

BÀI TẬP DẠY HỌC SINH GIỎI MÔN TIN HỌC THPT

Bảng mục lục

CHUYÊN ĐỀ: XÂU KÍ TỰ	4
XKT.BÀI 1. Xóa kí tự	4
XKT.BÀI 2. Giải nén xâu	4
XKT.BÀI 3. Xếp hàng	4
XKT.BÀI 4. Tìm chuỗi đối xứng dài nhất	5
XKT.BÀI 5. Tìm số kí tự cần chèn để được xâu đối xứng	6
XKT.BÀI 6. Tìm chữ số thứ N	6
XKT.BÀI 7. Tìm k chữ số từ chữ số thứ N	7
XKT.BÀI 8. Xóa số để thu được số lớn nhất	7
XKT.BÀI 9. Xóa kí tự để thu được số lớn nhất	7
XKT.BÀI 10. Xóa K số để thu được số nhỏ nhất	7
XKT.BÀI 11. Dây ngoặc đúng	8
XKT.BÀI 12. Bài toán biểu thức dấu ngoặc	8
XKT.BÀI 13. Mật mã	9
XKT.BÀI 14. Chuẩn hóa họ tên (dạng bài toán chuẩn hóa văn bản)	9
XKT.BÀI 15. Cộng hai số nguyên lớn	9
XKT.BÀI 16. So sánh hai số nguyên	9
XKT.BÀI 17. Chuẩn hóa ngày tháng	10
XKT.BÀI 18. Chuỗi chẵn - Lẻ	10
XKT.BÀI 19. Điền khuyết xâu kí tự	10
XKT.BÀI 20. Giải mã	11
XKT.BÀI 21. Rút gọn xâu	11
XKT.BÀI 22. Xâu nghịch đảo	11
XKT.BÀI 23. Tính số lần lặp lại của xâu s1 trong xâu s2	11
XKT.BÀI 24. Xâu con chung dài nhất	12
XKT.BÀI 25. Đoạn chung	13
XKT.BÀI 26. Nén và giải nén xâu (theo độ dài loạt)	13
XKT.BÀI 27. Sắp xếp xâu theo số lượng kí tự trong xâu	14
XKT.BÀI 28. Kiểm tra chuỗi	14
XKT.BÀI 29.	15
XKT.BÀI 30. Đoạn con dài nhất	15
XKT.BÀI 31. Mã hoá và giải mã văn bản	16
XKT.BÀI 32. Tính tỉ lệ chữ nguyên âm	16
XKT.BÀI 33. Chữ cái xuất hiện	16
XKT.BÀI 34. Biến đổi xâu	16
XKT.BÀI 35. Chuẩn hóa văn bản (1)	17
XKT.BÀI 36. Chuẩn hóa văn bản (2)	17
XKT.BÀI 37. Sắp xếp xâu	17
XKT.BÀI 38. Mã hóa Caesar (1)	18
XKT.BÀI 39. Mã hóa Caesar (2)	18
XKT.BÀI 40. Mã hoá hồ sơ	19
XKT.BÀI 41. Tìm từ đầu tiên dài nhất trong xâu	20
XKT.BÀI 42: Chiếc nón diệu kì	20
XKT.BÀI 43. Xâu nhị phân	20
XKT.BÀI 44. Tìm số nhỏ nhất lớn hơn X có cùng chữ số với X	21
CHUYÊN ĐỀ: HOÁN VỊ, CHỈNH HỢP, TỔ HỢP	23
HVCH. BÀI 1. Hoán vị (123...n)	23
HVCH. BÀI 2. Liệt kê các hoán vị của một xâu	23
HVCH. BÀI 3. Liệt kê xâu	23
HVCH. BÀI 4. Tạo sơn tổng hợp	23
HVCH. BÀI 5. Trộn đề	23
CHUYÊN ĐỀ: MẢNG MỘT CHIỀU, HAI CHIỀU	24

M1C2C. BÀI 1. Tần số.....	24
M1C2C. BÀI 2: Số may mắn.....	24
M1C2C. BÀI 3. Mã số nhân viên	25
M1C2C. BÀI 4. Mua hàng	25
M1C2C. BÀI 5. Tập số đặc biệt.....	25
M1C2C. BÀI 6. Mua vé	26
M1C2C. BÀI 7. Tổng các chữ số.....	26
M1C2C. BÀI 8. Chia đồ vật	26
M1C2C. BÀI 9. Tìm dãy con có tổng lớn nhất (không chọn 3 phần tử liên tiếp)	27
M1C2C. BÀI 10. Chọn phần thưởng	27
M1C2C. BÀI 11. Chia dãy số thành k nhóm có tổng bằng nhau.....	27
M1C2C. BÀI 12. Dãy con liên tiếp có tổng lớn nhất.....	28
M1C2C. BÀI 13. Dãy con cấp số cộng.....	28
M1C2C. BÀI 14. Dãy con liên tiếp có tổng chia hết cho k.....	28
M1C2C. BÀI 15. Dãy con liên tiếp có tổng bằng M	28
M1C2C. BÀI 16. Dãy con nguyên tố dài nhất	29
M1C2C. BÀI 17. Dãy con có tổng bằng S (dãy không liên tiếp).....	29
M1C2C. BÀI 18. Tính tổng	30
M1C2C. BÀI 19. Sắp xếp mảng 1 chiều.....	30
M1C2C. BÀI 20. Sắp xếp.....	30
M1C2C. BÀI 21. Số tự nhiên nhỏ nhất.....	31
M1C2C. BÀI 22. Dãy phân số.....	31
M1C2C. BÀI 23. Độ lệch.....	32
M1C2C. BÀI 24. Số siêu nguyên tố	32
M1C2C. BÀI 25. Bình chọn qua điện thoại	32
M1C2C. BÀI 26. Quan hệ huyết thống.....	33
M1C2C. BÀI 27. Tìm số	33
M1C2C. BÀI 28. Số âm lớn nhất.....	33
M1C2C. BÀI 29. Trò chơi may mắn.....	34
M1C2C. BÀI 30. Sắp xếp.....	34
M1C2C. BÀI 31. Xếp việc	34
M1C2C. BÀI 32. Dãy số	35
M1C2C. BÀI 33. Số thân thiện.....	35
M1C2C. BÀI 34. Sa mạc.....	35
M1C2C. BÀI 35. Vườn trường.....	36
M1C2C. BÀI 36. Kho an toàn	37
M1C2C. BÀI 37. Bố trí xe	37
M1C2C. BÀI 38. Kết bạn.....	38
M1C2C. BÀI 39. Đếm số ô vuông đơn vị.....	38
M1C2C. BÀI 40. Chuyển vị ma trận	39
M1C2C. BÀI 41. Sắp xếp trên ma trận	39
M1C2C. BÀI 42. Bài toán cái túi.....	40
M1C2C. BÀI 43. Trò chơi (dạng bài toán cái túi - chọn 1 hoặc nhiều vật)	40
M1C2C. BÀI 44. Tổng các hàng của ma trận	40
M1C2C. BÀI 45. Ma trận số	41
M1C2C. BÀI 46. Sắp xếp ma trận.....	41
M1C2C. BÀI 47. Sắp xếp mảng hai chiều tăng theo hàng, cột.....	41
M1C2C. BÀI 48. Rút tiền từ máy ATM	42
M1C2C. BÀI 49. Đóng gói sản phẩm.....	42
M1C2C. BÀI 50. Hệ thống cảnh báo thảm họa.....	42
M1C2C. BÀI 51. Các điểm cực tiểu.	45
M1C2C. BÀI 52. Chữ số thứ N	45
M1C2C. BÀI 53. Lá bài	45
M1C2C. BÀI 54. Trộn hai tệp	46
M1C2C. BÀI 55. Trộn hai tệp thành tệp tăng dần	46

M1C2C. BÀI 56. Tìm số lần lặp lại nhiều nhất của một số trong dãy số.....	47
M1C2C. BÀI 57. Xếp khách.....	47
M1C2C. BÀI 58. Mã số sách.....	47
M1C2C. BÀI 59. Tam giác số.....	48
M1C2C. BÀI 60. Thuê máy tính.....	48
M1C2C. BÀI 61. Hàng đợi.....	48
M1C2C. BÀI 62. Bản tin bóng đá.....	49
M1C2C. BÀI 63. Nhà chung cư.....	50
M1C2C. BÀI 64. Hành tinh XYZ.....	50
M1C2C. BÀI 65. Bảng số.....	51
M1C2C. BÀI 66. Tìm chữ số thứ M.....	51
M1C2C. BÀI 67. Diện tích miền phủ.....	52
M1C2C. BÀI 68. Tính diện tích.....	52
M1C2C. BÀI 69. Tạo bảng.....	53
M1C2C. BÀI 70. Tìm số âm lớn nhất.....	54
M1C2C. BÀI 71. Trạm canh.....	54
M1C2C. BÀI 72. Hình chữ nhật.....	55
M1C2C. BÀI 73. Dây con chung dài nhất.....	55
M1C2C. BÀI 74. Số lượng nhóm đề tài.....	56
M1C2C. BÀI 75. Siêu nguyên tố.....	56
M1C2C. BÀI 76. Cấp số cộng.....	56
M1C2C. BÀI 77. Đếm nhóm bạn trong hội trại.....	56
M1C2C. BÀI 78. Tổng các chữ số.....	57
M1C2C. BÀI 79. Phân tích một số thành tổng các số.....	57
M1C2C. BÀI 80. Phân phối hàng cứu trợ.....	61
M1C2C. BÀI 81. Tìm điểm thuộc tất cả N hình chữ nhật.....	61
M1C2C. BÀI 82. Con thạch sùng.....	61

CHUYÊN ĐỀ: XÂU KÍ TỰ

Các thủ tục và hàm chuẩn xử lý chuỗi ký tự (bổ sung, các hàm – thủ tục khác hs xem SGK)

* **Thủ tục STR(value, st)** Thủ tục này thực hiện việc chuyển đổi giá trị kiểu số(value) sang dạng chuỗi ký tự và gán cho biến st.

Ví dụ: n là một số nguyên có giá trị: n:=150;

STR(n:5,st) sẽ cho kết quả chuỗi st là: st=' 150';

* **Thủ tục VAL(st, value, code)** đổi một chuỗi ký tự st sang dạng số và gán cho biến value, nếu biến đổi thành công thì code sẽ nhận giá trị bằng 0, ngược lại thì cho giá trị khác không

Ví dụ: VAL('123',value,code) lúc này code sẽ nhận giá trị bằng 0 và value=123

BÀI TẬP

XKT.BÀI 1. Xóa ký tự

(câu 1, đề thi HSG tỉnh Quảng Bình năm 2013-2014, lớp 11)

Cho một chuỗi ký tự St có độ dài tối đa 255 ký tự, các ký tự được lấy từ tập: 'a' ... 'z'; 'A' ... 'Z'; '0' ... '9'.

Yêu cầu: Hãy xóa hết các ký tự chữ số trong chuỗi St.

Dữ liệu vào: Ghi trong file văn bản DELECHAR.INP có cấu trúc như sau:

- Dòng 1: Ghi chuỗi St.

Dữ liệu ra: Ghi ra file văn bản DELECHAR.OUT theo cấu trúc như sau:

- Dòng 1: Ghi chuỗi St sau khi đã xóa đi các ký tự chữ số.

Ví dụ:

DELECHAR.INP	DELECHAR.OUT
abc123DEA97ijk	abcDEAijk

XKT.BÀI 2. Giải nén chuỗi

(câu 2, đề thi HSG tỉnh Quảng Bình năm 2013-2014, lớp 11)

Để tiết kiệm chi phí trong việc gửi và nhận dữ liệu thông qua các kênh truyền, các file văn bản trước khi được gửi đi đều được nén lại để giảm dung lượng bộ nhớ. Sau khi nhận được các file văn bản gửi đến, muốn đọc được nội dung thì các file vừa nhận phải được giải nén.

Việc nén một chuỗi văn bản chỉ chứa các ký tự chữ cái in hoa được thực hiện như sau: Một đoạn liên tiếp các ký tự chữ cái giống nhau được thay bằng một đoạn ký tự mới gồm các ký tự chữ số thể hiện số lượng của chữ cái và tiếp theo sau là ký tự chữ cái đó (AAAAAAA được thay bằng 7A), nếu chỉ có một ký tự chữ cái thì giữ nguyên ký tự đó. Ví dụ: AAAAABBBBCDDD sẽ được nén lại là 6A3BC3D.

Việc giải nén là chuyển một chuỗi đã được nén trở về chuỗi gốc ban đầu trước khi nó được nén.

Ví dụ: 6A3BC3D được chuyển thành AAAAABBBBCDDD.

Cho một file văn bản gồm nhiều dòng, mỗi dòng chứa một chuỗi văn bản đã được nén, file chỉ chứa tối đa 100 dòng (Mỗi chuỗi văn bản gốc trước khi nén có tối đa 255 ký tự).

Yêu cầu: Hãy thực hiện giải nén các chuỗi văn bản trên mỗi dòng trong file đã cho chuyển thành các chuỗi gốc ban đầu.

Dữ liệu vào: Ghi trong file văn bản UNZIP.INP có cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương N là số lượng chuỗi trong file văn bản.

- Dòng thứ i trong N dòng tiếp theo: Mỗi dòng ghi một chuỗi đã được nén.

Dữ liệu ra: Ghi ra file văn bản UNZIP.OUT theo cấu trúc như sau:

Dữ liệu được ghi trên N dòng, dòng thứ i ghi chuỗi đã được giải nén tương ứng với chuỗi đã được nén trong file dữ liệu vào.

Ví dụ:

UNZIP.INP	UNZIP.OUT
2	AAAAAABBBBCDDD
6A3BC3D	AAAAABC
5ABC	


Mở rộng bài toán: viết chương trình nén chuỗi

XKT.BÀI 3. Xếp hàng

(câu 1, đề thi HSG tỉnh Đồng Tháp năm 2013-2014, lớp 12)

Trong giờ học thể dục, thầy giáo xếp N học sinh thành một hàng và vị trí của học sinh được đánh số từ 1 đến n từ trái sang phải. Ban đầu học sinh đứng tùy ý trong hàng. Tuy nhiên, để tôn trọng các bạn nữ, thầy muốn các bạn nam không được đứng liền trước bạn nữ nào (đứng liền trước ở đây được hiểu rằng vị trí của bạn nam là i và vị trí của bạn nữ là i+1). Để thực hiện quy định này, thầy bắt đầu đi từ đầu hàng đến cuối hàng, khi gặp bạn nam nào được đứng liền trước một bạn nữ, thầy sẽ yêu cầu bạn nam này đổi chỗ cho bạn nữ rồi đi tiếp đến bạn sau đó. Chú

ý rằng trong một lượt sắp xếp, thầy chỉ đi theo một chiều và mỗi bạn nam chỉ được đổi chỗ một lần. Tất nhiên là chỉ với một lượt sắp xếp như vậy thì vẫn có thể có nhiều vị trí mà bạn nam đứng trước nữ xuất hiện thêm nên ta cần phải đi làm thao tác sắp xếp này nhiều lần.

Yêu cầu: cho hai số nguyên dương n, t với  và một dãy kí tự G và B , trong đó G là kí hiệu bạn nữ và B là kí hiệu bạn nam thể hiện vị trí các học sinh của lớp ban đầu. Hỏi sau lần thay đổi thứ t thì vị trí các học sinh như thế nào và bao nhiêu lần thao tác thì thầy giáo sẽ hoàn tất việc sắp xếp này.

