng tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  1 3 2 5 6 4 | 2  11 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int n,a[10009],t,dem;  int main(){  cin>>n;  for (int i=1 ; i<=n ; i++) cin>>a[i];  dem=0; t=0;  for (int i=1 ; i<=n-1 ; i++){  if (a[i]>a[n]){  dem++;  t=t+a[i];  }  }  cout<<dem<<endl<<t;  return 0;  } |

## **Dãy số B (seqb.\*)**

Cho số nguyên dương và dãy số nguyên . Hãy tìm dãy số sao cho , hay nói cách khác là tổng của số đầu tiên trong dãy .

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương .

+ Dòng tiếp theo ghi lần lượt các số

**Kết quả:** Ghi lần lượt các số

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  1 4 3 2 | 1 5 8 10 |

|  |  |
| --- | --- |
| **Hướng dẫn** | **Chương trình tham khảo** |
| Sử dụng công thức:  +  + | #include <bits/stdc++.h>  #define ll long long  using namespace std;  ll n,i,a[1000001];  int main()  {  cin>>n;  for(i=1;i<=n;i++) cin>>a[i];  for(i=2;i<=n;i++) a[i]+=a[i-1];  for(i=1;i<=n;i++) cout<<a[i]<<" ";  return 0;  } |

## **Arr\_tbc (arrtbc.\*)**

Cho số nguyên dương  và dãy số nguyên . Hãy tính tổng các số có giá trị lớn hơn trung bình cộng của dãy số.

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số

**Kết quả:**

Một số nguyên duy nhất là tổng của các số có giá trị lớn hơn trung bình cộng của dãy số, trong trường hợp không có số nào lớn hơn trung bình cộng thì in -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 2 3 4 5 | 9 |

**Solution:**

Tính tổng của dãy số

Tính trung bình cộng của dãy số

Duyệt lại dãy số để tính tổng của các số có giá trị lớn hơn

Lưu ý trường hợp in số -1 và giới hạn dữ liệu: kết quả có thể vượt quá

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a[1000010];  int n,d=0;  double x=0;  int main(){  cin >>n;  for (int i=1; i<=n; i=i+1) cin >>a[i];  for (int i=1; i<=n; i=i+1) {  x=x+a[i];  d=d+1;  }  double y=x/d;  long long z=0;  for (int i=1; i<=n; i=i+1)  if (a[i]>y) z=z+a[i];  if (z>y) cout <<z; else cout <<-1;  } |

## **Tìm số K (findk.\*)**

Cho số nguyên dương  và dãy số nguyên dương . Hãy tìm vị trí của  trong dãy số.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi 2 số nguyên dương

+ Dòng thứ 2 ghi  số nguyên dương

+ Dòng thứ 3 ghi số nguyên dương

**Kết quả:** In ra tất cả các vị trí của tìm được trong dãy số, nếu không xuất hiện thì in .

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 6  1 2 3 2 3 1  2 | 2 4 |  | 6  1 2 3 2 3 1  0 | -1 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  #define ll long long  using namespace std;  ll n,a[1000001],i,k,d;  int main() {  cin>>n;  for(i=1;i<=n;i++) cin>>a[i];  cin>>k; d=0;  for(i=1;i<=n;i++)  if(a[i]==k){ d++; cout<<i<<' ';}  if(d==0) cout<<-1;  } |

## **Min max array (MinMaxArray.\*)**

Cho số nguyên dương  và dãy số nguyên .

**Yêu cầu:**Hãy tìm giá trị nhỏ nhất, giá trị lớn nhất, số lượng phần tử có giá trị nhỏ nhất, số lượng phần tử có giá trị lớn nhất trong dãy số đã cho.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi  số nguyên lần lượt là

**Kết quả:**

+ Dòng đầu tiên ghi giá trị nhỏ nhất và số lượng phần tử có giá trị nhỏ nhất.

+ Dòng thứ hai ghi giá trị lớn nhất và số lượng phần tử có giá trị lớn nhất.

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 6  -1 -1 4 0 4 2 | -1 2  4 2 |  | 10  -5 12 12 -2 -5 -2 1 6 3 2 | -5 2  12 2 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a[1000010];  int n;  int main()  {  ios\_base::sync\_with\_stdio(false);  cin.tie(NULL);  cout.tie(NULL);  cin >>n;  int x=0,y=0;  for (int i=1; i<=n; i=i+1) cin >>a[i];  int minx=a[1];  for (int i=1; i<=n; i=i+1)  if (a[i]<minx) minx=a[i];  for (int i=1; i<=n; i=i+1)  if (a[i]==minx) x=x+1;    int maxx=a[1];  for (int i=1; i<=n; i=i+1)  if (a[i]>maxx) maxx=a[i];  for (int i=1; i<=n; i=i+1)  if (a[i]==maxx) y=y+1;  cout <<minx<<" "<<x<<endl<<maxx<<" "<<y;  } |

## **Vòng tròn số (vongso.\*)**

Một dãy số gồm số nguyên được đánh số theo thứ tự từ 1 đến và được xếp thành một vòng tròn theo chiều kim đồng hồ.

**Yêu cầu:** Hãy tìm tổng lớn nhất của số liên tiếp nhau trong vòng tròn trên.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi hai số nguyên và cách nhau một dấu cách.

+ Dòng thứ hai ghi số nguyên trong dãy, mỗi số có giá trị tuyệt đối không vượt quá 1000. Giữa các số được ghi cách nhau một dấu cách.

**Kết quả:**

Ghi một số nguyên duy nhất là tổng lớn nhất của số liên tiếp nhau tìm được trong vòng tròn số.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3  10 2 3 5 7 | 22 |

**Hướng dẫn:**

+ Phía sau số thứ là số thứ 1 nên ta có thể gấp đôi dãy số bằng cách đặt với

+ Sau khi gấp đôi mảng ta xây dựng mảng với ý nghĩa là tổng của số đầu tiên trong

+ Để tính được tổng số liên tiếp trong bắt đầu từ vị trí thứ ta sử dụng công thức

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n,k;  int a[200011];  int main()  {  cin>>n>>k;  for(int i=1;i<=n;i++) cin>>a[i];  for(int i=1;i<=n;i++) a[i+n]=a[i];  n=n\*2;  int ans=0;  for(int i=1;i<=k;i++) ans+=a[i];  int s=ans;  for(int i=2;i<=n-k+1;i++)  {  s=s-a[i-1]+a[i+k-1];  if(ans<s) ans=s;  }  cout<<ans;  return 0;  } |

## **Tổng lớn nhất hai số (sum2n.\*)**

Cho số nguyên dương  và dãy số nguyên

Hãy tìm trong dãy hai số sao cho tổng của chúng là lớn nhất.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số

**Kết quả:** Ghi một số nguyên cho biết tổng lớn nhất tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  4 1 2 3 -1 5 | 9 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int a[1000011];  int main(){  int n,x=0,m;cin >>n;  for(int i=0;i<n;i++) cin >>a[i];  m=a[0];  for(int i=0;i<n;i++) if(a[i]>m){m=a[i];x=i;};swap(a[0],a[x]);  m=a[1]; x=1;  for(int i=1;i<n;i++) if(a[i]>m){m=a[i];x=i;};swap(a[1],a[x]);  cout <<a[0]+a[1];  return 0;  } |

## **Cùng dấu (cungdau.\*)**

Viết chương trình nhập số nguyên dương và dãy số nguyên .

