# Bài A:

Đầu tiên bạn cần tách các từ và các số vào 2 mảng riêng rẽ (nếu là từ thì chỉ gồm các kí tự từ a->z), sau đó, sắp xếp 2 mảng này, rồi in ra theo đúng thứ tự từ và số là AC.

# Bài B:

Làm trâu bò

# Bài C:

Thuật toán: Sắp xếp nhanh.

Tiêu chí sắp xếp: Đầu tiên bạn phải sắp xếp dãy từ 1->n thành dãy có giá trị tăng dần, nếu 2 ngày có giá trị bằng nhau thì ngày nào có số thứ tự ngày thấp hơn đứng trước. Sau đó, bạn tiếp tục sắp xếp dãy từ 1-.>k1 theo tiêu chí số thứ tự ngày thấp hơn đứng trước, rồi in ra dãy từ 1->k1 là được dãy đầu tiên. Dãy thứ hai tương tự như vậy.

Gợi ý: Sử dụng struct, viết hàm so sánh, và sử dụng hàm sort trong algorithm sẽ rất nhanh.

# Bài D:

Phương pháp quy hoạch động:

Gọi F[i] là số cách chọn nếu có i viên thuốc.

Gọi T[i][j] là số cách chọn i viên mà có j viên là nguyên.

Dễ thấy điều kiện chọn là j >= i-j (số viên nguyên luôn >= số viên nửa)

* Nếu chọn viên nguyên: T[i][j] = T[i-1][j-1]
* Nếu chọn viên nửa: T[i][j] = T[i-1][j]

Suy ra: T[i][j] = T[i-1][j-1] + T[i-1][j]

Dễ thấy F[j] = T[i][j] nếu j = i-j.

# Bài E:

Cách nhập dữ liệu: Dữ liệu kết thúc khi không nhập được nữa, bạn có thể dùng điều kiện ví dụ như “while (cin>>s) {...}” để nhập dữ liệu.

Xử lý số lớn + so sánh xâu.

# Bài F:

Làm trâu bò.

# Bài G:

# Bài H:

Phương pháp: Quy hoạch động

Gọi P[m][v] là số cách phân tích số v thành tổng các số nhỏ hơn hoặc bằng m.

Nếu v<m thì P[m][v] = P[m-1][v]. Ngược lại: P[m][v]=P[m-1][v]+P[m][v-m]. (Tham khảo phần quy hoạch động trong cuốn giải thuật và lập trình – Lê Minh Hoàng).

Gọi F[i] là số UNIMODAL PALINDROMIC DECOMPOSITIONS của số i.

Dễ thấy:

* Nếu i chẵn: F[i] = tổng các P[j][(i-j)/2] với các j chẵn <=i, trong trường hợp i=j thì F[i][j] cộng thêm P[j][j]

Ví dụ: với F[6] thì ta có: 6 = 3+3+0 = 2+2+2 = 1+1+4 = 0+0+6

* F[6] = tổng số cách phân tích 3 thành các số <=3, 2 thành các số <=2, 1 thành các số <=4, 0 thành các số <=6.
* Nếu i lẻ: F[i] = tổng các P[j][(i-j)/2] với các j lẻ <=i

# Bài I:

Trước tiên, bạn cần phải sinh tất cả các số nguyên tố bé hơn 123456\*2 vào một mảng riêng (ví dụ mảng A). Với mỗi số n, bạn chỉ cần đếm số các số trong mảng A mà lớn hơn n và bé hơn hoặc bằng 2\*n.

# Bài J:

Sử dụng cấu trúc stack. Mỗi khi đến dấu mở ngoặc thì push vào stack, nếu gặp đóng ngoặc thì xét xem phần tử ở đỉnh stack có phù hợp với dấu đóng ngoặc không (tức là nếu dấu đóng ngoặc là ‘)’ thì phần tử ở đỉnh stack là ‘(‘, nếu dấu đóng ngoặc là ‘]’ thì phần tử ở đỉnh stack là ‘[‘), nếu sai break luôn. Cứ thế xét cho hết xâu. Nếu sau khi xét hết, stack rỗng tức là biểu thức đúng.