Dữ liệu vào: trong file xephang.inp gồm có: dòng thứ nhất chứa hai số nguyên dương n và t cách nhau bởi một khoảng trắng, dòng thứ hai gồm một dãy n kí tự ' G ' và ' B ' biểu thị vị trí đứng của các học sinh trong hàng (từ trái qua phải tương ứng với chỉ số vị trí tăng dần).

Kết quả ra: in ra file xephang.out hai dòng: dòng thứ nhất gồm dãy n kí tự ' G ' và ' B ' biểu thị vị trí đứng của các học sinh sau khi thầy giáo xếp lại lần thứ t , dòng thứ hai là một số nguyên cho biết số lần thầy giáo cần sắp xếp. Chú ý rằng số t có thể lớn hơn số lần thầy giáo cần sắp xếp.

Ví dụ:

XEPHANG.INP	XEPHANG.OUT
5 1 BGGBG	GBGGB 3
5 2 BGGBG	GGBGB 3
4 1 GGGB	GGGB 0

DẠNG BÀI TOÁN LIÊN QUAN ĐẾN XÂU ĐỐI XỨNG

XKT.BÀI 4. Tìm chuỗi đối xứng dài nhất

(Câu 1, đề thi HSG tỉnh Ninh Thuận năm học 2013-2014, lớp 12)

Một chuỗi kí tự được gọi là đối xứng nếu đọc từ trái qua phải cũng giống như đọc nó từ phải qua trái.

Ví dụ: 'EUROORUE'; 'DATATAD' là chuỗi đối xứng.
'STRING'; 'TRANTIENDAT' là chuỗi không đối xứng.

Cho chuỗi kí tự S có chiều dài N ($10 \leq N \leq 1000$). Hãy tìm chiều dài chuỗi con đối xứng dài nhất trong S . Chuỗi con đối xứng trong S là chuỗi gồm một số kí tự liên tiếp nhau trong S có độ dài nhỏ hơn hoặc bằng N .

Dữ liệu: Cho trong file văn bản doixung.inp.

- Dòng đầu ghi giá trị N ($10 \leq N \leq 1000$).
- Dòng sau gồm N kí tự liên tiếp là các chữ cái in hoa ($A \rightarrow Z$).

Kết quả: Ghi vào file văn bản doixung.out độ dài của chuỗi con đối xứng dài nhất (trường hợp không có thì ghi 0).

Ví dụ:

DOIXUNG.INP	DOIXUNG.OUT
20 ABCDEFABABBABAFFFFFF	10

Ý tưởng

Cách 1: Duyệt xâu từ kí tự đầu đến kí tự thứ $\text{length}(S) - 1$. Với mỗi lần duyệt ta sẽ xét xâu con lần lượt từ vị trí đó có đối xứng không, nếu đối xứng thì so sánh độ dài với xâu kết quả, nếu lớn hơn thì thay đổi xâu kết quả.

Cách 2: Dùng phương pháp quy hoạch động

Dùng mảng $F[i, j]$ có ý nghĩa $F[i, j] = \text{true/false}$ nếu đoạn gồm các kí tự từ i đến j của S có/không là palindrome.

Ta có công thức:

- $F[i, i] = \text{true}$
- $F[i, j] = F[i+1, j-1]$ (nếu $s[i] = s[j]$)
- $F[i, j] = \text{false}$ (nếu $s[i] \neq s[j]$)

Đoạn chương trình như sau:

```
Fillchar(F, sizeof(F), false);
For i:=1 to n do F[i, i]:= true;
For k:=1 to n-1 do
  For i:=1 to n-k do
    Begin
      J:= i+k;
      F[i, j]:= F[i+1, j-1] and (s[i]=s[j]);
    End;
```

Kết quả là $\max(j-i+1) <= j$ thỏa $F[i, j] = \text{true}$;

*** Đoạn chương trình in ra chuỗi đối xứng dài nhất:**

```
for i:=1 to n do
  for j:=1 to n do
    begin
      d:=j-i+1;
      if F[i,j] and (d>dmax) then
        begin
          dmax:=d;
          csd:=i;
          csc:=j;
        end;
    end;
  end;
for i:=csd to csc do write(S[i]);
```

XKT.BÀI 5. Tìm số kí tự cần chèn để được xâu đối xứng

(Thanh Hóa, 2008-2009)

Xâu đối xứng là xâu đọc giống nhau nếu ta bắt đầu đọc từ trái qua phải hoặc từ phải qua trái. Ví dụ, xâu RADAR là xâu đối xứng, xâu TOMATO không phải là xâu đối xứng.

Yêu cầu: Cho một xâu S gồm không quá 200 kí tự. Cho biết S có phải là xâu đối xứng hay không? Nếu không, cho biết số kí tự ít nhất cần thêm vào S để S trở thành xâu đối xứng.

Dữ liệu vào: từ file văn bản **doixung.inp**, gồm duy nhất 1 dòng ghi xâu S.

Kết quả: Ghi ra file văn bản **doixung.out**, duy nhất số k là số kí tự ít nhất cần thêm vào S để S trở thành xâu đối xứng. Nếu xâu S đã cho là đối xứng thì ghi k = 0.

Ví dụ:

DOIXUNG.INP	DOIXUNG.OUT
RADAR	0

DOIXUNG.INP	DOIXUNG.OUT
TOMATO	3

Hướng dẫn:

Cách 1

Trong xâu đối xứng thì vị trí i sẽ đối xứng với vị trí $\text{length}(s)-i+1$

Do đó ta xét từng cặp kí tự ở vị trí đối xứng nhau:

+ Nếu kí tự ở vị trí i và kí tự ở vị trí đối xứng là $\text{length}(s)-i+1$ khác nhau và kí tự ở vị trí i và vị trí $\text{length}(s)-i$ cũng khác nhau thì ta chèn kí tự ở vị trí thứ i và xâu S ở vị trí $\text{length}(s)-i+2$ và tính lại độ dài xâu, khi đó ta sẽ có kí tự thứ i và kí tự thứ $\text{length}(s)-i+1$ sẽ giống nhau.

+ Nếu kí tự ở vị trí i và kí tự ở vị trí đối xứng là $\text{length}(s)-i+1$ khác nhau và kí tự ở vị trí i và vị trí $\text{length}(s)-i$ giống nhau thì ta chèn kí tự ở vị trí thứ $\text{length}(s)-i+1$ vào xâu S ở vị trí i và tính lại độ dài xâu, khi đó ta sẽ có kí tự thứ i và kí tự thứ $\text{length}(s)-i+1$ sẽ giống nhau, kí tự ở vị trí i+1 và $\text{length}(s)-i$ cũng giống nhau.

Ví dụ: S='TOMATO'

Ở vị trí i=1: kí tự ở vị trí i và vị trí đối xứng là $\text{length}(s)-i+1$ khác nhau; kí tự ở vị trí i và kí tự ở vị trí $\text{length}(s)-i$ giống nhau. Khi đó cần chèn kí tự 'O' vào vị trí 1 để được xâu mới, khi đó ta có cặp kí tự đối xứng.

$S[i] \neq S[\text{length}(s)-i+1]$

	1	2	3	4	5	6
S	T	O	M	A	T	O

$S[i] = S[\text{length}(s)-i]$

Ta có hình ảnh xâu S sau khi chèn:

i	1	2	3	4	5	6	7
S	O	T	O	M	A	T	O

Cách 2: Gọi S2 là xâu đảo của xâu ST ban đầu, T là xâu con chung của hai xâu S2 và ST. Khi đó các kí tự của ST không thuộc T chính là các kí tự cần chèn vào xâu ST để ST trở thành xâu đối xứng.

Bài toán trở thành bài toán tìm dãy con chung dài nhất của hai xâu bằng phương pháp quy hoạch động.

XKT.BÀI 6. Tìm chữ số thứ N.

Khi viết các số tự nhiên tăng dần từ 1, 2, 3, ... liên tiếp nhau, ta nhận được một dãy các chữ số thập phân vô hạn, đoạn đầu tiên của dãy sẽ là: 1234567891011121314151617181920...

Yêu cầu: Hãy tìm chữ số thứ N của dãy số vô hạn trên.

Dữ liệu: Cho trong file NUMBER.INP gồm một nguyên dương N ($N < 10^6$).

Kết quả: Ghi kết quả ra file NUMBER.OUT.

Ví dụ:

NUMBER.OUT	NUMBER.OUT	Giải thích kết quả
------------	------------	--------------------

21	5	Chữ số thứ 21 trong dãy là chữ số 5
----	---	-------------------------------------

XKT.BÀI 7. Tìm k chữ số từ chữ số thứ N

Từ chuỗi nhị phân $S=10^*$, người ta tạo ra chuỗi nhị phân mới bằng cách ghép chuỗi S ban đầu với chính nó sau khi đã đảo tất cả các bit của S (nghĩa là 1 thành 0 và 0 thành 1) và cứ lặp lại các thao tác trên cho đến khi chuỗi S có không ít hơn N chữ số

Ví dụ:

Độ dài 2: chuỗi $s=10^*$

Độ dài 4: chuỗi $s=1001^*$ (ghép '10' với '01');

Độ dài 8: chuỗi $s=10010110^*$ (ghép '1001' với '0110');

Độ dài 16: chuỗi $s=1001011001101001^*$ (ghép '10010110' với '01101001');

Yêu cầu:

Cho 2 số N, K ($0 < N < 10^8$, $0 < K < 10$). Hãy tìm K kí tự liên nhau bắt đầu từ kí tự thứ N của chuỗi S

XKT.BÀI 8. Xóa số để thu được số lớn nhất

(Câu 2, đề thi hsg Thanh Hóa năm học 2013-2014, lớp 12)

Cho một dãy gồm N các kí tự có mặt trên bàn phím trong đó có ít nhất 4 chữ số. ($N < 10^6$).

Yêu cầu: Hãy loại bỏ một số kí tự khỏi dãy sao cho 4 kí tự cuối cùng còn lại theo đúng thứ tự đó tạo nên 1 số lớn nhất.

Dữ liệu vào: File văn bản XOASO.INP chứa N kí tự.

Dữ liệu ra: File văn bản XOASO.OUT chứa 4 chữ số tạo thành số lớn nhất.

Ví dụ:

XOASO.INP	XOASO.OUT
24t5j4r05f704y652k393	7693

Ý tưởng:

- Tìm số nhỏ nhất đầu tiên từ bên trái qua và xóa chữ số đó đi, nếu không có thì xóa chữ số cuối cùng.
- Lặp lại thao tác trên cho đến khi chỉ còn lại 4 kí tự.

Bài toán tương tự:

Hãng cung cấp dịch vụ điện thoại XYZ khuyến khích nhiều người đăng kí thuê bao bằng cách: Khi khách hàng đến đăng kí thuê bao thì sẽ được cấp hai số may mắn là số nguyên dương n và k, hãng sẽ khuyến mại người đó một số tiền nhận được từ số n sau khi xóa đúng k chữ số (k nhỏ hơn số chữ số của n).

Hải vừa mới đăng kí thuê bao của hãng và được cung cấp hai số n và k, bạn hãy giúp Hải xóa đi k chữ số của số n để số nhận được là lớn nhất.

Dữ liệu vào file văn bản XOASO.inp:

- Dòng thứ nhất là số n (số chữ số của $n \leq 10^5$)
- Dòng thứ hai là số k ($k < n$)

Kết quả ra file văn bản XOASO.out:

- Một dòng duy nhất là số lớn nhất có được sau khi xóa đi k chữ số của n

Ví dụ:

XOASO.INP	XOASO.OUT
58816	886
2	
2357111317192329	7317192329
6	

XKT.BÀI 9. Xóa kí tự để thu được số lớn nhất

(câu 2, HSG Bạc Liêu, 2011-2012, bảng A, ngày 2)

Cho xâu s gồm ít nhất 3 kí tự số. Xóa bỏ một số kí tự trong xâu s chỉ để lại 3 kí tự số sao cho, vẫn giữ nguyên thứ tự của chúng tạo nên một số có giá trị lớn nhất.

- Dữ liệu vào: từ tệp XOAKT.INP gồm 1 dòng chứa xâu s

- Dữ liệu ra: Ghi ra tệp XOAKT.OUT xâu s chứa 3 kí số còn lại tạo thành số lớn nhất.

Ví dụ:

XOAKT.INP	XOAKT.OUT
124512Hoctin8126123	863

Ý tưởng: thực hiện tương tự bài “Xóa số để thu được số lớn nhất”

XKT.BÀI 10. Xóa K số để thu được số nhỏ nhất

Cho một số nguyên dương có N chữ số ($N \leq 255$). Hãy tìm cách xóa đi k chữ số của N để thu được số nhỏ nhất.

Dữ liệu vào: tệp XOASO.INP gồm một dòng chứa số nguyên dương N

Dữ liệu ra: XOASO.OUT là số lớn nhất tìm được.

XOASO.INP	XOASO.OUT
1245128126123	1112123
6	

Ý tưởng: thực hiện tương tự bài “Xóa số để thu được số lớn nhất”

XKT.BÀI 11. Dãy ngoặc đúng

(Câu 3, đề thi hsg thanh hóa năm học 2013-2014, lớp 12)

Người ta định nghĩa một chuỗi ký tự gồm các ký tự ‘(’ và ‘)’ là một dãy ngoặc đúng như sau:

- Chuỗi rỗng là một dãy ngoặc đúng.
- Nếu X là dãy ngoặc đúng thì (X) cũng là một dãy ngoặc đúng
- Nếu X, Y là những dãy ngoặc đúng thì XY cũng là dãy ngoặc đúng.

Những dãy ngoặc sau là những dãy ngoặc đúng:

- ()()
- ((()))

Những dãy ngoặc sau thì không:

-)()
- (((()))
-)()()

Cho một chuỗi ký tự $T = T_1, T_2, \dots, T_n$, trong đó T_i là một trong hai ký tự ‘(’ hoặc ‘)’ với mọi $i=1..n$.

Yêu cầu: Hãy đếm số cặp i, j ($i < j$) mà chuỗi ký tự thu được từ việc ghép các ký tự liên tiếp T_i, T_{i+1}, \dots, T_j (giữ nguyên thứ tự) là một dãy ngoặc đúng.

Dữ liệu vào: File văn bản DAYNGOAC.INP gồm 2 dòng:

- Dòng thứ nhất chứa số n là độ dài của chuỗi ký tự T ($n \leq 1000$).
- Dòng thứ 2 chứa chuỗi ký tự T

Dữ liệu ra: Ghi ra file văn bản DAYNGOAC.OUT một số duy nhất là kết quả tìm được.

Ví dụ:

DAYNGOAC.INP	DAYNGOAC.OUT
10 (())()()	5

Ý tưởng

Bài này là dạng toán kiểm tra dãy ngoặc đúng. Nhưng trong bài này ta kiểm tra xem có bao nhiêu dãy ngoặc đúng.

1. Viết hàm kiểm tra dãy ngoặc đúng từ i đến j : $ktngoacdung(i, j)$

2. Duyệt chuỗi:

for $i=1$ to $\text{length}(s) - 1$ do

for $j=i+1$ to $\text{length}(s)$ do

if $ktngoacdung(i, j)$ then tăng biến đếm

3. Kết quả chính là giá trị của biến đếm.

XKT.BÀI 12. Bài toán biểu thức dấu ngoặc

Biểu thức dấu ngoặc đúng dẫn nhận được từ biểu thức toán học có chứa các dấu ngoặc tròn bằng cách bỏ hết tất cả các toán hạng và các phép toán.