Hãy in cặp số liên tiếp đầu tiên cùng dấu nhau (hai số cùng là số âm hoặc cùng là số dương). Dữ liệu nhập vào luôn đảm bảo có kết quả.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương .

+ Dòng thứ hai ghi lần lượt các số nguyên

**Kết quả:** Ghi hai số cho biết kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 9  3 0 2 -1 ***5 6*** 2 -4 -5 | 5 6 |

**Hướng dẫn:**

Duyệt tất cả các cặp số gần nhau nếu tích thì in ra mà hình giá trị và rồi ***kết thúc*** chương trình bằng cách dùng

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n, a[1010];  int main() {  cin >>n;  for (int i=1; i<=n; i=i+1) cin >>a[i];  for (int i=1; i<=n; i=i+1)  {  if ((a[i]>0 and a[i+1]>0) or (a[i]<0 and a[i+1]<0))  {cout <<a[i]<<" "<<a[i+1]; return 0; }  }  } |

## **Cực đại địa phương (cddp.\*)**

Viết chương trình nhập số nguyên dương và dãy số nguyên . Hãy tính số lượng các số cực đại địa phương. Một số được gọi là số cực đại địa phương nếu và .

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương .

+ Dòng thứ hai ghi lần lượt các số

**Kết quả:** Một số nguyên duy nhất cho biết kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10  4 **5** 2 **3** 1 5 **7** 3 **5** 2 | 4 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a[1000000];  int n,d=0;  int main(){  cin>>n;  for(int i=1; i<=n; i++) cin>>a[i];  for (int i=2; i<n; i++) {  if( a[i]>a[i-1] and a[i] > a[i+1]) d++;  }  cout<<d;  } |

## **Tổng của đoạn số (SumP.\*)**

Cho số nguyên dương và dãy số nguyên . Có câu hỏi, mỗi câu hỏi được cho bởi hai số nguyên , với mỗi câu hỏi hãy cho biết tổng các số từ vị trí đến vị trí trong dãy là bao nhiêu?

**Dữ liệu vào:**

+ Dòng đầu tiên ghi hai số nguyên dương .

+ Dòng thứ hai ghi lần lượt các số

+ dòng tiếp theo mỗi dòng ghi một cặp số nguyên

**Kết quả:** Ghi trên dòng, mỗi dòng là tổng các số từ đến trong dãy tương ứng với thứ tự trong input

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 2  1 4 2 8  1 3  1 4 | 7  15 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n,q;  int a[100011];  long long b[100011];  int main() {  cin>>n>>q;  for(int i=1;i<=n;i++) cin>>a[i];  b[0]=0;  for(int i=1;i<=n;i++) b[i]=b[i-1]+a[i];  for(int i=1;i<=q;i++) {  int x1, x2;  cin>>x1>>x2;  cout<<b[x2]-b[x1-1]<<endl;  }  return 0;  } |

## **Bộ 3 số (b3so.\*)**

Cho số nguyên dương và dãy số nguyên . Hãy cho biết trong dãy số có bao nhiêu bộ 3 số liên tiếp nhau mà thương của tích 3 số đó với tổng của chúng đúng bằng . Hay nói cách khác, có bao nhiêu bộ 3 số mà ; với

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số nguyên

**Kết quả:** Một số nguyên duy nhất cho biết kết quả bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 1  1 2 3 4 | 1 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long a[1000000+11];  int main() {  int n,k;  cin >> n >> k; int d=0;  for (int i=1;i<=n;i++) cin >> a[i];  for (int i=1;i<=n-2;i++) {  long long x=a[i]\*a[i+1]\*a[i+2];  long long y=a[i]+a[i+1]+a[i+2];  if (y\*k==x) d++;  }  cout << d;  } |

## **Cặp số có tích lớn nhất (max2num.\*)**

Cho dãy gồm số nguyên , hãy tìm cặp số sao cho tích của chúng là lớn nhất.

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương

+ Dòng tiếp theo ghi lần lượt các số

**Kết quả:** Một số nguyên duy nhất là tích lớn nhất tìm được

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8  3 4 1 2 3 9 1 3 | 36 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a[10000001], n;  int main()  {  cin>>n;  for(int i=1;i<=n;i++) cin>>a[i];  for(int i=1;i<=2;i++)  for(int j=i+1;j<=n;j++)  if(a[i]>a[j]) swap(a[i],a[j]);  for(int i=n;i>=n-1;i--)  for(int j=2;j<i;j++)  if(a[j]>a[i]) swap(a[i],a[j]);  long long m1=(long long)a[1]\*a[2];  long long m2=(long long)a[n]\*a[n-1];  cout<<max(m1,m2);  return 0;  } |

## **Array Manipulation (ArrM.\*)**

Cho dãy số nguyên ban đầu có giá trị tất cả các phần tử bằng 0. Cho một dãy gồm truy vấn, mỗi truy vấn có dạng với ý nghĩa tăng giá trị các phần tử có vị trí từ đến lên đơn vị.

Hãy cho biết giá trị lớn nhất của dãy sau khi thực hiện lần lượt truy vấn.

**Dữ liệu vào:**

+ Dòng đầu ghi 2 số nguyên

+ dòng tiếp theo, mỗi dòng ghi 3 số nguyên lần lượt là )

**Kết quả:** Một số nguyên là giá trị lớn nhất của dãy số sau khi thực hiện truy vấn.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 3  1 2 100  2 5 100  3 4 100 | 200 |

**Hướng dẫn:**

Với mỗi truy vấn tương đương với:

+ Tăng giá trị tất cả các phần tử từ đến lên đơn vị

+ Giảm giá trị tất cả các phần tử từ đến xuống đơn vị

Từ đó với mỗi truy vấn ta xử lý như sau:

Sau đó thực hiện cộng dồn:

Cuối cùng là tìm giá trị lớn nhất trong dãy số.

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  long long x[10000000+3];  int main(){  int n,q,a,b,k,i; cin >>n>>q;  for(i=0;i<=n;i++) x[i]=0;  while(q--){  cin >>a>>b>>k;  x[a-1]+=k; x[b]-=k;  }  for(i=1;i<n;i++) x[i]=x[i-1]+x[i];  long long m=x[0];  for(i=0;i<n;i++) m=max(m,x[i]);  cout <<m;  return 0;  } |

## **Ăn kẹo (ecandy.\*)**

Po có hộp kẹo, hộp thứ có viên kẹo bên trong.

Po muốn số lượng kẹo trong hộp kẹo bằng nhau nên cậu ta sẽ ăn một số kẹo trong các hộp.

**Yêu cầu:** Hãy cho biết Po cần ăn ít nhất bao nhiêu kẹo để số kẹo trong các hộp bằng nhau.

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+ Dòng thứ hai ghi lần lượt các số nguyên dương

**Kết quả:** Một số nguyên duy nhất là kết quả của bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 2 3 4 5 | 10 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n;  int a[1000011];  int main()  {  ios\_base::sync\_with\_stdio(false);  cin.tie(NULL);  int minx=1e9+1;  cin>>n;  for(int i=1;i<=n;i++) { cin>>a[i]; minx=min(minx,a[i]);}  long long ans=0;  for(int i=1;i<=n;i++) ans+=(a[i]-minx);  cout<<ans<<endl;  return 0;  } |

## **Fast Food (fafo.\*)**

Po là chủ một cửa hàng bán đồ ăn nhanh, anh ta có kế hoạch chuyển cửa hàng sang một khu phố khác để kinh doanh. Nhờ vào chương trình trí tuệ nhân tạo do chính Po xây dựng, Po có thể dự đoán được doanh thu của ngày ngày tiếp theo nếu cửa hàng vẫn ở vị trí cũ là và nếu cửa hàng ở khu phố mới là .

**Yêu cầu:** Hãy cho biết Po nên chuyển cửa hàng qua khu mới vào ngày thứ mấy để doanh thu trong ngày tiếp theo là lớn nhất.

**Dữ liệu vào:**

+ Dòng đầu chứa hai số nguyên dương ;

+ Dòng thứ hai chứa 𝑛 số nguyên dương ;

+ Dòng thứ hai chứa 𝑛 số nguyên dương ;

**Kết quả:** Một số nguyên duy nhất cho biết doanh thu lớn nhất của Po trong ngày tiếp theo.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  1 10 2 6  7 3 20 1 | 32 |

**Hướng dẫn:**

Yêu cầu của bài là tìm một vị trí sao cho sao cho đạt giá trị lớn nhất.

Do vậy để làm bài này ta chỉ cần xây dựng mảng và với ý nghĩa

+ là tổng của phần tử đầu tiên của mảng

+ là tổng của các phần tử

Việc xây dựng mảng được thực hiện bằng cách cộng dồn.

Kết quả là

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long a[500011],b[500011];  int main() {  long long ans=0;  int n; cin>>n;  for(int i=1;i<=n;i++) cin>>a[i];  for(int i=1;i<=n;i++) cin>>b[i];  b[n+1]=0; a[0]=0;  for(int i=n;i>=1;i--) b[i]=b[i+1]+b[i];  for(int i=2;i<=n;i++) a[i]=a[i-1]+a[i];  for(int i=1;i<=n+1;i++) ans=max(a[i-1]+b[i],ans);  cout<<ans<<endl;  return 0;  } |

## **Số 10 (number10.\*)**

Bạn yêu thích nhất số nào? Riêng Nam thích nhất số 10 vì nó gồm 2 chữ số là 1 và 0, nó còn đại diện cho 2 bit 1 và 0 trong hệ nhị phân. Nam cũng thích luôn những số chứa số 10 bên trong. Ví dụ 19102, 18110110, 129392101,…là những số Nam thích, còn những số 1920, 39201,…Nam sẽ không thích vì không có số 10 ở trong.

**Yêu cầu:**Số nguyên dương  và các số nguyên dương ; hãy cho biết những số Nam thích ở trong dãy số.

**Dữ liệu vào**

+ Dòng đầu tiên ghi số nguyên dương  .

+ Dòng thứ 2 ghi lần lượt các số  . Các số cách nhau ít nhất 1 dấu cách

**Kết quả:** Ghi các số Nam thích theo thứ tự nhập vào, mỗi số ghi trên một dòng. Nếu tất cả các số trong dãy Nam đều không thích thì ghi .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  1021 3019 7821 10 | 1021  10 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int n, s = 0; long long a[1000001];  long long tg;  int main()  {  ios\_base::sync\_with\_stdio(false);  cin.tie(NULL);  cin>>n;  for (int i = 1; i <=n; i++) cin>>a[i];  for (int i = 1; i <=n; i++)  { int ok = 0;  tg = a[i];  while (tg > 0)  {  if (tg % 100 == 10) ok = 1;  tg = tg/10;  }  if (ok == 1) {cout<<a[i]<<endl; s++;}  }  if (s == 0) cout<<-1;  return 0;  } |

# **PHẦN V: STRING**

## **So sánh tên hai người (str01.\*)**

Nhập họ tên của hai người vào hai biến xâu rồi đưa ra màn hình xâu dài hơn, nếu bằng nhau thì đưa ra xâu nhập sau.

**Dữ liệu vào:**

+ Dòng đầu tiên cho biết họ tên của người thứ nhất

+ Dòng thứ hai cho biết họ tên của người thứ hai

**Kết quả:** Một xâu là kết quả của bài toán

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Nguyen Nam  Le Minh | Nguyen Nam |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  string s,t;  getline(cin,s);  getline(cin,t);  if(s.length()>t.length()) cout<<s;  else cout<<t;  return 0;  } |

## **Ký tự đầu tiên và cuối cùng (str02.\*)**

Viết chương trình nhập vào hai xâu  và .

**Yêu cầu:** Hãy cho biết:

+ Ký tự đầu tiên của xâu ?

+ Ký tự cuối cùng của xâu ?

+ Ký tự đầu tiên của xâu  có giống ký tự cuối cùng của xâu  hay không?