Ví dụ từ biểu thức: $a - b(c+2(x+y(z+1))) + a(c+x)$ ta nhận được biểu thức dấu ngoặc đúng dẫn $((()))()$. Chính xác hơn, biểu thức dấu ngoặc đúng dẫn được định nghĩa như sau:

a. $()$ là biểu thức dấu ngoặc đúng dẫn.

b. Nếu P là biểu thức dấu ngoặc đúng dẫn thì (P) là biểu thức dấu ngoặc đúng dẫn.

c. Nếu P và Q là biểu thức dấu ngoặc đúng dẫn thì PQ là biểu thức dấu ngoặc đúng dẫn.

Input: tệp NGOAC.INP gồm 2 dòng:

+ Dòng 1: chứa số tự nhiên n .

+ Dòng 2: chứa 1 chuỗi gồm n dấu ngoặc tròn.

Output: tệp NGOAC.OUT ghi "Đúng" nếu biểu thức ngoặc là đúng hoặc ghi "Sai" nếu biểu thức ngoặc là sai.

Input	Output
8 ((()))()	TRUE

Ý tưởng:

Gv hướng dẫn hs viết chương trình kiểm tra việc mở ngoặc và đóng ngoặc có thực hiện đúng hay không.

Nếu gặp đóng ngoặc trước mở ngoặc thì kết luận không hợp lệ, ngược lại thì dùng biến đếm: gặp dấu mở ngoặc thì tăng 1, gặp đóng ngoặc thì giảm 1. Nếu duyệt hết chuỗi mà biến đếm bằng 0 thì kết luận hợp lệ, ngược lại không hợp lệ.

XKT.BÀI 13. Mật mã**(câu 3, đề thi tỉnh Đồng Nai năm học 2011-2012)**

Trung tâm phân gián "XYZ" vừa mới phát hiện một tập tin có chứa mật khẩu để giải mã cho một tài liệu quan trọng. Nhưng tập tin có cài virus nên khi vừa mở tập tin virus bắt đầu hoạt động, và sau một giây sẽ làm treo máy, không xử lý được nữa.

Yêu cầu: hãy xác định mật mã trong tập tin trước khi bị virus tấn công. Biết rằng tập tin được tạo ra từ tập các kí tự sau: dấu chấm phẩy ‘;’, các chữ số ‘0’...‘9’, các chữ cái thường ‘a’,...,‘z’. Trong tập tin có nhiều xâu, mỗi xâu cách nhau bởi kí tự ‘;’ và độ dài không quá k kí tự ($1 \leq k \leq 30$). Mật mã là xâu đứng thứ P theo thứ tự từ điển trong danh sách các xâu.

Hạn chế kĩ thuật: dung lượng tập tin không quá 10^7

Dữ liệu vào từ file văn bản **MATMA.INP** có cấu trúc như sau:

- Dòng đầu là số tự nhiên P ($1 \leq p \leq 10^6$)
- Các dòng sau là nội dung tập tin.

Kết quả: ghi ra file văn bản **MATMA.OUT**

Xâu mật khẩu

Ví dụ

MATMA.INP	MATMA.OUT
3 dhj8k; d9bdc; 2t7p; sa ab4; mmxtq;df k5;pqkv	dfk5

Ý tưởng

Bài này là dạng toán tìm phần tử ở vị trí thứ P trong mảng một chiều.

1. Ta đọc dữ liệu đã cho vào một mảng xâu (mỗi phần tử là một xâu kí tự, các xâu ngăn cách bởi dấu ‘;’).
2. Sử dụng thuật toán QuickSort để sắp xếp các xâu tăng dần theo thứ tự từ điển.
3. Đưa ra xâu ở vị trí P.

XKT.BÀI 14. Chuẩn hóa họ tên (dạng bài toán chuẩn hóa văn bản)

Viết chương trình chuẩn hóa họ tên cho đúng với cách viết tên riêng và cắt bỏ các khoảng trắng thừa (nếu có).

Ví dụ: “*nguyen vAn baY*” thì cho kết quả là “*Nguyen Van Bay*”

Dữ liệu vào: CHUANHOA.INP

- Dòng đầu là số nguyên N.
- N dòng tiếp theo, mỗi dòng là tên một người chưa chuẩn hóa.

Dữ liệu ra: CHUANHOA.OUT mỗi dòng là tên một người đã được chuẩn hóa tương ứng.

Ví dụ

Input	Output
<i>nguyen vAn baY</i>	<i>Nguyen Van Bay</i>

XKT.BÀI 15. Cộng hai số nguyên lớn

Viết chương trình cộng hai số nguyên lớn có độ dài không quá 250 kí tự

Dữ liệu vào: cho trong tệp **CONGSN.INP** có 2 dòng, mỗi dòng là một số tự nhiên.

Dữ liệu ra: ghi ra tệp **CONGSN.OUT** chứa tổng của hai số.

CONGSN.INP	CONGSN.OUT
123234354	123247589010
123124354656	

Ý tưởng

Do số lượng các chữ số có thể lên đến 250 kí tự nên ta không thể sử dụng các kiểu dữ liệu nguyên mà phải sử dụng kiểu xâu để lưu trữ thì mới xử lý đạt yêu cầu.

Bài toán trở thành bài toán tính tổng hai số nguyên lớn.

XKT.BÀI 16. So sánh hai số nguyên

Cho 2 số nguyên dương A, B ($0 < A, B < 10^{100}$).

Yêu cầu: hãy so sánh giá trị của 2 số.

Dữ liệu cho trong file văn bản có tên **SO.INP** gồm 2 dòng:

Dòng đầu chứa số A.

Dòng thứ 2 chứa số B.

Kết quả xuất ra file văn bản SO.OUT gồm 1 dòng duy nhất, chứa số -1,0 hoặc 1 lần lượt tương ứng với các trường hợp sau: $A < B$, $A = B$, và $A > B$.

Ví dụ:

SO.INP
12345678900000001
12345678900000000

SO.OUT
1

Ý tưởng

Do bài này số lượng chữ số có thể rất lớn nên ta phải áp dụng kỹ thuật xử lý số lớn mới giải quyết triệt để được bài toán.

Thủ tục so sánh hai số A, B có thể được xây dựng như sau:

- Xem A, B là xâu kí tự. Nếu độ dài của hai xâu khác nhau, ta thêm chữ số 0 vào đầu xâu ngắn hơn cho đến khi độ dài của hai xâu bằng nhau:

```
while (length(a) < length(b)) do a := '0' + a;  
while (length(b) < length(a)) do b := '0' + b;
```

- Sau đó ta duyệt các chữ số từ trái sang phải, nếu chữ số tương ứng của A bé hơn/lớn hơn của B thì kết luận $A < B$ hay $A > B$ và thoát khỏi thủ tục. Nếu tất cả các cặp chữ số đều giống nhau thì ta kết luận $A = B$.

XKT.BÀI 17. Chuẩn hóa ngày tháng

Viết chương trình chuẩn hóa ngày tháng theo qui định sau:

- Ngày từ 1, 2, ..., 9 thì ghi là 01, 02, ..., 09
- Tháng 1, 2 thì ghi là 01, 02; từ tháng 3, 4, ..., 9 thì chỉ ghi là 3, 4, ..., 9
- Năm phải có đủ 4 chữ số thập phân (chỉ tính từ 1980 đến 2010)
- Dấu phân cách ngày tháng năm là "/"

Ví dụ:

Nguyen Thi A 2-08-93	=>	Nguyen Thi A 02/8/1993
Tran Van B 12/09/05	=>	Tran Van B 12/9/2005
Le Van C 02/07/1996	=>	Le Van C 02/7/1996

Dữ liệu vào: CHUANHOA.INP mỗi dòng là tên một người và ngày sinh chưa chuẩn hóa cách nhau một dấu cách.

Dữ liệu ra: CHUANHOA.OUT mỗi dòng là tên một người và ngày sinh đã được chuẩn hóa cách nhau một dấu cách.

XKT.BÀI 18. Chuỗi chẵn - Lẻ

(BÌNH DƯƠNG – 2013)

Một chuỗi kí tự được định nghĩa như sau:

- Chuỗi lẻ: Là chuỗi có số lần xuất hiện của một tự là số lẻ. Ví dụ: starring;
- Chuỗi chẵn: là chuỗi có số lần xuất hiện của mỗi kí tự là số chẵn. Ví dụ: staats.

Viết chương trình kiểm tra tính chẵn lẻ của mỗi chuỗi kí tự.

Dữ liệu vào: Cho trong file STR.INP: Chứa chuỗi kí tự (tối đa là 1000000 kí tự), từ 'a' đến 'z', 'A' đến 'Z';

Dữ liệu ra: Ghi trong file STR.OUT: Nếu là chuỗi lẻ thì ghi "CHUOILE", nếu là chuỗi chẵn thì ghi "CHUOICHAN", ngoài ra thì ghi "*".

Ví dụ:

STR.INP	STR.OUT
Staats	CHUOICHAN

Ý tưởng: kiểm tra số lần xuất hiện của các kí tự để biết chuỗi chẵn hay lẻ.

XKT.BÀI 19. Điền khuyết xâu kí tự

Cho trước 2 xâu kí tự a, b (chiều dài của mỗi xâu không quá 100).

Yêu cầu: Viết chương trình bổ sung một số kí tự vào a và một số kí tự vào b để hai xâu a và b trở nên giống nhau (phân biệt chữ hoa, thường). Tổng số kí tự bổ sung vào là ít nhất.

Input: File văn bản FS.INP cấu trúc như sau:

- Dòng 1: xâu kí tự a
- Dòng 2: xâu kí tự b

Output: File văn bản FS.OUT cấu trúc như sau:

- Xâu kết quả

Ví dụ:

FS.INP	FS.OUT
abcde	abcdef
abdf	

Ý tưởng.

Ta xóa bớt trong a và b 1 vài kí tự để thu được xâu kí tự con giống nhau dài nhất. Sau đó ta điền khuyết những kí tự dư ra bên a vào bên b và ngược lại.

Ta thực hiện như sau:

- + Tìm xâu con chung dài nhất của hai xâu;
- + Thêm các kí tự dư của a và b vào xâu con chung để thu được xâu kết quả.

XKT.BÀI 20. Giải mã

Trong giờ bài tập hóa học, thay vì cân bằng các phương trình phản ứng, Bòm lại viết một đoạn mã bí mật lên một mảnh giấy và gửi cho Cuội. Cách tạo ra mã bí mật như sau: nếu gặp các nguyên âm *a, e, o, u, i* trong từ nào đó thì Bòm viết thêm chữ cái *p* đằng sau nguyên âm, rồi thêm chính nguyên âm đó đằng sau chữ cái. Chẳng hạn, khi gặp nguyên âm *e*, Bòm sẽ viết *epe*; như thế từ *welcome* sẽ được mã hóa thành từ *wepelcopomepe*. Cuội nhận được mảnh giấy và lắc đầu không hiểu Bòm viết gì. Bạn hãy lập chương trình giúp cuội giải mã đoạn thư trên.

Dữ liệu vào: cho trong tệp **DECODE.INP** gồm một dòng duy nhất ghi xâu đã được mã hóa, độ dài xâu không quá 255, xâu chỉ bao gồm các chữ cái in thường của tiếng Anh và kí tự trắng.

Kết quả ra: ghi ra tệp **DECODE.OUT** gồm một dòng duy nhất ghi xâu kết quả giải mã.

Ví dụ

Input	Output
<i>wepelcopomepe</i>	<i>welcome</i>

Ý tưởng:

Duyệt xâu đã mã hóa, nếu kí tự thứ *i* không thuộc tập kí tự *a, e, o, u, i* thì ta thêm kí tự đó vào xâu *S1* và tăng *i* lên 1, ngược lại thì ta thêm kí tự thứ *i* vào xâu *S1* và tăng *i* lên 3 đơn vị

XKT.BÀI 21. Rút gọn xâu

Cho một xâu *S* chỉ gồm các chữ cái in thường với độ dài tối đa 250 kí tự. Em hãy viết chương trình để tạo ra xâu *SG* từ xâu *S* bằng cách xóa các kí tự liên tiếp giống nhau trong xâu *S* và chỉ để lại một kí tự đại diện trong đoạn đó.

Dữ liệu vào: Đọc từ file văn bản **RGXAU.INP** chứa xâu *S* chỉ gồm các chữ cái in thường.

Kết quả: Ghi vào file văn bản **RGXAU.OUT** là xâu *SG* tìm được.

Ví dụ:

RGXAU.INP	RGXAU.OUT
hheeeeeeeelloooo	helo

Thuật toán

- Đọc dữ liệu từ tệp;
- Duyệt qua các kí tự của xâu, mỗi loại kí tự chỉ ghi vào tệp **RGXAU.OUT** 1 lần.

XKT.BÀI 22. Xâu nghịch đảo

Hãy sử dụng kỹ thuật đệ quy trong lập trình để tìm xâu nghịch đảo của một xâu nhị phân cho trước (xâu nhị phân là xâu chỉ gồm hai kí tự '0' và '1').

Input: tệp **XAU.INP** gồm một dòng chứa xâu nhị phân.

Output: tệp **XAU.OUT** chứa xâu nghịch đảo tìm được.

Ví dụ:

XAU.INP	XAU.OUT
100111010	011000101

XKT.BÀI 23. Tính số lần lặp lại của xâu s1 trong xâu s2

Cho trước hai xâu kí tự *S1* và *S2*. Viết chương trình tính số lần lặp lại của xâu *S1* trong xâu *S2*.

Dữ liệu: Vào từ tệp văn bản **XAU.INP** gồm:

Dòng đầu tiên chứa xâu *S1*.

Dòng thứ hai chứa xâu *S2*.

Kết quả: Ghi ra tệp văn bản **XAU.OUT**:

Chỉ một dòng duy nhất ghi số lần lặp lại của xâu *S1* trong xâu *S2*.

Ví dụ:

XAU.INP	XAU.OUT
aba bababababa	4

Thuật toán

Chú ý: không sử dụng cách tìm vị trí xâu s1 trong xâu s2 rồi xóa các ký tự S1 trong S2 vì sẽ làm sai kết quả bài toán.

VD: s1=1231; s2=1231231;

Cách 1: tìm vị trí xuất hiện của S1 trong S2 là P. Xóa một ký tự tại vị trí P trong S2. Mỗi lần tìm thấy ta ghi nhận lại số lần xuất hiện.

Cách 2: Dùng một xâu S3 làm trung gian.

Lần lượt duyệt xâu S2

Với mỗi ký tự thứ i ta sẽ tạo xâu S3 có số lượng ký tự bắt đầu tại i và độ dài bằng i+ length(S)-1.

Nếu S3=S1 thì tăng biến đếm để ghi nhận số lần xuất hiện.

XKT.BÀI 24. Xâu con chung dài nhất

(Dạng bài toán tìm dãy con chung dài nhất)

Xâu ký tự X được gọi là xâu con của xâu ký tự Y nếu ta có thể xóa đi một số ký tự trong xâu Y để được xâu X. Cho biết hai xâu ký tự A và B, hãy tìm xâu ký tự C có độ dài lớn nhất và là con của cả A và B.

Dữ liệu vào: cho trong tệp **XAU_MAX.INP** gồm hai dòng, dòng thứ nhất chứa xâu A, dòng thứ hai chứa xâu B.