***Mỗi xâu có không quá  ký tự***

**Dữ liệu vào:**

+ Dòng đầu ghi xâu

+ Dòng thứ hai ghi xâu

**Kết quả:** Ghi lần lượt  trong đó  là ký tự đầu tiên của xâu ;  là ký tự cuối cùng của xâu ;  nếu  ngược lại .

**Ví dụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| tinhoc  coban | t n 0 |  | abcdef  fudhska | a a 1 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  string s,t;  getline(cin,s);  getline(cin,t);  cout<<s[0]<<" ";  cout<<t[t.length()-1]<<" ";  cout<<(s[0]==t[t.length()-1]);  return 0;  } |

## **Xâu đảo ngược (str03.\*)**

Cho xâu

**Yêu cầu:** Tạo xâu  là xâu đảo ngược của xâu

**Dữ liệu vào**: Xâu

**Kết quả:** Xâu

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| abcd | dcba |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  string s;  getline(cin,s);  string t="";  for(int i=0;i<s.length();i++) t=s[i]+t;  cout<<t;  return 0;  } |

## **Xóa ký tự trắng trong xâu (str04.\*)**

Cho xâu

**Yêu cầu:** Hãy xóa hết các ký tự trắng (dấu cách) trong xâu

**Dữ liệu vào:**

+ Xâu

**Kết quả:** Xâu  sau khi xóa bỏ hết các ký tự trắng

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ab ac a | abaca |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  string s;  getline(cin,s);  while((int)s.find(" ")!=-1)  s.erase(s.find(" "),1);  cout<<s;  return 0;  } |

## **Xóa ký tự trắng dư thừa (str05.\*)**

Cho xâu

**Yêu cầu:** Hãy xóa các ký trắng trong xâu  để xâu  thỏa mãn điều kiện:

+ Không có ký tự trắng đầu xâu

+ Không có ký tự trắng cuối xâu

+ Không có hai ký tự trắng liên tiếp nhau

**Dữ liệu vào:** Xâu

**Kết quả:** Xâu  sau khi thực hiện yêu cầu đề bài

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ab ac bc | ab ac bc |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  string s;  getline(cin,s);  while(s[0]==' ') s.erase(0,1);  while(s[s.length()-1]==' ') s.erase(s.length()-1,1);  while((int)s.find(" ")>-1) s.erase(s.find(" "),1);  cout<<s;  } |

## **Xóa chữ số trong xâu (str06.\*)**

Cho xâu

**Yêu cầu:** Hãy xóa hết tất cả các ký tự chữ số trong xâu

**Dữ liệu vào:** Xâu

**Kết quả:** Xâu  sau khi đã xóa hết tất cả các ký tự chữ số

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| ab12cd34ef566 | abcded |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  string s;  getline(cin,s);  for(char c='0';c<='9';c++)    while((int)s.find(c)!=-1)  s.erase(s.find(c),1);  cout<<s;  } |

## **Số từ trong xâu (str07.\*)**

Trong xâu, ***từ*** là một dãy các ký tự liên tiếp nhau không chứa ký tự trắng, hai từ liên tiếp nhau được ngăn cách nhau bằng ký tự trắng.

Ví dụ xâu  có 03 từ là ,  và .

**Yêu cầu:** Hãy cho biết số từ trong .

**Dữ liệu vào:** Xâu

**Kết quả:** Một số nguyên duy nhất cho biết số từ trong xâu

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| abc a ab ab ac | 5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  string s;  int m=1;  int main(){  getline (cin,s);  if (s.length()==0) cout<<0;  else  {  for (int i=0; i<=s.length()-2; i++){  if ((s[i]==' ') and (s[i+1]!=' ')) m+=1;  }    if (s[0]==' ') cout<<m-1;  else cout<<m;  }  } |

## **Xâu đối xứng (str08.\*)**

Một xâu được gọi là đối xứng khi viết từ trái sang phải có giá trị giống như viết từ phải sang trái. Các xâu sau là các xâu đối xứng:

Các xâu sau không phải là xâu đối xứng:

Cho số nguyên dương nn và các xâu

**Yêu cầu:** Hãy cho biết xâu sisi có phải là xâu đối xứng hay không?

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên dương

+  dòng tiếp theo, dòng thứ  cho biết xâu

**Kết quả:**

+ Ghi  số, trong đó số thư  có giá trị  nếu xâu  đối xứng, ngược lại ghi

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  abcba  abcab  a | 101 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int main()  {  int n;  cin>>n;  for (int k=0;k <n; k++)  {  string s;  cin>>s;  int x = 1;  for(int i = 0;i<(int)s.length()/2;i++)  {  if(s[i]!=s[(int)s.length()-1-i]) x=0;  }  cout<<x;  }  return 0;  } |

## **Số lượng ký tự (str09.\*)**

Cho xâu  chỉ bao gồm các chữ cái in HOA, in thường, ký tự chữ số và dấu cách.

Hãy cho biết trong  có bao nhiêu chữ cái in HOA, in thường, chữ số

**Dữ liệu vào:** Xâu

**Kết quả:**

+ Dòng đầu ghi số lượng chữ cái in HOA trong xâu .

+ Dòng thứ hai ghi số lượng chữ cái in thường trong xâu .

+ Dòng thứ 3 ghi số lượng chữ số trong xâu .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| C0 b4n v3 X4u tr0ng Cpp | 3  10  5 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  string s;  getline(cin,s);  int du=0, dn=0, dl=0;  for(int i=0;i<s.length();i++)  {  if(s[i]<='z' and s[i]>='a') dl++;  if(s[i]<='Z' and s[i]>='A') du++;  if(s[i]<='9' and s[i]>='0') dn++;  }  cout<<du<<endl<<dl<<endl<<dn<<endl;  return 0;  } |

## **In hoa in thường (str10.\*)**

Cho xâu  chỉ bao gồm các chữ cái tiếng Anh in hoa, in thường và dấu cách.

Hãy chuẩn hóa xâu theo quy tắc: Chữ cái đầu tiên của mỗi ***từ*** trở thành in HOA, các chữ cái khác trở thành in thường

Biết rằng: trong  không có 2 ký tự trắng liên tiếp nhau, không có ký tự trắng đầu xâu

**Dữ liệu vào:** Xâu

**Kết quả:** Xâu  sau khi đã chuẩn hóa.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| lUu hAI Phong | Luu Hai Phong |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  string s; getline(cin,s);  if(s[0]<='z' and s[0]>='a') s[0]=(char)((int)s[0]-32);  for(int i=0;i<s.length();i++) {  if(s[i]==' ' and s[i+1]<='z' and s[i+1]>='a')  s[i+1]= (char)((int)s[i+1]-32);  if(s[i]!=' ' and s[i+1]<='Z' and s[i+1]>='A')  s[i+1]= (char)((int)s[i+1]+32);  }  cout<<s;  return 0;  } |

## **Thay thế chuỗi (str11.\*)**

Cho 3 xâu ,  và .

Hãy thay thế tất cả các xâu  trong xâu  bằng xâu .

**Dữ liệu vào:**

+ Dòng đầu tiên ghi xâu

+ Dòng thứ hai ghi xâu

+ Dòng thứ ba ghi xâu

***Các ký tự trong xâu  không xuất hiện trong xâu  và ngược lại***

**Kết quả:** Xâu  sau khi thay thế.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| abcxyhxyabxy  xy  12 | abc12h12ab12 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  string x,s,y;  getline(cin,s); getline(cin,x); getline(cin,y);  while ((int)s.find(x)!=-1) {  int vt=(int)s.find(x);  s.erase(vt,x.length());  s.insert(vt,y);  }  cout<<s;  return 0;  } |

## **So sánh hai xâu (str12.\*)**

Tí có hai xâu lần lượt là  và . Hai xâu có cùng độ dài, bao gồm các ký tự tiếng Anh in thường và ký tự

Tí tự hỏi có thể thay thế các ký tự  trong cả hai xâu  và  để hai xâu bằng nhau hay không?

Hãy giúp Tí trả lời câu hỏi đó.

**Dữ liệu vào:**

+ Dòng đầu tiên là số nguyên   cho biết số lượng bộ dữ liệu

+ Cứ mỗi 2 dòng tiếp theo:

- Dòng đầu cho biết giá trị xâu

- Dòng thứ hai cho biết giá trị xâu

**Kết quả:**

+ Với mỗi bộ dữ liệu ghi kết quả trên một dòng. Dòng thứ  ghi  nếu bộ dữ liệu thứ  trong input có thể thay thế dấu  để  bằng ; ngược lại ghi .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  s?or?  sco??  stor?  sco?? | 1  0 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int i=0, n;  cin>>n;  while(n--){  string s,t;  cin>>s; cin>>t;  int a=s.length();  for(i=0;i<a;i++)  if((s[i]!=63) and (t[i]!=63) and (s[i]!=t[i]))  {  cout<<0<<endl;  i=a+1;  }  if(i==a) cout<<1<<endl;  }  return 0;  } |

## **Xâu con (substr.\*)**

Cho xâu , hãy tìm xâu dài nhất thỏa mãn các điều kiện:

+ là xâu con của

+ Không có xâu con khác rỗng của là tiền tố của .

+ Không có xâu con khác rỗng của là hậu tố của .

**Dữ liệu:**

+ Dòng đầu ghi số nguyên dương cho biết số lượng testcase. dòng tiếp theo mỗi dòng ghi một xâu .

Tổng độ dài của trong tất cả testcase không quá

Trong xâu chỉ chứa các ký tự chữ cái latinh in thường.

**Kết quả:** Với mỗi testcase in ra độ dài xâu tìm được, nếu không tìm được xâu thì in .

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  abcdab  aaa | 2  -1 |

**Solution:**

Nhận xét: Để thỏa mãn các điều kiện của bài toán thì xâu không được chứa ký tự và .

Từ đó suy ra ta chỉ cần tìm đoạn con dài nhất không chứa và

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int t; cin>>t;  while(t--)  {  string s;  cin>>s;  int ans=0,vt=0;  for(int i=1;i<s.length();i++)  if(s[i]==s[0] or s[i]==s[s.length()-1])  {  ans=max(ans,i-vt-1);  vt=i;  }  if(ans==0) cout<<-1<<endl; else cout<<ans<<endl;  }  return 0;  } |

## **Super digit (superdigit.\*)**

Ta định nghĩa của một số nguyên là:

1. nếu chỉ có một chữ số

2. của tổng các chữ số của nếu có nhiều hơn một chữ số

Ví dụ của

Bạn được cho 2 số nguyên và , gọi là số nguyên được tạo thành bằng cách ghép lần liên tiếp số với nhau. Ví dụ và thì

**Yêu cầu:** Hãy tìm của số .

**Dữ liệu vào:** hai số nguyên dương

**Kết quả:** một số nguyên duy nhất là giá trị của số .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 148 3 | 3 |

**Solution:**

Để ý rằng rất lớn nên không thể sử dụng các kiểu dữ liệu số nguyên để lưu trữ, tuy nhiên trong bài toán này chỉ cần làm việc với tổng các chữ số. Với có chữ số thì giá trị lớn nhất của tổng các chữ số là . Do vậy chỉ cần khai báo biến có kiểu số nguyên để lưu giá trị tổng. Việc đọc giá trị được thực hiện bằng cách đọc từ ký tự chữ số hoặc dùng .

Sau khi tính được tổng các chữ số của , cần thực hiện tăng lần giá trị đó. Sau đó tính giá trị như hướng dẫn trong đề bài.

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  string n;  long long k,s=0;  cin>>n>>k;  for (long long i=0;i<n.length();i++) s+=n[i]-48;  s=s%9;  k=k%9;  if ((s\*k)%9==0) cout<<'9';  else cout<<(k\*s)%9;  return 0;  } |

## **Khoảng cách giữa hai số (kc2cs.\*)**

Với hai chữ số và , khoảng cách của chúng được định nghĩa là số nguyên không âm nhỏ nhất mà khi cộng thêm vào một chữ số nào đó trong hai chữ số thì kết quả nhận được là một số nguyên có chữ số hàng đơn vị trùng với chữ số còn lại. Ví dụ: vìvì, còn vì .

Với hai số nguyên dương và có cùng số lượng chữ số, khoảng cách giữa hai số và là tổng khoảng cách giữa các cặp chữ số cùng hàng tương ứng.

Ví dụ .

**Bài toán:** Cho hai số và có cùng lượng chữ số , hãy tìm khoảng cách .

**Dữ liệu vào:** Dòng đầu chứa số ; dòng thứ hai chứa số thỏa mãn dàng buộc của bài toán.

**Kết quả:** Ghi một số nguyên duy nhất là kết quả tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 213  419 | 6 |

**Hướng dẫn:**

Ta xây dựng hàm tính khoảng cách hai chữ số và theo định nghĩa của bài toán, sau đó dùng hàm này để tính tổng khoảng cách giữa các cặp chữ số của hai số.