Dữ liệu ra: ghi ra tệp **XAU_MAX.OUT** gồm hai dòng:

+ Dòng 1: chứa số nguyên là độ dài lớn nhất của xâu con tìm được.

+ Dòng 2: chứa xâu con chung dài nhất của hai xâu A và B.

Ví dụ:

XAU_MAX.INP	XAU_MAX.OUT
abc1def2ghi3 abcdefghi123	10 abcdefghi3

Thuật toán: giải bằng phương pháp quy hoạch động

Định nghĩa xâu con của một xâu: một xâu X có độ dài l là xâu con của xâu Y có độ dài m khi có một dãy c sao cho $1 \leq c[1] \leq c[2] \leq c[3] \leq \dots \leq c[i] \leq \dots \leq c[l] \leq m$, và $X[c[i]] = Y[c[i]]$. Nói một cách khác, từ xâu gốc ta xóa đi các ký tự tại các vị trí tùy ý, các ký tự còn lại được dồn lại kế nhau và vẫn bảo toàn thứ tự.

Ví dụ: Có xâu X: abcdf. Các xâu con có thể là a, ab, bc, bd, dfr, abcdf, ... Các xâu không phải xâu con là ba, cda, ...

Để giải quyết bài toán này đầu tiên, ta xét bài toán đơn giản hơn.

Bài toán 1: Có 2 xâu, một xâu có m ký tự, xâu còn lại không có ký tự nào. Với bài toán này dễ thấy độ dài xâu con chung dài nhất là 0 (không có xâu nào).

Bài toán 2: Có hai xâu, mỗi xâu có đúng một ký tự.

Ví dụ 2.1: xâu X: a, xâu Y: b. Dễ thấy độ dài xâu con chung dài nhất là 0, vì không có xâu nào thỏa mãn.

Ví dụ 2.2: xâu X: a, xâu Y: a. Dễ thấy độ dài xâu con chung dài nhất là 1, xâu con chung khi đó là xâu có một ký tự a.

Bài toán 3: Có hai xâu, xâu X là: abc1def2ghi3, xâu Y là: abcdefghi123. Làm thế nào để có đáp án?

Ta giả sử đã biết câu trả lời trong trường hợp xâu X có độ dài i-1 ký tự đầu tiên, xâu Y có độ dài j-1 ký tự đầu tiên, khi đó đáp án trong trường hợp xâu X có độ dài i ký tự đầu tiên, xâu Y có độ dài j ký tự đầu tiên thì sao?

Ta sử dụng mảng hai chiều $L[i, j]$ để lưu độ dài xâu con chung dài nhất của xâu X đến ký tự thứ i và xâu Y đến ký tự thứ j.

Giả sử: $L[i-1, j-1]$, $L[i-1, j]$, $L[i, j-1]$ đã biết, tính $L[i, j] = ?$.

Xâu X: $x_1x_2 \dots x_{i-1}x_i$

Xâu Y: $y_1y_2 \dots y_{j-1}y_j$

Ta có các trường hợp sau:

+ TH1: $x_i = y_j$, như vậy thì $L[i, j] = L[i-1, j-1] + 1$.

+ TH2: $x_i \neq y_j$, $L[i, j] = \max(L[i-1, j], L[i, j-1])$.

Tính như thế nào?

Trong bài này ta tính từng dòng, nghĩa là ta tính hết $L[i-1, j]$, thì sau đó ta mới tính $L[i, j]$, với mọi j. Trên một dòng muốn tính $L[i, j]$ thì trước hết ta phải tính $L[i, j-1]$.

Bài tập tương tự:

Xâu chung 2. Cho hai xâu x gồm m và y gồm n kí tự. Cần xóa đi từ xâu x, dx kí tự và từ xâu y, dy kí tự để thu được hai xâu giống nhau. *Hãy xác định giá trị nhỏ nhất của tổng dx+dy.*

Thuật toán:

$$k = F[m,n];$$

$$dx = \text{length}(x) - k;$$

$$dy = \text{length}(y) - k;$$

$$\Rightarrow dx + dy \rightarrow \max$$

(Đối với bài toán cho hai dãy số nguyên ta vẫn áp dụng thuật toán tương tự, thay vì xử lí các kí tự, ta sẽ xử lí từng phần tử của mảng)

XKT.BÀI 25. Đoạn chung

Hãy tìm chiều dài lớn nhất k trong số các đoạn chung của hai xâu x và y.

Thí dụ, x = "xabcxxab~~cd~~xd", y = "aybcyab~~cd~~dy" có chiều dài của đoạn chung dài nhất là 4 ứng với đoạn "abcd".

Input: tệp DCHUNG.INP gồm hai dòng:

Dòng 1 chứa xâu x;

Dòng 2 chứa xâu y;

Output: tệp DCHUNG.OUT chứa một số nguyên k.

Ví dụ:

DCHUNG.INP	DCHUNG.OUT
xabcxxab cd xd aybcyab cd dy	4

Thuật toán.

Cách 1. Dùng mảng hai chiều L[i,j] là chiều dài lớn nhất của hai đoạn giống nhau x[i-k+1..i] và y[j-k+1..j], sao cho k → max. Ta có,

- Nếu x[i] = y[j] thì L[i,j] = L[i-1,j-1] + 1;
- Nếu x[i] ≠ y[j] thì L[i,j] = 0.

Chiều dài đoạn con chung dài nhất sẽ là $\text{Max} \{ L[i,j] \mid 1 \leq i \leq \text{length}(x), 1 \leq j \leq \text{length}(y) \}$.

Cách 2. Ta cũng có thể sử dụng một mảng một chiều a và hai biến phụ v và t. Biến t lưu tạm giá trị trước khi tính của a[j]. Biến v lấy lại giá trị t để tính cho bước sau.

Độ phức tạp: O(m.n), m = length(x), n = length(y).

Các bài tương tự

1. Đoạn chung 2. Cho xâu x gồm m kí tự và xâu y gồm n kí tự. Tìm đoạn chung dài nhất của hai xâu này. Kết quả cho ra 4 giá trị dx, cx, dy, cy, trong đó x[dx..cx] = y[dy..cy] là hai đoạn tìm được.

Yêu cầu: Kết quả ghi ra file "DCHUNG.OUT" gồm một dòng chứa 4 số nguyên tương ứng với dx, cx, dy, cy. Mỗi số cách nhau ít nhất một dấu cách.

Bài toán mở rộng: yêu cầu ghi vào tệp "DCHUNG.OUT" các phần tử của đoạn chung.

Ví dụ:

DCHUNG.INP	DCHUNG.OUT
xabcxxab cd xd aybcyab cd dy	4 abcd

Thuật toán bài đoạn chung 2:

Khi phát hiện a[j] > kmax ta ghi nhận imax = i; jmax = j; kmax = k. Cuối thủ tục ta tính cx = imax;

dx = cx-kmax+1; cy = jmax; dy = cy-kmax+1.

*** Bài giải bài toán mở rộng:**

Yêu cầu: in ra độ dài đoạn chung, vị trí đầu và cuối của hai đoạn chung ở hai xâu x và y; và in ra đoạn chung. Ta vẫn xử lí như bài toán đoạn chung 2 và sau cùng ta chỉ thêm việc ghi kết quả đoạn chung.

2. Đoạn chung 3. Cho hai dãy số nguyên a gồm m và b gồm n phần tử. Xác định chiều dài lớn nhất k để hai dãy cùng chứa k phần tử liên tiếp như nhau: a[i] = b[j], a[i+1] = b[j+1], ..., a[i+k-1] = b[j+k-1].

Thuật toán: xử lí trên mảng số nguyên, cách làm tương tự bài đoạn chung.

XKT.BÀI 26. Nén và giải nén xâu (theo độ dài loạt)

Một xâu kí tự có thể được nén theo cách sau: Một xâu con gồm n kí tự giống nhau (n>1), chẳng hạn gồm n kí tự 'a' sẽ thành na.

Yêu cầu:

Với một xâu kí tự cho trước không có chữ số, hãy nén xâu đó theo cách đã nêu.

Dữ liệu vào: tệp **XAUNEN.INP** có 1 dòng chứa xâu cần nén (chỉ gồm chữ cái).

Kết quả ghi trong tệp **XAUNEN.OUT** gồm 1 dòng chứa xâu đã nén.

Ví dụ:

Input	Output
aaaabbcd	4a2bcd

Bài toán mở rộng: Viết chương trình giải nén xâu

Dữ liệu vào: tệp **XAUNEN1.INP** chứa xâu đã nén.

Dữ liệu ra: tệp **XAUNEN1.OUT** chứa xâu đã giải nén.

XKT.BÀI 27. Sắp xếp xâu theo số lượng kí tự trong xâu

Mỗi xâu kí tự St được lấy từ tập các kí tự 'a'...'z', '0'...'9' và có độ dài tối đa là 255 kí tự. Cho N xâu kí tự St ($0 < N \leq 200$).

Yêu cầu: Thực hiện sắp xếp N xâu kí tự St theo thứ tự không giảm của số lượng các kí tự chữ số có trong mỗi xâu St.

Dữ liệu vào: Cho trong file văn bản SAPXEP.INP có cấu trúc như sau:

- Dòng 1: Ghi số nguyên N.

- N dòng tiếp theo: Mỗi dòng ghi một xâu St.

Dữ liệu ra: Ghi ra file văn bản SAPXEP.OUT theo cấu trúc như sau:

- Ghi N dòng: Mỗi dòng ghi một xâu St, các xâu được ghi theo thứ tự đã sắp xếp.

Ví dụ:

SAPXEP.INP	SAPXEP.OUT
3	cb1
abc1x2y3z	1cd7hd
cb1	abc1x2y3z
1cd7hd	

XKT.BÀI 28. Kiểm tra chuỗi

Cho một tập tin văn bản có n dòng ($3 \leq n \leq 30000$), mỗi dòng là một chuỗi s có tối đa 255 kí tự, các kí tự $s[i] \in ('a', \dots, 'z')$, với $1 \leq i \leq \text{length}(s)$. Trong đó chỉ có duy nhất một chuỗi s có số lần xuất hiện là một số lẻ, các chuỗi khác có số lần xuất hiện là một số chẵn. Hãy tìm chuỗi S (có số lần xuất hiện là một số lẻ) đó.

Dữ liệu vào: từ file văn bản CHUOI.INP có cấu trúc như sau:

- Dòng đầu là một số nguyên dương N

- N dòng tiếp theo, mỗi dòng là một chuỗi kí tự S

Dữ liệu ra: đưa vào file văn bản CHUOI.OUT chứa chuỗi kí tự tìm được.

Ví dụ:

CHUOI.INP	CHUOI.OUT
7	rach gia
ha tien	
phu quoc	
rach gia	
chau thanh	
ha tien	
chau thanh	
phu quoc	

Cách 1. Sắp xếp xâu tăng dần theo thứ tự từ điển. Duyệt và đếm số lần xuất hiện mỗi xâu, khi gặp xâu có số lần xuất hiện lẻ thì ghi kết quả và ngừng duyệt.

Cách 2: sử dụng phép toán cộng loại trừ trên bit: xor

```
type xau = string;  
var f: text;  
A: array[1..30000] of Xau;  
s: array[1..255] of longint;  
n: longint;
```

```

procedure mo_file; // tương tự cách 1
procedure Xuli;
var i, j: longint;
s1: string;
begin
    assign(f, 'chuoioi.out');    rewrite(f);
    fillchar(s, sizeof(s), 0);
    for i:= 1 to n do
    begin
        s1:=a[i];
        for j:= 1 to length(s1) do
            s[j]:= s[j] xor ord(s1[j]);
        end;
    for i:= 1 to 255 do
        write(f, chr(s[i]));
    close(f);
end;
begin
    mo_file; xuli;
end.

```

XKT.BÀI 29.

XKT.BÀI 30. Đoạn con dài nhất

Cho chuỗi kí tự S gồm toàn các chữ cái in hoa (A...Z) với độ dài không vượt quá 255. Hãy tìm đoạn con các kí tự liên tiếp dài nhất sao cho không có kí tự nào xuất hiện nhiều hơn một lần. Trong trường hợp có nhiều hơn một đoạn con có cùng chiều dài dài nhất, hãy chỉ ra đoạn xuất hiện đầu tiên trong chuỗi S.

Dữ liệu: Vào từ văn bản **SUBSTR.INP** gồm một dòng duy nhất chứa chuỗi S.

Kết quả: Ghi ra file văn bản **SUBSTR.OUT** hai số nguyên P và L tương ứng là vị trí và chiều dài của đoạn con dài nhất tìm được (kí tự đầu tiên trong chuỗi có vị trí là 1).

Ví dụ:

SUBSTR.INP	SUBSTR.OUT
ABABCDAC	3 4

Ý tưởng

Cách 1.

- Ta duyệt qua lần lượt các kí tự của chuỗi.
 - Khi duyệt qua kí tự thứ i, ta sẽ tìm cách xây dựng chuỗi dài nhất không có kí tự nào xuất hiện hai lần và kết thúc tại vị trí i. Để xây dựng được chuỗi này, ta cần lưu trữ các thông tin sau đây:

- Vị trí bắt đầu của chuỗi dài nhất mà không có kí tự nào xuất hiện hai lần, ta kí hiệu là p.
- Mảng b['A'..'Z'] trong đó với kí tự c thì b[c] lưu vị trí xuất hiện cuối cùng của kí tự c.

Khi duyệt qua kí tự thứ i (kí hiệu là s[i]):

- Nếu vị trí xuất hiện cuối cùng của kí tự này không nhỏ hơn p, nghĩa là kí tự này sẽ xuất hiện 2 lần và làm cho chuỗi đang xét trở nên không hợp lệ. Khi đó ta cập nhật lại vị trí bắt đầu p bằng b[s[i]]+1.
- Ta cập nhật lại giá trị i cho b[s[i]] để đảm bảo ý nghĩa của mảng b
- Chuỗi đang xét sẽ bắt đầu từ vị trí p và kết thúc tại vị trí i, do đó chuỗi này có độ dài là i-p+1, ta dùng giá trị này so sánh với kết quả để tìm ra chuỗi dài nhất.

Đoạn chương trình sau đây thể hiện thuật toán:

```

p:=1;           {gán giá trị khởi tạo cho p: vị trí bắt đầu là 1}
{duyệt qua từng kí tự của chuỗi}
for i:=1 to length(s) do
begin
    if (b[s[i]]>=p) then {kí tự s[i] xuất hiện hai lần trong chuỗi đang xét!}
        p:=b[s[i]]+1; {cập nhật lại vị trí bắt đầu mới}
    b[s[i]]:=i;      {cập nhật lại vị trí xuất hiện của kí tự s[i]}
    if (i-p+1>l) then {i-p+1 là độ dài của chuỗi đang xét}
    begin
        {so sánh với độ dài lớn nhất l}
        luup:=p;      {lưu lại vị trí bắt đầu}
        l:=i-p+1;      {cập nhật lại độ dài lớn nhất}
    end;
end;

```

Cách 2.

Ý tưởng:

- Xét từng đoạn con bắt đầu tại vị trí i ($i=1,2,\dots, \text{length}(s)-1$)
 - + Dùng một chuỗi $S1$ để lưu lại đoạn con bắt đầu từ i . Với mỗi ký tự tại j ($S[j]$) ta chỉ thêm vào $S1$ nếu nó chưa xuất hiện trong $S1$, ngược lại ngắt vòng lặp và so sánh độ dài chuỗi $S1$ với độ dài lớn nhất đang có, nếu độ dài $S1$ lớn hơn độ dài lớn nhất thì ta thay đổi độ dài lớn nhất và ghi nhận lại vị trí bắt đầu là i .

XKT.BÀI 31. Mã hoá và giải mã văn bản

(câu 2, đề thi HSG tỉnh Lâm Đồng năm 2008-2009, lớp 12)

Bài toán sau mô tả một thuật toán mã hoá đơn giản

Tập hợp các chữ cái tiếng Anh bao gồm 26 chữ cái được đánh số thứ tự từ 0 đến 25 như sau:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Quy tắc mã hoá một ký tự như sau (lấy ví dụ ký tự Z):

- Tìm số thứ tự tương ứng của ký tự ta được 25
- Tăng giá trị số này lên 7 ta được 32
- Tìm số dư trong phép chia số này cho 26 ta được 6
- Tra ngược bảng chữ cái ta thu được G.

Ví dụ:

Sử dụng quy tắc trên để mã hoá dòng chữ TIN HOC thành APU OVJ

Sử dụng quy tắc trên để giải mã các dòng chữ JOBJ JHJ LT SHT IHP AOHA AVA thành

CHUC CAC EM LAM BAI THAT TOT

Hãy xây dựng 2 hàm mã hóa và giải mã. Viết chương trình cho phép người dùng có thể chọn để thực hiện một trong hai công việc là mã hóa hoặc giải mã. Yêu cầu người dùng nhập trực tiếp và báo kết quả trên màn hình.

* **Ý tưởng:** Ta dùng một mảng ký tự bang[0..25] để lưu các chữ cái trong bảng chữ cái (chú ý: vị trí 0 tương ứng với chữ A, ..., vị trí 25 tương ứng với chữ Z)

XKT.BÀI 32. Tính tỉ lệ chữ nguyên âm

(Câu 2 - đề thi HSG tỉnh Bạc Liêu năm học 2011-2012, bảng A, ngày 1)

Cho một văn bản chứa trong một text file. Bạn hãy viết chương trình tính tỉ lệ các nguyên âm có mặt trong văn bản theo thứ tự của bảng chữ cái. Định nghĩa các nguyên âm là: A, E, I, O, U, Y

Dữ liệu vào: file NGUYENAM.INP gồm nhiều dòng chứa các ký tự của văn bản

Dữ liệu ra: file NGUYENAM.OUT mỗi dòng ghi ký tự và tỷ lệ % (lấy đến 4 chữ số thập phân) của các nguyên âm.

Ví dụ:

NGUYENAM.INP	NGUYENAM.OUT
ANHEMHOATHUANHUONGYEUNAMMOIMAYMAN	A: 17.6471 % E: 5.8824 % I: 2.9412 % O: 8.8235 % U: 8.8235 % Y: 5.8824 %

Ý tưởng: Chỉ cần đếm số lần xuất hiện mỗi loại ký tự và tính tỉ lệ phần trăm (đây là một dạng bài cơ bản xử lý các ký tự)

XKT.BÀI 33. Chữ cái xuất hiện

(câu 3, HSG Thanh Hóa, 2011-2012)

Cho chuỗi St chỉ gồm các chữ cái. Tính số lần xuất hiện của chữ cái xuất hiện nhiều nhất trong chuỗi (không phân biệt chữ in hoa và in thường).

Dữ liệu vào: Từ file CHUCAL.INP gồm: Chuỗi St (độ dài ≤ 500 ký tự).

Kết quả: Ghi ra file CHUCAL.OUT gồm: Một dòng duy nhất là bội số chung nhỏ nhất của kết quả bài toán và 10^5 .

Thuật toán:

Gọi số lần xuất hiện chữ cái nhiều nhất trong chuỗi là a thì kết quả bài toán là $BCNN(a, 100000)$.

Vì thế cần xây dựng một hàm tính $UCLN(a, b)$ và $BCNN(a, b)$ để tìm ước chung lớn nhất và bội chung nhỏ nhất của hai số.

XKT.BÀI 34. Biến đổi chuỗi

(câu 1, HSG Thanh Hóa, 2010-2011)

Cho trước một chuỗi nhị phân có độ dài bất kỳ. Cần biến đổi chuỗi nhị phân này về dạng toàn số 0. Các phép biến đổi chỉ có thể là một trong các loại sau:

- Biến đổi xâu con 11 thành 00.
 - Biến đổi xâu con 010 thành 000.
- Hãy chỉ ra một cách biến đổi xâu đã cho thành xâu có toàn 0.
- Dữ liệu vào:** từ file BDXAU.INP xâu nhị phân độ dài bất kỳ.

Kết quả: ghi ra file BDXAU.OUT như sau:

- Dòng đầu tiên chứa xâu ban đầu.
- Sau đó mỗi dòng là một xâu tiếp theo sau một phép biến đổi. Xâu cuối cùng là xâu toàn 0.
- Nếu không biến đổi được thì ghi "Khong the bien doi duoc".

(Nếu có nhiều cách thì chỉ cần in ra một cách biến đổi)

Ví dụ

BDXAU.INP	BDXAU.OUT
11010011	11010011 00010011 00010000 00000000

BDXAU.INP	BDXAU.OUT
10101101	Khong the bien doi duoc

Ý tưởng

Trước khi biến đổi ta kiểm tra:

Nếu (`copy(S, 1, 2) = '10'`) or (`copy(S, length(s) - 2, 3) = '101'`) thì kết luận không thể biến đổi được ngược lại:

- Duyệt xâu, tìm vị trí của xâu '11' và thay thành '00' (lặp lại cho đến hết)
- Duyệt xâu, tìm vị trí của xâu '010' và thay thành '000' (lặp lại cho đến hết)

Mỗi lần duyệt ta ghi nhận xâu thay đổi vào một mảng một chiều.

Khi không thể thay đổi được nữa, ta kiểm tra xâu kết quả xem có đúng toàn số 0 hay không, nếu đúng thì ta ghi kết quả các xâu biến đổi được vào tệp kết quả.

Chú ý: phải lưu xâu ban đầu trước khi thực hiện biến đổi xâu.

XKT.BÀI 35. Chuẩn hóa văn bản (1)

(câu 1, HSG Cà Mau, 2013-2014)

Cho một đoạn văn bản chỉ gồm các chữ cái, các dấu cách và các dấu câu. Hãy sửa lại đoạn văn bản theo yêu cầu:

- + Không được có 2 dấu cách (khoảng trắng) đứng liền nhau.
- + Dấu câu ('.', '?', '!', '...', ',', ';', ':') phải đứng sau chữ cái và trước dấu cách.
- + Sau các dấu câu ('.', '?', '!', '...', ':') phải viết hoa.
- Dữ liệu vào trong tệp tin TEXT_RE.INP.
- Kết quả ghi trong tệp tin TEXT_RE.OUT.

XKT.BÀI 36. Chuẩn hóa văn bản (2)

(câu 3, HSG Thanh Hóa, 2010-2011)

Một văn bản được gọi là văn bản chuẩn nếu:

- Hai từ liền nhau có duy nhất một dấu cách trống.
- Dấu ngắt câu (dấu chấm, dấu phẩy, dấu chấm phẩy, dấu chấm hỏi, dấu chấm than) được đặt sát vào từ ngay trước nó, sau đó mới đến dấu cách trống.
- Dấu mở ngoặc đặt sát vào phía bên trái của từ bắt đầu mở ngoặc.
- Dấu đóng ngoặc đặt sát bên phải từ cuối cùng được đóng ngoặc.

Hãy viết chương trình để kiểm tra và đưa một đoạn văn bản về dạng văn bản chuẩn.

Dữ liệu vào: từ file VANBAN.INP

Kết quả: ghi ra file VANBAN.OUT văn bản đã được chuẩn hoá.

Ví dụ: (khi tạo tệp nguồn ta gõ không bỏ dấu tiếng Việt)

VANBAN.INP	VANBAN.OUT
Thấy rét u tôi bọc lại mền Cô nàng cất rượu ủ thêm men . (trích Hoa với rượu – Nguyễn Bính)	Thấy rét u tôi bọc lại mền Cô nàng cất rượu ủ thêm men. (trích Hoa với rượu – Nguyễn Bính)

XKT.BÀI 37. Sắp xếp xâu.

Người ta định nghĩa: Từ là một nhóm kí tự đứng liền nhau.

Cho một xâu St gồm các kí tự lấy từ tập 'a' .. 'z' và dấu cách. Xâu không quá 20 từ, mỗi từ dài không quá 10 kí tự.

Yêu cầu: Sắp xếp các từ của xâu kí tự theo thứ tự không giảm của độ dài các từ trong xâu St.

Dữ liệu vào: Cho trong file văn bản SAPXAU.INP, có cấu trúc:

- Dòng 1: Ghi một xâu kí tự St (có ít nhất 1 từ).

Dữ liệu ra: Ghi ra file văn bản SAPXAU.OUT, theo cấu trúc:

- Dòng 1: Ghi các từ của xâu kí tự sau khi được sắp xếp. Các từ được ghi cách nhau đúng một dấu cách.

Ví dụ:

SAPXAU.INP	SAPXAU.OUT
acb abcde abcd abc	acb abc abcd abcde

Ý tưởng:

- Dùng một mảng xâu kí tự để lưu các từ trong xâu ban đầu.
- Sắp xếp xâu tăng dần theo độ dài.
- Ghi mảng kết quả sau khi sắp xếp.

XKT.BÀI 38. Mã hóa Caesar (1)

(câu 1, tin học trẻ tỉnh Bình Định, 2010-thpt)

Trong mật mã học, mật mã Caesar (Xê - da), còn gọi là mật mã dịch chuyển, là một trong những mật mã đơn giản và được biết đến nhiều nhất. Mật mã Caesar là một dạng của mật mã thay thế, trong đó mỗi kí tự trong văn bản được thay thế bằng một kí tự cách nó một đoạn trong bảng chữ cái để tạo thành bảng mã.

Ví dụ, đối với bảng mã tiếng anh (ABCDEFGHIJKLMNOPQRSTUVWXYZ), nếu độ dịch là 3, A sẽ được thay bằng D, B sẽ được thay bằng E, ..., W sẽ thay bằng Z, X sẽ thay bằng A, Y sẽ thay bằng B và Z thay bằng C. **Yêu cầu:** cho một chuỗi kí tự S gồm các chữ cái in hoa và dấu cách, một số nguyên dương k ($0 \leq k \leq 26$). Hãy tìm chuỗi kí tự T đã được mã hóa theo phương pháp trên.

Dữ liệu vào: Cho trong file **CAESAR.INP** gồm nhiều dòng

Dòng đầu tiên là một chuỗi kí tự có độ dài tối đa 80 kí tự.

Các dòng sau, mỗi dòng ghi một số nguyên k

Dữ liệu ra: ghi vào file **CAESAR.OUT** gồm nhiều đoạn phân cách nhau bởi dấu *. Mỗi đoạn ghi chuỗi T tương ứng với khóa k trong file **caesar.inp**.

Ý tưởng:

- Mỗi kí tự ta sẽ xác định vị trí của nó trong bộ mã ASCII là $\text{ord}(S[i]) - \text{ord}('A')$
- Khi dịch chuyển k vị trí ta sẽ xác định vị trí mới là: $(\text{ord}(S[i]) - \text{ord}('A') + k) \bmod 26$
- Khi đã có vị trí mới ta chỉ cần ghi kí tự tại vị trí đó vào tệp kết quả.

Mở rộng bài toán: Hãy viết chương trình giải mã.

XKT.BÀI 39. Mã hóa Caesar (2)

Đồng dư có nhiều ứng dụng trong toán học và trong Tin học, một trong những ứng dụng của phép đồng dư liên quan đến là mật mã học, một lĩnh vực nghiên cứu các thư tín bí mật. Một trong số những người sử dụng mật mã được biết sớm nhất là Julius Caesar. Ông đã làm cho các bức thư trở nên bí mật bằng cách dịch mỗi chữ cái đi ba chữ cái về phía trước trong bảng chữ cái (và ba chữ cái cuối cùng thành ba chữ cái đầu tiên). Ví dụ, theo sơ đồ đó, chữ B được chuyển thành chữ E và chữ X được chuyển thành chữ A. Để biểu diễn quá trình mã hóa của Caesar một cách toán học: Trước hết thay mỗi chữ cái bằng một số nguyên từ 0 đến 25, dựa vào vị trí của nó trong bảng chữ cái. Ví dụ: A thay bằng 0, K bằng 10 và Z bằng 25. Phương pháp này có thể được biểu diễn bởi hàm f, hàm này gán cho số nguyên không âm p, $p \leq 25$, số nguyên $f(p)$ trong tập $\{0, 1, 2, \dots, 25\}$ sao cho $f(p) = (p+3) \bmod 26$. Như vậy, trong phiên bản mã hóa của bức thư, chữ cái được biểu diễn bởi p sẽ được thay bằng chữ cái biểu diễn bởi $(p+3) \bmod 26$.

Ví dụ: Chuyển nội dung “DO NOT PASS GO” thành bức thư bí mật.

Thay các chữ cái trong nội dung trên thành số, ta được:

D	O		N	O	T		P	A	S	S		G	O
3	14		13	14	19		15	0	18	18		6	14

Bây giờ, thay các số p đó bằng $f(p) = (p+3) \bmod 26$ ta được:

D	O		N	O	T		P	A	S	S		G	O
3	14		13	14	19		15	0	18	18		6	14
6	17		16	17	22		18	3	21	21		9	17

Dịch ngược trở lại các chữ cái, ta được nội dung đã mã hóa:

D	O		N	O	T		P	A	S	S		G	O
3	14		13	14	19		15	0	18	18		6	14
6	17		16	17	22		18	3	21	21		9	17
G	R		Q	R	W		S	D	V	V		J	R

Tuy nhiên, trong thực tế, mã hóa Caesar không có độ an toàn cao, có nhiều cách để nâng cao độ an toàn của phương pháp này. Một trong những phương pháp đó là dùng hàm có dạng $f(p) = (ap+k) \bmod 26$, trong đó, a và k nguyên.

Yêu cầu: Dùng mã hóa Caesar để mã hóa nội dung bức thư chỉ gồm dấu cách và các chữ cái in hoa thành bức thư có nội dung bí mật.

Dữ liệu vào: Dòng đầu ghi hai số a ($1 \leq a \leq 1000000$) và k ($1 \leq k \leq 1000000$), dòng hai ghi một xâu chỉ bao gồm các chữ cái và dấu cách là nội dung bức thư cần mã hóa.

Dữ liệu ra: Một dòng duy nhất chứa nội dung đã được mã hóa.

Ví dụ:

Input	Output
1 3 DO NOT PASS GO	GR QRW SDVV JR

Ý tưởng

Cách 1. Ta thực hiện tương tự bài “Bài toán mật mã Ceasar”, nhưng chú ý dữ liệu a và k rất lớn nên ta phải sử dụng phép toán mod như sau:

```
a:=a mod 26;
k:=k mod 26;
```

Cách 2.