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int dis(int x,int y)  {  if (x<y) swap(x,y);  return min(y+10-x,x-y);  }  int main()  {  string x,y;  cin>>x;  cin>>y;  int sum=0;  for(int i=0;i<x.size();i++)  sum+=dis(x[i]-'0',y[i]-'0');  cout<<sum<<endl;  } |

# **PHẦN V: TOÁN**

## **Restaurant (rest.\*)**

Martha đang phỏng vấn xin vào làm việc tại một công ty. Câu hỏi đầu tiên cô ấy phải giải quyết là: Cho một chiếc bánh mì hình chữ nhật kích thước , hãy cắt bánh mì thành những miếng bánh hình vuông giống nhau có cạnh lớn nhất sao cho chiếc bánh ban đầu không có miếng thừa.

**Dữ liệu vào:** Gồm nhiều bộ dữ liệu:

+ Dòng đầu ghi số nguyên   cho biết số lượng bộ dữ liệu.

+  dòng tiếp theo mỗi dòng ghi hai số nguyên  và  cho biết kích thước của một chiếc bánh mì .

**Kết quả:**

+ Với mỗi bộ dữ liệu tương ứng trong input đưa ra số lượng miếng bánh cắt được thỏa mãn yêu cầu bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2 2  6 9 | 1  6 |

**Chương trình tham khảo:**

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  int t,l,b,v;  int main()  {  cin>>t;  for(int i=1;i<=t;i++)  {  cin>>l>>b;  v=\_\_gcd(l,b);  cout<<l/v\*b/v<<endl;  }  return 0;  } |

## **Summing the series (sns.\*)**

Phần tử thứ  trong dãy được tính theo công thức: .

**Yêu cầu:**Hãy tính  Trong đó  là phép chia lấy phần dư

**Dữ liệu vào:** Gồm nhiều bộ dữ liệu:

+ Dòng đầu tiên ghi số nguyên dương   cho biết số lượng bộ dữ liệu

+  dòng tiếp theo mỗi dòng ghi một số nguyên dương

**Kết quả:** Với mỗi bộ dữ liệu trong Input, tính và ghi ra giá trị  tương ứng, mỗi kết quả ghi trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  2  1 | 4  1 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int t;  int main()  {  int MOD = 1e9+7;  cin>>t;  for(int k=1; k<=t; k++)  {  long long n;  cin>>n;  cout<<(n%MOD)\*(n%MOD)%MOD<<endl;  }  return 0;  } |

## **Tổng của đoạn thứ k (sumofk.\*)**

Cho dãy số nguyên dương lẻ: 1, 3, 5, 7, 9, 11, 13, 15, 17,….Thực hiện gộp nhóm với dãy số theo quy tắc: nhóm thứ nhất là {1}; nhóm thứ hai là {3, 5}; nhóm thứ 3 là {7, 9, 11},…cứ như vậy nhóm thứ gồm số tiếp theo trong dãy số.

***Yêu cầu:*** Cho số nguyên dương , hãy tính tổng các số trong nhóm thứ .

**Dữ liệu vào:** Số nguyên dương ;

**Kết quả:** Ghi một số nguyên cho biết kết quả của bài toán.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | 27 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  long long k;  cin>>k;  cout<<k\*k\*k;  } |

## **Connecting towns (coto.\*)**

Có  thành phố, các thành phố được đánh số từ 0 đến , giữa thành phố thứ  đến thành phố thứ  có   con đường khác nhau. Hãy cho biết có bao nhiêu cách đi từ thành phố 0 đến thành phố . Số lượng cách đi có thể rất lớn nên kết quả sẽ được lấy dư cho .

**Dữ liệu vào:**

Dòng đầu tiên ghi số nguyên   cho biết số lượng bộ dữ liệu, mỗi bộ dữ liệu có cấu trúc:

+ Dòng đầu là số nguyên dương   cho biết số lượng thành phố

+ Dòng tiếp theo ghi  số nguyên dương, trong đó số thứ   ghi giá trị  .

**Kết quả:** Mỗi bộ dữ liệu trong Input ghi một số nguyên trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3  1 3  4  2 2 2 | 3  8 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int t;  cin>>t;  while(t--)  {  int n;  cin>>n;  long long ans=1;  for(int i=1;i<n;i++)  {  int x;  cin>>x;  ans=(ans\*x)%1234567;  }  cout<<ans<<endl;  }  } |

## **Tam giác vuông (tgvuong.\*)**

Một xưởng cơ khí cần hàn các đầu mút của ba thanh thép thẳng lại với nhau để tạo thành một tam giác vuông. Hiện tại xưởng có thanh thép nêu trên đã được đánh số thứ tự từ 1 đến , thanh thứ có độ dài ( là số nguyên dương).

**Yêu cầu**: Bạn hãy giúp xưởng cơ khí tìm xem có bao nhiêu cách chọn ba thanh thép trong thanh thép trên sao cho trong mỗi cách chọn đều hàn được thành một tam giác vuông (không được làm thay đổi độ dài ban đầu của các thanh thép).

**Dữ liệu vào**:

+ Dòng đầu ghi số nguyên dương ;

+ Dòng thứ hai ghi số nguyên dương là độ dài của các thanh thép, các số viết cách nhau một dấu cách.

**Kết quả**: Ghi ra trên mỗi dòng gồm ba số nguyên cách nhau một dấu cách là số thứ tự của ba thanh thép thoả yêu cầu bài toán. Nếu không có cách chọn nào thoả yêu cầu thì ghi số -1.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  5 4 3 6 5 | 1 2 3  2 3 5 |

**Hướng dẫn:**

Sử dụng ba vòng lặp lồng nhau để xét hết tất cả các bộ ba ; với mỗi bộ ba cần kiểm tra xem nó có tạo thành tam giác vuông hay không?

Cần có biến đếm số lượng tam giác vuông tạo thành để ghi kết quả -1.