- Sử dụng một mảng $\text{CH}[0..25]$ để lưu bảng chữ cái từ ‘A’,...,‘Z’.
- Tính:

```
a:=a mod 26;
k:=k mod 26;
```

- Duyệt từng kí tự của xâu S , tính vị trí P của kí tự $S[i]$ sau khi dịch chuyển k vị trí thông qua hàm $f(p)=(ap+k) \bmod 26$ như sau:

```
P:= ord(S[i])-ord('A'); {vị trí ban đầu của S[i] trong mảng CH}
P:= (a*p+k) mod 26; {vị trí mới của S[i] trong mảng CH}
S1:=S1+CH[p]
```

XKT.BÀI 40. Mã hoá hồ sơ

Để quản lý tốt các hồ sơ trong kỳ thi tuyển sinh, hội đồng tuyển sinh trường PTNK đã quyết định đánh số các hồ sơ theo một phương pháp khoa học. Mã hồ sơ của thí sinh là một chuỗi gồm 10 chữ số.

Tuy nhiên không phải bất kỳ chuỗi 10 chữ số nào cũng là mã hồ sơ hợp lệ bởi vì hội đồng tuyển sinh đưa ra một quy định ràng buộc chặt chẽ cho các chữ số đó. Nếu $M=a_1a_2..a_{10}$ là một mã hồ sơ thì M phải thỏa mãn ràng buộc:

Nếu đặt $S(M)=1a_1+2a_2+3a_3+...+10a_{10}$ thì $S(M)$ phải là một số chia hết cho 11.

Nhờ quy định này, trong những trường hợp do sơ xuất có một chữ số trong mã hồ sơ bị mờ, không đọc được thì ta vẫn có thể xác định được giá trị của nó. Ví dụ như: (quy ước ? là chữ số bị mờ):

- Với $M=00000000?1$ thì có thể suy ra chữ số bị mờ là 5 vì theo ràng buộc, để $S(M)$ là một số chia hết cho 11 nó chỉ có thể có giá trị là 55.
- Tương tự với $M=00000001?1$ thì có thể suy ra chữ số bị mờ là 9.
- Tương tự với $M=00722?0858$ thì có thể suy ra chữ số bị mờ là 6.

Yêu cầu: Hãy viết chương trình giúp hội đồng tuyển sinh suy ra được chữ số bị mờ trong mã hồ sơ.

Dữ liệu: Vào từ file văn bản ENCODE.INP có chứa mã hồ sơ có 1 chữ số bị mờ được thay bằng dấu chấm hỏi.

Kết quả: Ghi ra file văn bản ENCODE.OUT chứa giá trị của chữ số bị mờ trong mã hồ sơ đã cho.

Ví dụ:

ENCODE.INP	ENCODE.OUT
00000000?1	5
00000001?1	9
00722?0858	6

Ý tưởng

- Ta thử thay từng chữ số từ 0 đến 9 vào vị trí “?” và tính xem điều kiện $S(M)$ chia hết cho 11 có được thỏa mãn hay không như đoạn chương trình sau đây:

```
for i:=1 to 10 do
  {r là tổng các số hạng của S(M) trừ vị trí có dấu '?'}
  if (s[i]<>'?') then r:=(r+i*(ord(s[i])-ord('0')) mod 11;
  {ord(s[i])-ord('0'): xác định giá trị số s[i]}
  i:=pos('?',s); {i là vị trí của dấu '?'}
  for d:=0 to 9 do {duyệt qua tất cả các chữ số từ 0 đến 9}
    if (r+i*d) mod 11 = 0 then {kiểm tra xem S(M) có chia hết cho 11}
      begin
        writeln(d); {d chính là chữ số cần tìm}
        break {thoát khỏi vòng lặp}
      end;
```

XKT.BÀI 41. Tìm từ đầu tiên dài nhất trong xâu*(Câu 1 - đề thi HSG tỉnh Bạc Liêu năm học 2011-2012, bảng B, ngày 2)*

Cho xâu khác rỗng. Tìm từ đầu tiên dài nhất trong xâu. (Từ là một dãy kí tự liên tiếp không chứa dấu cách).

- Dữ liệu vào: từ tệp **TIMTU.INP** gồm một dòng chứa xâu s.

- Dữ liệu ra: Ghi ra tệp **TIMTU.OUT** gồm 1 dòng chứa câu trả lời: “Tu dau tien dai nhat trong xau la: a”.

(Với a là từ đầu tiên dài nhất trong xâu s)

Ví dụ:

TIMTU.INP	TIMTU.OUT
Hoc tin rat thu vi	Tu dau tien dai nhat trong xau la: Hoc

Ý tưởng:

- Nếu bài toán cho xâu chưa chuẩn thì ta phải viết chương trình con chuẩn hóa xâu rồi mới xử lí.
- Ta tách các từ và tính độ dài, sau đó so sánh các từ để tìm từ dài nhất.

XKT.BÀI 42: Chiếc nón diệu kì

Một lần trong chương trình “Chiếc nón diệu kì”, ở phần chơi dành cho khán giả, thay vì đoán chữ như mọi khi, người dẫn chương trình tự mình quay “chiếc nón” và cho hiện lên màn hình trước mặt khán giả trong trường quay các số trong các ô mà kim chỉ thị lần lượt đi qua. “Chiếc nón” quay đúng một số nguyên vòng, nên trong dãy số hiện lên màn hình, số cuối cùng trùng với số đầu tiên. Sau đó, người dẫn chương trình mời một khán giả ở cuối trường quay (chỉ nhìn thấy màn hình mà không nhìn thấy “chiếc nón”) cho biết chiếc nón có tối thiểu bao nhiêu ô?

Yêu cầu: Hãy trả lời câu hỏi của người dẫn chương trình.

Dữ liệu: Vào từ tệp tin văn bản CNDK.INP gồm hai dòng:

+ Dòng 1 ghi số N là số lượng số đã hiện lên màn hình, ($2 \leq N \leq 100$).

+ Dòng 2 ghi lần lượt N số, mỗi số có giá trị không quá 32000.

Kết quả: Ghi ra tệp tin văn bản CNDK.OUT số ô tối thiểu của “chiếc nón”.

Lưu ý: Các số trên cùng một dòng cách nhau ít nhất một khoảng trắng.

Ví dụ:

CNDK.INP	CNDK.OUT
13 5 3 1 3 5 2 5 3 1 3 5 2 5	6

Ý tưởng: Bài này thực chất là bài toán kiểm tra dãy số tuần hoàn dài nhất trong một dãy đã cho.

XKT.BÀI 43. Xâu nhị phân*(Câu 3, đề thi HSG tỉnh Đồng Tháp năm học 2012-2013)*

Xét các xâu nhị phân có độ dài N được thành lập như sau:

+ Bắt đầu là xâu gồm N bit 0.

+ Xâu nhị phân tiếp theo được tạo thành từ xâu nhị phân trước đó bằng cách tìm bit 0 đầu tiên tính từ phải sang trái đổi thành bit 1 và đổi tất cả các bit bên phải bit vừa thay đổi đó thành bit 0. Ví dụ xâu tiếp theo của xâu 0100111 thành xâu 0101000

Lặp lại cho đến khi N bit đầu là 1.

Ví dụ với N=3 ta có:

000 -> 001 -> 010 -> 011 -> 100 -> 101 -> 110 -> 111

Sắp xếp các xâu N bit này theo thứ tự từ điển và đánh số thứ tự từ 0 đến hết.

Thứ tự từ điển được tính như sau:

Với hai xâu A và B có độ dài N:

+ Xâu A được gọi là nhỏ hơn xâu B nếu như bit khác nhau đầu tiên tính từ phải sang trái của xâu A nhỏ hơn xâu B.

+ Xâu A và B được gọi là bằng nhau nếu như tất cả các bit tương ứng đều giống nhau.

Yêu cầu: Cho hai số tự nhiên N và K ($3 \leq N \leq 1000$, $0 \leq K \leq 2^N$). Hãy tìm xâu nhị phân có độ dài N có thứ tự từ điển là K

Dữ liệu vào: cho từ file văn bản BIN.INP gồm hai dòng:

+ Dòng đầu ghi số N.

+ Dòng thứ hai ghi số K.

Kết quả: ghi ra file văn bản BIN.OUT gồm dãy N bit tương ứng là xâu nhị phân có độ dài N có thứ tự từ điển là K.

Ví dụ

BIN.INP	BIN.OUT
3 2	010

XKT.BÀI 44. Tìm số nhỏ nhất lớn hơn X có cùng chữ số với X

Cho trước số nguyên dương $X (X < 10^6)$. Tìm số nguyên nhỏ nhất lớn hơn X có cùng các chữ số với X.
Dữ liệu vào: cho trong file văn bản SNNN.INP gồm một dòng chứa số X.
Kết quả ra: ghi ra file văn bản SNNN.OUT trên một dòng số tìm được, nếu không tìm được ghi số 0

Ý tưởng

Thực chất của bài toán dạng này chính là tìm hoán vị sát sau của một xâu S.

Để tổng quát, ta xây dựng hàm NEXT(n) để tìm số tự nhiên nhỏ nhất lớn hơn x có cùng số chữ số với x như sau (lưu số x vào xâu S):

Bước 1.

Tìm điểm gãy: Tìm ngược từ $s[n]$ trở về trước đến vị trí i đầu tiên thoả điều kiện $s[i] < s[i+1]$.

Nếu không tìm được i tức là s là hoán vị lớn nhất. Đặt trị **false** cho hàm **Next** và dừng thuật toán.

Next = false có nghĩa là không tồn tại hoán vị sát sau hoán vị s hay s là hoán vị lớn nhất.

- Nếu tìm được: thực hiện bước 2.

Bước 2.

Bước 2.1 Tìm điểm vượt: Tìm ngược từ $s[n]$ trở về trước đến vị trí j đầu tiên thoả điều kiện $s[j] > s[i]$.

Bước 2.2. Đổi chỗ $s[i]$ với $s[j]$.

Bước 2.3. Lật: Đảo lại trật tự của dãy $s[i+1..n]$ ta sẽ thu được hoán vị đứng sát sau hoán vị s .

Bước 3. Đặt trị true cho hàm **Next**. **Next = true** có nghĩa là tìm được hoán vị sát sau hoán vị s .

Chú ý: Khi khởi trị hoán vị đơn vị ta sử dụng phần tử $s[0] = '0'$ làm lính canh, cần xác định độ dài xâu trước khi gán $s[0] = '0'$. Nhờ vậy, khi duyệt ngược để tìm điểm gãy ta không phải kiểm tra giới hạn mảng. Thay vì viết

```
i := n-1;  
while (i > 0) and (s[i] >= s[i+1]) do i := i-1;
```

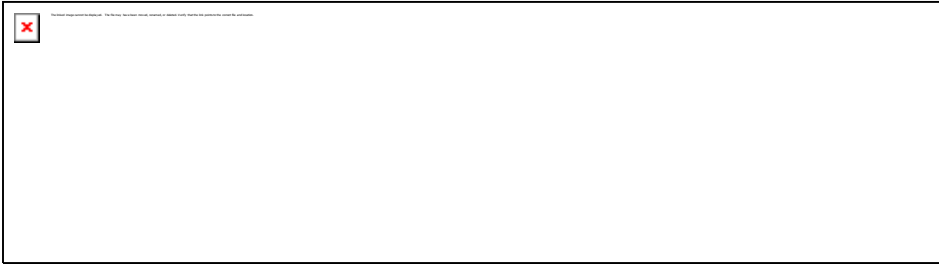
ta chỉ cần viết

```
i := n-1;  
while (s[i] >= s[i+1]) do i := i-1;
```

Hàm Next được mô tả như sau:

```
function Next(n: integer): Boolean;  
var i, j: integer;  
    t: char;  
begin  
    Next := false; i := n-1;  
    while (s[i] >= s[i+1]) do i := i-1;  
    if i = 0 then exit;  
    { s[i] < s[i+1], i là điểm gãy }  
    j := n; { Tìm điểm vượt a[j] > a[i] }  
    while (s[j] < s[i]) do j := j-1;  
    { Đổi chỗ s[i] , s[j] }  
    t := s[i]; s[i] := s[j]; s[j] := t;  
    { Lật s[i+1..n] }  
    i := i+1; j := n;  
    while i < j do  
    begin  
        t := s[i];  
        s[i] := s[j];  
        s[j] := t;  
        i := i+1;  
        j := j-1;  
    end;  
    Next := true;  
end;
```

Thí dụ, với $n = 8$, giả sử ta đã ghi được hoán vị $s = 74286531$, khi đó hoán vị sát sau s sẽ được xây dựng như sau:



Chương trình

```
var s: string;
procedure nhap;
Var f: text;
begin
    assign(f,'inp.inp'); reset(f);
    readln(f,s);
    close(f);
end;
function Next(n: integer): Boolean;
var i, j: integer;
    t: char;
begin
    Next := false; i := n-1;
    while (s[i] >= s[i+1]) do i:= i-1;
    if i = 0 then exit;
    {s[i] < s[i+1], i là điểm gãy}
    j := n; {Tìm điểm vượt a[j] > a[i]}
    while (s[j] < s[i]) do j := j-1;
    {Đổi chỗ s[i] , s[j]}
    t:= s[i]; s[i]:= s[j]; s[j]:= t;
    {Lật s[i+1..n]}
    i:= i+1; j:= n;
    while i < j do
    begin
        t:= s[i];
        s[i]:= s[j];
        s[j]:= t;
        i:= i+1;
        j:= j-1;
    end;
    Next:= true;
end;
procedure xuli;
var n: byte;
    flag: boolean;
F: text;
begin
    assign(f,'out.out'); rewrite(f);
    n:=length(s);
    S[0]:='0';
    flag:= next(n);
    if flag then write(f,s)
    else write(f,'0');
    close(f);
end;
begin
    nhap;
    xuli;
End.
```

*** Nếu đề bài yêu cầu sinh ra tất cả các số sát sau của X thì ta lặp lại cho đến khi hàm Next(x) cho giá trị *false*.**

CHUYÊN ĐỀ: HOÁN VỊ, CHỈNH HỢP, TỔ HỢP

HVCH. BÀI 1. Hoán vị (123...n)

(Câu 1, HSG Đà Nẵng 2001-2002, vòng 1)

Nhập vào một số tự nhiên N. Lập chương trình đưa ra màn hình tất cả các hoán vị của N số 1,2,3,...,N. (N<10)

Ý tưởng:

Vận dụng thuật toán sinh các chỉnh hợp lặp chập k của N phần tử (trong bài này k=n) ta sẽ sinh ra được tất cả các hoán vị cần tìm.

HVCH. BÀI 2. Liệt kê các hoán vị của một xâu

(trích bài tập 4.3 tài liệu giáo khoa chuyên tin quyển 1)

Cho xâu S chỉ gồm các kí tự 'A' đến 'Z', các kí tự đôi 1 khác nhau. Hãy liệt kê các hoán vị khác nhau của xâu S

Dữ liệu vào: nhập từ file HVX.INP gồm 1 dòng duy nhất là xâu S

Dữ liệu ra: Ghi ra file HVX.OUT gồm nhiều dòng, mỗi dòng là 1 hoán vị của xâu S

Ý tưởng

Bài này ta cũng vận dụng thuật toán như bài "HVCH.Bài 1" nhưng ở đây áp dụng trên kiểu kí tự.

HVCH. BÀI 3. Liệt kê xâu

(trích bài tập 4.4 tài liệu giáo khoa chuyên tin quyển 1)

Cho số nguyên dương n ($n \leq 20$). Hãy liệt kê tất cả các xâu độ dài n chỉ gồm 2 kí tự 'A' và 'B' sao cho không có 2 kí tự 'B' nào đứng cạnh nhau.

Dữ liệu vào: nhập từ file dayab.INP gồm 1 dòng duy nhất ghi số nguyên dương n

Dữ liệu ra: Ghi ra file DAYAB.OUT gồm nhiều dòng, mỗi dòng là 1 xâu độ dài n tìm được.

Ý tưởng

Vận dụng thuật toán sinh các chỉnh hợp lặp chập k của n phần tử, trước khi ghi kết quả ta cần kiểm tra điều kiện không có 2 kí tự 'B' đứng cạnh nhau có thỏa hay không.

HVCH. BÀI 4. Tạo sơn tổng hợp

Từ N loại sơn ban đầu có số hiệu là 1, 2, ..., N ($1 \leq N \leq 9$), người ta có thể tạo ra rất nhiều loại sơn tổng hợp khác nhau bằng cách trộn lẫn một số loại sơn nào đó lại với nhau theo một liều lượng nào đó của mỗi loại. Khi tham gia trộn để được một loại sơn tổng hợp nào đó, các loại sơn khác nhau được đưa vào từ các vị trí khác nhau và liều lượng của mỗi loại sơn là bao nhiêu phụ thuộc vào thứ tự vị trí đưa vào của loại sơn đó. Liều lượng của mỗi loại sơn mà khác nhau trong khi trộn thì cho ra các loại sơn tổng hợp khác nhau. Hãy liệt kê ra tất cả các phương án trộn M loại sơn ($M \leq N$) trong N loại sơn đã cho để có được các loại sơn tổng hợp.

Dữ liệu vào: Giá trị của các số N, M được đưa vào từ bàn phím.

Dữ liệu ra là tệp văn bản PAINT.OUT có cấu trúc: Mỗi dòng ghi số hiệu của M loại sơn theo thứ tự khi đưa vào trộn để tạo ra một loại sơn tổng hợp nào đó. Dòng cuối cùng ghi số lượng các loại sơn tổng hợp tạo ra.

Ví dụ:

	PAINT.OUT
N=3	1 2
M=2	1 3
	2 1
	2 3
	3 1
	3 2
	6

Ý tưởng

Bài này chính là bài toán sinh ra số chỉnh hợp chập M của N phần tử.

HVCH. BÀI 5. Trộn đề

(câu 2, đề thi HSG tỉnh Lâm Đồng năm 2008-2009, lớp 12)

Một giáo viên ra đề cho m ($4 \leq m \leq 50$) câu hỏi trắc nghiệm. Giáo viên muốn trộn ngẫu nhiên m câu hỏi đó thành n ($2 \leq n \leq 30$) đề khác nhau (không xét đến phương án trộn các câu gợi ý trả lời bên trong từng câu hỏi).

Viết chương trình đọc nội dung tập tin **degoc.txt** và trộn ngẫu nhiên các câu hỏi bên trong tập tin đó, mỗi phương án trộn được sẽ cho ra một đề mới.

Cho xuất n phương án trộn được ra n tập tin **de_1.txt, de_2.txt, ..., de_n.txt**

Đề gốc cho trong tập tin **degoc.txt** với hàng 1 là số m và n (cách nhau một dấu cách), m hàng tiếp theo là nội dung m câu hỏi (mỗi hàng độ dài không quá 255 kí tự).

Thí dụ: degoc.txt cho 8 câu hỏi và 4 đề

8 4

Noi dung cau hoi 1

Noi dung cau hoi 2

.....

Noi dung cau hoi 7

Noi dung cau hoi 8

Ý tưởng:

Áp dụng thuật toán tìm các chỉnh hợp không lặp chập n của m phần tử, ta đánh số thứ tự các câu hỏi rồi giải tương tự bài chỉnh hợp không lặp.

CHUYÊN ĐỀ: MẢNG MỘT CHIỀU, HAI CHIỀU

MIC2C. BÀI 1. Tần số

Cho dãy số nguyên dương, số lần xuất hiện của một số được gọi tần số của số nguyên đó. Hãy tìm số nguyên dương có tần số cao nhất và tần số tương ứng của nó.

Dữ liệu vào: Cho từ file văn bản MAX.INP bao gồm:

Dòng đầu tiên chứa một số nguyên N ($1 \leq N \leq 10000$) là số lượng các số nguyên trong dãy.

Mỗi dòng trong N dòng tiếp theo chứa số nguyên M ($1 \leq M \leq 1000$) trong dãy.

Kết quả: Ghi ra file văn bản MAX.OUT, gồm 2 số nguyên viết trên một dòng, số thứ nhất ghi số nguyên có tần số cao nhất, số thứ 2 là tần số của nó (trong trường hợp có nhiều số nguyên có tần số cao nhất bằng nhau, hãy đưa ra số nguyên nhỏ nhất và tần số của nó). Hai số cách nhau một kí tự trắng.

MAX.INP	MAX.OUT
9	2 3
1	
2	
5	
6	
3	
7	
2	
5	
2	

MAX.INP	MAX.OUT
7	2 2
2	
4	
6	
7	
7	
2	
4	

Ý tưởng: yêu cầu của bài này là tìm số có số lần xuất hiện nhiều nhất, đây là dạng bài cơ bản, gv hướng dẫn học sinh đếm số lần xuất hiện từng giá trị rồi tìm ra giá trị max.

MIC2C. BÀI 2: Số may mắn

Cho số nguyên dương N và số nguyên dương K ($0 < N \leq 2 \times 10^9$; $1 \leq K \leq 82$). M là tổng giá trị các chữ số của N. Số N được gọi là “Số may mắn” nếu M chia hết cho K. Ví dụ: Cho số N = 234 và số K = 3 ta có N là “Số may mắn” vì N có tổng các chữ số là $M = 2 + 3 + 4 = 9$ chia hết cho 3.

Cho 2 số nguyên dương P và Q ($0 < P \leq Q \leq 2 \times 10^9$).

Yêu cầu: Hãy đếm số lượng các “Số may mắn” nằm trong đoạn [P, Q].

Dữ liệu vào: Cho trong file văn bản SOMM.INP có cấu trúc như sau:

- Dòng 1: Ghi 3 số P Q K, các số được ghi cách nhau ít nhất một dấu cách.

Dữ liệu ra: Ghi ra file văn bản SOMM.OUT theo cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương D là số lượng các “Số may mắn” tìm được.

Ví dụ:

SOMM.INP	SOMM.OUT
1 30 3	10

SOMM.INP	SOMM.OUT
10 30 10	2

Ý tưởng

- Viết hàm kiểm tra số may mắn như định nghĩa của đề bài.
- Duyệt từ P đến Q để tìm đếm số lượng số may mắn.

M1C2C. BÀI 3. Mã số nhân viên

Để tiện trong công tác quản lý, một công ty đã gán mã số cho các nhân viên của mình. Mã số của mỗi nhân viên bao gồm bảy chữ số (gọi là phần đầu của mã nhân viên) và một chữ cái in hoa nổi phía sau. Chữ cái này được tính toán từ bảy chữ số phần đầu của mã theo các bước như sau:

1. Chữ số thứ i được gán một số nguyên dương a_i được gọi là trọng lượng của nó (thứ tự các chữ số trong phần đầu của mã tính từ trái qua phải). Người ta lấy lần lượt các chữ số trong mã nhân với trọng lượng tương ứng của nó.
2. Cộng tất cả các kết quả.
3. Tìm số dư của phép chia tổng trên cho số nguyên dương K ($1 \leq K \leq 11$)
4. Mỗi số dư tương ứng với một chữ cái in hoa. Chữ cái in hoa tương ứng tìm được là chữ cái nổi phía sau của mã nhân viên.

Cho trước phần đầu của mã nhân viên, hãy tìm chữ cái in hoa nổi phía sau.

Dữ liệu vào: Từ file văn bản MASO.INP có cấu trúc như sau:

Dòng đầu tiên ghi số nguyên dương K .

Dòng thứ 2 ghi một xâu gồm 7 chữ số là phần đầu của mã nhân viên

Dòng thứ 3 ghi 7 số nguyên dương, số thứ i là trọng lượng của chữ số thứ i trong phần đầu của mã. Các số cách nhau một kí tự trắng.

Dòng cuối cùng ghi một xâu có độ dài K gồm các chữ cái tiếng Anh in hoa, chữ cái thứ i tương ứng với số $i-1$ là số dư của phép chia của tổng tìm được cho K .

Kết quả: Ghi ra file văn bản MASO.OUT, chứa chữ cái tìm được.

Ví dụ:

MASO.IN	MASO.OUT
11 6830907 2 7 6 5 4 3 2 JABCDEFGHIZ	D

Ý tưởng

1. Mã số nhân viên là xâu chữ số, do đó để tính tổng thì ta phải chuyển từng chữ số thành số rồi lấy giá trị số vừa chuyển nhân với trọng lượng của nó. Thủ tục để chuyển kí tự thành số là **val(s,n,code)**, trong đó S là xâu kí tự, n là biến kiểu số, $code$ là mã số để kiểm tra quá trình chuyển đổi. $code=0$ là chuyển thành công, ngược lại là không thành công.
2. Ở dòng cuối của input ta thấy mỗi chữ cái thứ i ứng với số $i-1$ là số dư của phép chia tổng tìm được cho k .
3. Do đó: khi muốn lấy kí tự tương ứng với biến du thì ta tham chiếu đến phần tử ở vị trí $du+1$

Bài toán tương tự: Thẻ sử dụng Internet cho HS gồm 2 phần. Phần đầu là 7 số & phần sau là 1 chữ cái. Chữ cái đó được tạo từ phần số. Gồm các bước

B1: Nhân mỗi chữ cái của số đã cho với số của nó. Chữ số bên trái cùng có trong số là 2. Chữ số thứ 2 có trong số là 7, chữ số thứ 3 có trong số là 6,... cho đến chữ số bên phải cùng có trong số là 2.

B2: cộng tất cả các tích tìm được ở bước 1

B3: Tìm số dư của phép chia tổng tính được ở bước 2 cho 11

B4: Đưa vào số dư để tìm chữ cái tương ứng theo quy định: 0-J ; 1 -A ; 2- B ; 3-C ; 4- D

Ví dụ: Cho số: 6830907

tổng: $2*6+7*8+6*3+5*0+4*9+3*0+2*7=136$

số dư là $136 \bmod 11 = 4 \Rightarrow$ chữ tương ứng là D

M1C2C. BÀI 4. Mua hàng

Có n người xếp thành hàng theo thứ tự để mua hàng. Thời gian người bán hàng phục vụ cho người thứ i là t_i đơn vị thời gian. Hãy tìm thời gian mà người thứ k phải chờ để mua hàng.

Dữ liệu vào: cho trong tệp văn bản MH.INP

- Dòng 1: chứa 2 số n, k nguyên dương ($1 \leq k \leq n \leq 100$);

- Dòng 2: chứa n số t_1, t_2, \dots, t_n (Các giá trị t_i đều nguyên dương và nhỏ hơn 1000).

Kết quả: ghi ra tệp văn bản MH.OUT chứa duy nhất số C thỏa mãn yêu cầu của dữ liệu vào.

Ý tưởng: Thời gian phục vụ người thứ k bằng tổng thời gian phục vụ những người từ 1 đến $k-1$.

M1C2C. BÀI 5. Tập số đặc biệt

Xét một tập hợp A gồm các số tự nhiên được xác định như sau:

- 1 thuộc A ;

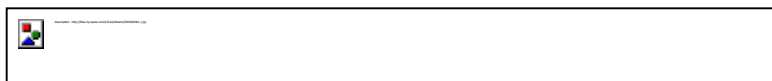
- Nếu k thuộc A thì $2k+1$ và $3k+1$ cũng thuộc A .

Cho số tự nhiên N ($1 \leq N \leq 1000$). Em hãy chỉ ra N phần tử đầu tiên của tập A theo thứ tự tăng dần.

Dữ liệu vào: Đọc từ file văn bản TAPSO.INP gồm số N duy nhất.

Kết quả: Ghi vào file văn bản TAPSO.OUT là dãy N phần tử đầu tiên của tập A theo thứ tự tăng dần.

Ví dụ:



Ý tưởng:

- Dựa vào tính chất dãy số của đề bài để viết chương trình. (Sử dụng vòng lặp While – do)

MIC2C. BÀI 6. Mua vé

N người có số hiệu $1 \dots N$ xếp hàng mua vé ($n \leq 100$). Người có số hiệu bé hơn thì xếp trước và được mua vé trước. Mỗi người chỉ được mua một vé cho mình và có thể mua hộ một vé cho người đứng ngay sau đó. Người có số hiệu thứ i khi mua một vé mất thời gian là T_i phút. Người thứ k khi mua 1 vé cho mình và mua hộ một vé cho người khác đứng ngay sau đó thì mất thời gian R_k phút. Hãy sắp xếp xem người nào cần ở lại trong hàng để mua vé, người nào cần phải ra khỏi hàng và nhờ người đứng ngay trước mình mua hộ vé để tổng thời gian bán đủ vé cho N người là ít nhất

ví dụ:

T_i : 3 2 4 3

R_k : 4 3 7

Kết quả là: 9

Số hiệu người đứng mua vé là 4,2,1

MIC2C. BÀI 7. Tổng các chữ số

Cho dãy a_1, a_2, \dots, a_N số tự nhiên. Hãy tính tổng các số của dãy, cho biết số hạng lớn nhất trong dãy và vị trí của các số hạng này.

Dữ liệu: Vào từ tệp văn bản **SO.INP**:

Chỉ một dòng chứa các số a_1, a_2, \dots, a_N ghi cách nhau một dấu cách.

($0 < N \leq 10000$; $0 \leq a_i \leq 6 \cdot 10^4$ với $i = 1, 2, \dots, N$)

Kết quả: Ghi ra tệp văn bản **SO.OUT** có cấu trúc như sau:

Dòng thứ nhất ghi tổng các số của dãy.

Dòng thứ 2 ghi số lớn nhất.

Dòng thứ 3 ghi các vị trí xuất hiện của số lớn nhất.

Các số trên cùng một dòng ghi cách nhau một dấu cách.

Ví dụ:

SO.INP	SO.OUT
4 3 2 1 0 8 7 4 2 5 4 6 7 8 2 3 1	67
	8
	6 14

Ý tưởng: bài toán này là bài toán cơ bản, gv hướng dẫn hs vận dụng kiến thức kiểu mảng và thuật toán tìm max để giải.

MIC2C. BÀI 8. Chia đồ vật

Trong một gia đình nông dân có bốn người con, khi các con đến tuổi trưởng thành, người cha gọi các con tới và nói: “Cả đời làm lụng vất vả của cha mẹ có tích trữ được một số của cải là các đồ vật có giá trị. Nay muốn chia cho các con...”. Người cha có N đồ vật, đồ vật thứ i có giá trị là $a[i]$. Người cha muốn chia N đồ vật trên cho 4 người con sao cho độ chênh lệch giữa tổng giá trị các đồ vật mà mỗi người con nhận được là nhỏ nhất.

Yêu cầu: Hãy giúp người cha tìm một cách chia N đồ vật theo yêu cầu như trên.

Ví dụ

CHIA.INP	CHIA.OUT
7	1
3 12 8 9 5 10 4	1 6
	2
	3 5
	4 7

Dữ liệu vào: Từ tệp văn bản **CHIA.INP** gồm:

- Dòng 1: Số nguyên N là số đồ vật cần chia ($4 \leq N \leq 100$).
- Dòng 2: Gồm N số nguyên $a[i]$ là giá trị của đồ vật thứ i ($1 \leq a[i] \leq 100$), mỗi số ghi cách nhau một dấu cách.