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main()  {  int a[101], n, d=0;;  cin>>n;  for(int i=1;i<=n;i++) cin>>a[i];  for(int i=1;i<n-2;i++)  for(int j=i+1;j<n;j++)  for(int k=j+1;k<=n;k++)  {  int ok1=(a[i]+a[j]>a[k] and a[i]+a[k]>a[j] and a[j]+a[k]> a[i]);  int ok2=(a[i]\*a[i]+a[j]\*a[j] == a[k]\*a[k] );  ok2=ok2 or (a[i]\*a[i]+a[k]\*a[k] == a[j]\*a[j] );  ok2=ok2 or (a[j]\*a[j]+a[k]\*a[k] == a[i]\*a[i] );  if(ok1 and ok2) { cout<<i<<" "<<j<<" "<<k<<endl; d++; }  }  if(d==0) cout<<-1;  return 0;  } |

## **Số nguyên tố đặc biệt (sntdb.\*)**

Hải là người yêu thích các số nguyên tố chính vì vậy cậu ta thường tìm ra những số nguyên tố có tính chất đặc biệt. Hải đã phát hiện ra có những số nguyên tố mà tổng các chữ số của nó cũng là số nguyên tố. Ví dụ: số 67 có tổng hai chữ số của nó bằng 13 cũng là một số nguyên tố. Hải gọi những số nguyên tố như vậy là số nguyên tố đặc biệt.

**Yêu cầu:** Cho hai số nguyên hãy cho biết trong đoạn từ đến có những số nguyên tố đặc biệt nào?

**Dữ liệu vào:** Hai số nguyên dương trên một dòng và cách nhau một dấu cách. Dữ liệu vào luôn đảm bảo có bài toán có nghiệm.

**Kết quả:** Ghi các số nguyên tố đặc biệt từ đến . Các số in ra theo thứ tự tăng dần và cách nhau một dấu cách.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 50 | 11 23 29 41 43 47 |

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int n,l,r,s,p;  int snt[10000011];  int main()  {  cin>>l>>r;  int Nmax=10000011;  for(int i=1;i<=Nmax;i++) snt[i]=0;  snt[1]=2;  for(int i=2;i\*i<=Nmax;i++)  if(snt[i]==0)  for(int j=2;j\*i<=Nmax;j++) snt[i\*j]=1;  for(int n=l; n<=r; n++){  s=0; p=n;  while(p!=0) { s=s+p%10; p=p/10; }  if(snt[n]==0 and snt[s]==0) cout<<n<<" ";  }  return 0;  } |

## **Kiểm tra số fibonacci (isfibo.\*)**

Một số phần tử đầu tiên của dãy số Fibonacci là Trừ hai số đầu tiên, các số trong dãy Fibonacci bằng tổng của hai số liền trước nó. Công thức của dãy số Fibonacci:

Bạn được cho một số nguyên , hãy cho biết có phải là số Fibonacci hay không?

**Dữ liệu vào:**

+ Dòng đầu tiên ghi số nguyên cho biết số lương testcase

+ dòng tiếp theo mỗi dòng ghi một số nguyên

**Kết quả:** Mỗi testcase ghi kết quả trên một dòng, tương ứng nếu là số Fibonacci thì ghi “IsFibo” ngược lại ghi “IsNotFibo”

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  5  7  8 | IsFibo  IsNotFibo  IsFibo |

**Hướng dẫn:**

+ Đầu tiên ta xây dựng mảng chứa tất cả các số Fibonacci trong đoạn từ 0 đến

+ Với mỗi số thực hiện tìm kiếm trong mảng , nếu tìm thấy thì là số Fibonacci, ngược lại không phải là số Fibonacci

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int i,p;  long long f[1000000];  int main()  {  f[0]=0; f[1]=1;  for(i=2;f[i-1]+f[i-2]<=1e10;i++) f[i]=f[i-1]+f[i-2];  int t; cin>>t;  for(int a=1; a<=t; a++) {  int d=0;  long long k; cin>>k;  for(int b=0; b<i; b++)  if(k==f[b]) d=1;  if(d!=0) cout<<"IsFibo"<<endl;  else cout<<"IsNotFibo"<<endl;  }  } |

## **Số chính phương chẵn (mahl.\*)**

Cho số nguyên dương , hãy cho biết xác suất để chọn được một ước số chính phương chẵn của số nguyên dương trong số các ước số thực sự của nó (không tính ).

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương cho biết số lượng testcase.

+ dòng tiếp theo, mỗi dòng ghi một số nguyên dương .

**Kết quả:** Với mỗi testcase tương ứng ghi kết quả trên một dòng dưới dạng phân số tối giản

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **output** |
| 4  2  8  36  900 | 0  1/3  1/8  3/26 |

**Hướng dẫn:** Với mỗi số nguyên dương ta cần:

+ Tìm bằng cách đếm số lượng ước số của nó (không tính )

+ Tìm bằng cách đếm số lượng ước số chẵn là số chính phương của nó (không tính )

+ Tối giản phân số

Lưu ý trường hợp

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main(){  int t; cin>>t;  while(t--){  int n; cin>>n;  int tu=0,mau=1;  for(int i=2;i\*i<n;i++)  if(n%i==0) {  mau+=2;  if((int)sqrt(i)==sqrt(i) and i%2==0) tu++;  if((int)sqrt(n/i)==sqrt(n/i) and n/i%2==0) tu++;  }  int sn=sqrt(n);  if(sn==sqrt(n)) {  mau++;  if((int)sqrt(sn)==sqrt(sn) and sn%2==0) tu++;  }  if(tu==0) cout<<0<<endl; else {  int tmp=\_\_gcd(tu,mau);  cout<<tu/tmp<<"/"<<mau/tmp<<endl;  }  }  } |

## **Reverse Game (rega.\*)**

Tí và Tèo đang chơi một trò chơi. Họ có quả bóng được xếp thành một hàng và đánh số từ 0 đến . Ban đầu quả bóng có số hiệu sẽ nằm ở vị trí thứ . Tí và Tèo sẽ lật ngược dãy các quả bóng liên tiếp nhau lần, lần thứ họ sẽ lật ngược các quả bóng từ vị trí đến vị trí .

Ví dụ với quả bóng ban đầu

Lần 0: 3 2 1 0

Lần 1: 3 0 1 2

Lần 2: 3 0 2 1

Lần 3: 3 0 2 1

**Yêu cầu:** Hãy cho biết sau lần lật ngược vị trí các quả bóng như trên thì quả bóng có số hiệu nằm ở vị trí bao nhiêu?

**Dữ liệu vào:** Gồm nhiều bộ dữ liệu

+ Dòng đầu tiên ghi số nguyên cho biết số lượng bộ dữ liệu

+ dòng tiếp theo mỗi dòng cho biết thông tin một bộ dữ liệu gồm hai số nguyên lần lượt là và .

**Kết quả:**

+ Với mỗi bộ dữ liệu in ra một số nguyên cho biết vị trí của sau lần lật ngược dãy số.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  3 1  5 2 | 2  4 |

**Hướng dẫn:**

***Cách 1:*** Thực hiện các bước như trong đề bài:

Tại lượt thứ lật ngược dãy số từ vị trí đến vị trí thứ

Sau lượt lật ngược ta tìm vị trí của

Độ phức tạp:

***Cách 2:*** Tìm quy luật

Nhận thấy với cách làm như trên thì sau lượt lật ngược, các giá trị lần lượt nằm ở vị trí tương tự các giá trị nằm ở vị trí Từ quy luật này ta có thể tạo được dãy số sau lần lật ngược trong

Việc còn lại là tìm kiếm giá trị .

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int a[100001];  int main()  {  int t; cin>>t;  while(t--) {  int n,k; cin>>n>>k;  int na=0;  for(int i=0;i<n/2;i++) {  a[n-i-1]=na++;  a[i]=na++;  }  if(n %2==1) a[n/2]=n-1;  cout<<a[k]<<endl;  }  } |

## **Hai điểm xa nhất – easy (2point.\*)**

Cho điểm trên mặt phẳng tọa độ, điểm thứ có tọa độ trong đó . Hãy tìm hai điểm sao cho khoảng cách giữa hai điểm đó là xa nhất.

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương

+ dòng tiếp theo, dòng thứ ghi hai số nguyên cho biết tọa độ của điểm thứ

**Kết quả:** In ra khoảng cách giữa hai điểm xa nhất, kết quả chỉ cần lấy phần nguyên

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  -1 0  1 0  0 1  0 -1 | 2 |

**Hướng dẫn:**

Với một điểm bất kỳ đều có hoành độ hoặc tung độ bằng 0 do vậy các điểm này chỉ nằm trên trục tung hoặc trục hoành.

Ta sẽ cần tìm các giá trị lần lượt là hoành độ nhỏ nhất, hoành độ lớn nhất, tung độ nhỏ nhất, tung độ lớn nhất.

Khoảng cách xa nhất giữa hai điểm là giá trị lớn nhất trong các giá trị:

1.

2.

3.

4.

5.

6.

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  int main() {  int n;  long long xmax=-1e6,ymax=-1e6,xmin=1e6,ymin=1e6;  cin>>n;  while(n--) {  long long x,y; cin>>x>>y;  xmax=max(xmax,x);  xmin=min(xmin,x);  ymax=max(ymax,y);  ymin=min(ymin,y);  }  double rest=max(xmax-xmin,ymax-ymin);  rest=max(rest,sqrt(xmax\*xmax+ymin\*ymin));  rest=max(rest,sqrt(xmax\*xmax+ymax\*ymax));  rest=max(rest,sqrt(xmin\*xmin+ymin\*ymin));  rest=max(rest,sqrt(xmin\*xmin+ymax\*ymax));  cout<<(long long)rest;  } |

## **Chọn một số (tcone.\*)**

Cho số nguyên dương và dãy số nguyên . Hãy tìm số sao cho là ước chung của đúng số trong dãy. Ví dụ với dãy số 3, 6, 18, 12 thì hoặc là kết quả vì là ước chung của các số và không phải là ước của 3.

**Dữ liệu vào:**

+ Dòng đầu ghi số nguyên dương

+ Dòng thứ 2 ghi dãy số

+ Dữ liệu vào luôn đảm bảo có kết quả

**Kết quả:** Số nguyên tìm được, nếu có nhiều số thỏa mãn yêu cầu bài toán thì in số có giá trị lớn nhất.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  3 6 18 12 | 6 |

**Hướng dẫn:**

Đề bài cho hướng dẫn rất rõ là là ước chung của số, do vậy ta cần thực hiện lượt, lượt thứ cần tìm ước chung lớn nhất của tất cả các số trong dãy (trừ số ). Kết quả cuối cùng là số lớn nhất trong các số tìm được.

Nếu làm đúng ý tưởng ở trên ta có độ phức tạp thuật toán là trong đó là độ phức tạp khi tìm ước chung lớn nhất của 2 số.

Để đạt 100% số điểm của bài này ta chuẩn bị trước 2 mảng:

+ là ước chung lớn nhất của phần tử đầu tiên trong

+ là ước chung lớn nhất của các phần tử

Như vậy ước chung lớn nhất của dãy số khi không có là ước chung lớn nhất của và .

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long a[100011];  long long ucb[100011],uce[100011];  int n;  int main()  {  cin>>n;  for(int i=1;i<=n;i++) cin>>a[i];  if(n==1) {cout<<a[1]+1; return 0;}  ucb[1]=a[1];  uce[n]=a[n];  for(int i=2;i<=n;i++) ucb[i]=\_\_gcd(ucb[i-1],a[i]);  for(int i=n-1;i>=1;i--) uce[i]=\_\_gcd(uce[i+1],a[i]);  long long res=-1;  for(int i=2;i<n;i++)  {  long long tmp=\_\_gcd(ucb[i-1],uce[i+1]);  if(a[i]%tmp!=0 and res<tmp) res=tmp;  }  if(a[1]%uce[2]!=0 and res<uce[2] ) res=uce[2];  if(a[n]%ucb[n-1]!=0 and res<ucb[n-1]) res=ucb[n-1];  cout<<res;  return 0;  } |

## **Tổng cặp số (pairsum.\*)**

Cho dãy số nguyên dương , ma trận kích thước được xác định như sau:

***ai*** + ***aj*** với ***i***≠ j,

0 với ***i***= ***j***.

***Si,j*** =

**Yêu cầu:** cho và ma trận . Hãy xác định . Dữ liệu đảm bảo có lời giải duy nhất. Các phần tử của có giá trị không vượt quá .

**Dữ liệu vào:**

+ Dòng đầu tiên chứa số nguyên ,

+ Dòng thứ trong dòng sau chứa số nguyên xác định dòng của ma trận .

**Kết quả:** Đưa ra trên một dòng số nguyên xác định dãy số .

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  0 3 6 7  3 0 5 6  6 5 0 9  7 6 9 0 | 2 1 4 5 |

**Hướng dẫn:** Ta có

Từ 3 phương trình trên ta có thể tìm được

Sau khi tính được ta dễ dàng tính được các giá trị con lại dựa vào công thức

**Chương trình tham khảo:**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  const long lm = 1009;  long A[lm], F[lm][lm], n;  int main(){  cin >> n;  for (int i=1; i<=n; i++){  for (int j=1; j<=n; j++) cin >> F[i][j];  }  A[1] = (F[1][2]-F[2][3]+F[1][3])/2;  cout << A[1] << " ";  for (int i=2; i<=n; i++){  A[i] = F[1][i]-A[1];  cout << A[i] << " ";  }  } |