Dữ liệu ra: Ghi ra tệp văn bản CHIA.OUT gồm:

- Dòng 1: Chứa độ chênh lệch nhỏ nhất tìm được.
- Bốn dòng tiếp theo: Mỗi dòng ghi thứ tự các đồ vật mà người con thứ 1, 2, 3, 4 nhận được.

M1C2C. BÀI 9. Tìm dãy con có tổng lớn nhất (không chọn 3 phần tử liên tiếp)

Cho dãy n số nguyên dương a_1, a_2, \dots, a_n . Một dãy con của dãy nói trên là dãy được lập từ dãy đã cho bằng cách bỏ đi một số số hạng của dãy và giữ nguyên trật tự các số còn lại. Hãy tìm một dãy con thoả mãn tính chất:

- Không có ba số liên tiếp nào của dãy ban đầu có mặt trong dãy con
- Trong ba số liên tiếp của dãy ban đầu có ít nhất một số có mặt trong dãy con
- Tổng các số hạng của dãy con được chọn là lớn nhất có thể được.

Dữ liệu: Vào từ file CHONSO.INP:

- Dòng đầu tiên chứa số nguyên dương N ($N \leq 1000$)
- N dòng tiếp theo, dòng thứ i chứa số nguyên dương a_i ($a_i \leq 30000$)

Kết quả: Ghi ra file văn bản CHONSO.OUT:

- Dòng đầu tiên chứa hai số nguyên dương M và T trong đó M là số lượng các số hạng của dãy con được chọn, T là tổng các số của dãy con được chọn.
- M dòng tiếp theo lần lượt mô tả các số hạng của dãy con được chọn, dòng thứ k ghi số j_k là chỉ số của số hạng được chọn thứ k .

Ví dụ:

CHONSO.INP	CHONSO.OUT
6	21 4
2	2
6	3
5	5
1	6
7	
3	

M1C2C. BÀI 10. Chọn phần thưởng

(dạng bài toán tìm dãy con có tổng lớn nhất theo điều kiện không được chọn 3 phần tử liên tiếp)

Bạn là người đạt giải đặc biệt trong một kỳ thi tin học. Ban tổ chức cho phép bạn chọn phần thưởng cho mình. Các phần thưởng của ban tổ chức được xếp thành một dãy, đánh số từ 1 đến N , phần thưởng thứ i có giá trị là a_i . Bạn sẽ được phép chọn phần thưởng cho mình theo nguyên tắc không được chọn 3 phần thưởng liên tiếp nhau trong dãy phần thưởng của ban tổ chức.

Yêu cầu: Viết chương trình nêu ra cách chọn các phần thưởng của ban tổ chức sao cho tổng giá trị của các phần thưởng mà bạn nhận được là lớn nhất.

Dữ liệu: Vào từ file văn bản PTHUONG.INP:

- Dòng đầu ghi số nguyên N là số phần thưởng của ban tổ chức ($N \leq 25000$)
- Trong N dòng tiếp theo, dòng thứ i ghi số nguyên a_i là giá trị của phần thưởng thứ i ($a_i \leq 10^6$)

Kết quả: Ghi ra file văn bản PTHUONG.OUT:

- Dòng đầu tiên ghi tổng giá trị của các phần thưởng được lựa chọn.
- Trong các dòng tiếp theo, lần lượt ghi số hiệu của các phần thưởng mà bạn sẽ lấy, mỗi dòng ghi đúng 10 số hiệu, trừ dòng cuối cùng có thể ít hơn 10 số hiệu. Các số hiệu được ghi lần lượt theo trình tự tăng dần, các số hiệu ghi trên một dòng cách nhau ít nhất một kí tự trắng.

Ví dụ:

PTHUONG.INP	PTHUONG.OUT
5	23
6	1 2 4 5
9	
1	
3	
5	

M1C2C. BÀI 11. Chia dãy số thành k nhóm có tổng bằng nhau

(trích bài tập 4.5 tài liệu giáo khoa chuyên tin quyển 1)

Cho dãy số A gồm n ($n \leq 10$) số nguyên A_1, A_2, \dots, A_n và 1 số nguyên dương k ($1 < k < n$). Hãy đưa ra 1 cách chia dãy số thành k nhóm sao cho các nhóm có tổng bằng nhau.

Dữ liệu vào: Nhập từ file CHIA.INP gồm dòng đầu tiên chứa số nguyên n và k ; dòng sau chứa dãy số nguyên A

Dữ liệu ra: Ghi ra file CHIA.OUT gồm k dòng, mỗi dòng là 1 nhóm.

Ý tưởng

Bài này thực hiện tương tự bài toán chia vật ở trên nhưng số nhóm chia là k, ta thử tất cả các khả năng có thể để chọn được phương án tối ưu.

MIC2C. BÀI 12. Dãy con liên tiếp có tổng lớn nhất

(câu 1, HSG Long An, lớp 12, vòng 2, NH: 2011-2012)

Cho dãy số nguyên A gồm n phần tử a_1, a_2, \dots, a_n , tìm số cặp chỉ số (i, j) thỏa mãn: $S = a_i + a_{i+1} + \dots + a_j$ đạt giá trị lớn nhất (với $1 \leq i < j \leq n$).

Dữ liệu: Vào từ file văn bản DAYSO.INP

Dòng đầu là số nguyên dương n ($2 \leq n \leq 10^5$)

Dòng tiếp theo chứa n số nguyên a_1, a_2, \dots, a_n ($|a_i| < 10^9$), các số cách nhau một dấu cách.

Kết quả: Đưa ra file văn bản DAYSO.OUT

Dòng 1 chứa tổng tìm được.

Dòng 2 chứa 2 số nguyên i, j (hai số cách nhau một dấu cách).

Ví dụ

DAYSO.INP	DAYSO.OUT	Giải thích
8 -2 5 9 -4 -6 8 7 -5	19 2 7	Dãy A gồm 8 số nguyên -2 5 9 -4 -6 8 7 -5, tìm được 1 cặp chỉ số (i, j) với i=2, j=7, tổng S=19 đạt giá trị lớn nhất.

MIC2C. BÀI 13. Dãy con cấp số cộng

(Câu 3 - đề thi HSG tỉnh Bạc Liêu năm học 2010-2011)

Cho dãy N số tự nhiên ($2 \leq N \leq 1000$). Hãy tìm trong dãy con dài nhất tạo thành CSC có công sai d.

Dữ liệu vào: DAYCON.INP

+ Dòng đầu tiên là số N, d.

+ Các dòng tiếp theo là N số trong dãy, các số cách nhau một dấu cách.

Kết quả ra: DAYCON.OUT

+ Dòng đầu ghi số lượng phần tử trong dãy con tìm được.

+ Dòng thứ 2 ghi các phần tử trong dãy con đã tìm được (sắp xếp tăng)

Ví dụ

DAYCON.INP
15 4
9 7 4 1 2 6 4 9 7 11 8 10 15 11 10

DAYCON.OUT
3
2 6 10

Ý tưởng: áp dụng thuật toán tìm dãy con có nhiều phần tử và có tổng lớn nhất.

MIC2C. BÀI 14. Dãy con liên tiếp có tổng chia hết cho k

Cho một mảng số nguyên gồm n phần tử. Tìm tất cả dãy con gồm m phần tử ($m \leq n$) sao cho dãy con này có tổng chia hết cho k. (Dãy con là dãy các phần tử liên tiếp nhau trong mảng).

Input: file văn bản DAYSO.INP

Dòng 1: Chứa 3 số n, m và k cách nhau ít nhất một khoảng cách.

Dòng 2: Chứa n số A_1, A_2, \dots, A_n cách nhau ít nhất một khoảng cách.

Output: file văn bản DAYSO.OUT

Dòng i: ($1 \leq i \leq d$): Ghi các phần tử trong dãy con thứ i các phần tử cách nhau ít nhất một khoảng cách.

Dòng d+1: Ghi số d (d là số lượng dãy con chia hết cho k).

Ví dụ:

DAYSO.INP	DAYSO.OUT
10 4 7	Day con thu 1: 7 9 3 2 co tong la: 21
1 7 9 3 2 0 2 10 1 1	Day con thu 2: 9 3 2 0 co tong la: 14
	Day con thu 3: 3 2 0 2 co tong la: 7
	Day con thu 4: 2 0 2 10 co tong la: 14
	Day con thu 5: 2 10 1 1 co tong la: 14
	So luong day con chia het cho 7 la: 5

MIC2C. BÀI 15. Dãy con liên tiếp có tổng bằng M

(câu 2, HSG Quảng Bình, 2012-2013)

Cho dãy số nguyên dương gồm N phần tử A_1, A_2, \dots, A_n .

Yêu cầu: Hãy liệt kê tất cả các dãy con gồm các phần tử đứng liên tiếp của dãy trên sao cho tổng giá trị các phần tử của dãy con đó bằng M.

Dữ liệu vào: Cho trong file văn bản DAYCON.INP, có cấu trúc như sau:

- Dòng 1: Ghi 2 số nguyên dương N và M. Hai số được ghi cách nhau ít nhất một dấu cách.

- Dòng 2: Ghi N số nguyên dương A_i . Các số được ghi cách nhau ít nhất một dấu cách ($N \leq 100$)

Dữ liệu ra: Ghi ra file văn bản DAYCON.OUT, gồm nhiều dòng. Mỗi dòng ghi một dãy con tìm được. Các dãy con được ghi theo thứ tự xuất hiện của nó trên dãy ban đầu. Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Nếu không tìm được dãy con thỏa mãn điều kiện thì ghi ra file một số: 0.

Ví dụ:

DAYCON.INP	DAYCON.OUT
11 6 1 4 3 1 2 1 6 5 1 4 1	3 1 2 6 5 1 1 4 1
5 6 1 2 5 2 1	0

MIC2C. BÀI 16. Dãy con nguyên tố dài nhất

(Câu 3, HSG Đà Nẵng 2001-2002, vòng 1)

Cho dãy N ($N \leq 100$) số nguyên dương a_1, a_2, \dots, a_N đôi một khác nhau, giá trị mỗi số không vượt quá 10000. Một dãy con được tạo ra bằng cách chọn một số phần tử trong dãy đã cho nhưng phải giữ đúng thứ tự (từ trái sang phải) như ban đầu. Một dãy con gọi là “dãy con nguyên tố” nếu hai phần tử bất kỳ nằm cạnh nhau trong dãy con đó đều nguyên tố cùng nhau.

Yêu cầu: Viết chương trình tìm một dãy con nguyên tố dài nhất.

Dữ liệu vào: trong file văn bản DATA.TXT:

- Dòng đầu tiên ghi số N.

- Các dòng tiếp theo chứa các số a_1, a_2, \dots, a_N được ghi cách nhau bởi ít nhất một khoảng trắng.

Dữ liệu ra: In ra màn hình số phần tử của dãy con nguyên tố dài nhất và giá trị các phần tử đó nếu tìm được, hoặc đưa ra lời phủ định nếu không tìm được.

Ví dụ: - Dữ liệu cho ở file DATA.TXT như sau:

10
5 7 9 15 12 4 6 13 8 21

- Kết quả in ra màn hình:

Số phần tử của dãy con nguyên tố dài nhất: 7

Các phần tử của dãy con là: 5 7 9 4 13 8 21

MIC2C. BÀI 17. Dãy con có tổng bằng S (dãy không liên tiếp)

Cho N số nguyên dương tạo thành dãy $A = \{A_1, \dots, A_N\}$. Tìm một dãy con của A có tổng các phần tử bằng S.

Dữ liệu vào từ file DAYCON.INP

Dòng đầu tiên ghi hai số nguyên dương N ($0 \leq N \leq 200$) và S ($0 \leq S \leq 40000$)

Các dòng tiếp theo lần lượt ghi N số hạng của dãy A ($0 \leq A_i \leq 200$)

Kết quả ra ghi ra file DAYCON.OUT

Nếu bài toán vô nghiệm ghi số 0.

Nếu bài toán có nghiệm thì trên dòng thứ nhất ghi số 1. Các dòng tiếp theo, mỗi dòng ghi hai số là chỉ số trong dãy A và giá trị của một phần tử được chọn.

Bài toán tương tự: Tìm một dãy con (không cần liên tiếp) có tổng lớn nhất.

Dữ liệu vào: file daycon.inp có

+ Dòng đầu chứa một số nguyên là số phần tử của dãy;

+ Dòng 2: Chứa n số nguyên ứng với số phần tử của dãy

Kết quả: ghi ra file daycon.out

+ Dòng đầu ghi tổng lớn nhất tìm được

+ Dòng sau: ghi các giá trị các phần tử của dãy (theo đúng thứ tự đã nhập)

Ví dụ

DAYCON.INP	DAYCON.OUT
10 1 7 -9 3 2 0 -2 10 1 1	25 1 7 3 2 10 1 1

MIC2C. BÀI 18. Tính tổng

(câu 1, HSG An Giang 2009-2010) (câu 2, HSG Đắk Lắk, 2011-2012)

Trên một màn hình lớn, người ta lần lượt cho hiện ra các số của một dãy gồm N số nguyên không âm a_1, a_2, \dots, a_N và cứ lặp đi lặp lại như thế. Mỗi người theo dõi màn hình được đề nghị tính tổng của K số nguyên liên tiếp xuất hiện trên màn hình bắt đầu từ số nguyên thứ B .

Viết chương trình giúp cho những người theo dõi màn hình tính được tổng như đề nghị.

Dữ liệu vào: chứa trong tệp văn bản SUM.INP gồm hai dòng

+ Dòng đầu tiên ghi ba số nguyên N, K và $B, 1 \leq N \leq 100, 1 \leq K \leq 100, 1 \leq B \leq 10^9$.

+ Dòng thứ hai chứa dãy số nguyên không âm a_1, a_2, \dots, a_N .

Dữ liệu ra: ghi vào tệp văn bản SUM.OUT gồm một dòng chứa tổng cần tính.

Ví dụ:

SUM.INP	SUM.OUT
5 7 4	24
1 2 3 4 5	

MIC2C. BÀI 19. Sắp xếp mảng 1 chiều

(Câu 2, HSG Long An, lớp 12 Bảng B, vòng 1 2011-2012)

Cho mảng một chiều A có n phần tử, phần tử thứ i của mảng A có giá trị là i . Người ta sắp xếp mảng A thành mảng B theo thứ tự với hai phần tử a, b trong mảng B, a ở trước b khi số dư của a chia cho 5 nhỏ hơn số dư của b chia 5. Trong trường hợp hai số dư bằng nhau thì a trước b khi a nhỏ hơn b .

Yêu cầu: Viết chương trình nhập vào các số nguyên dương n, m . Phần tử thứ m của mảng A thành phần tử thứ k của mảng B . Xuất ra màn hình giá trị của k .

Ví dụ: Nhập vào $n = 10, m = 9$ ta được $k = 10$ vì

Mảng A	1	2	3	4	5	6	7	8	9	10
--------	---	---	---	---	---	---	---	---	---	----

Sắp xếp thành mảng B

Mảng B	5	10	1	6	2	7	3	8	4	9
--------	---	----	---	---	---	---	---	---	---	---

Ý tưởng.

Tạo mảng C lưu trữ số dư của từng phần tử của A chia cho 5, mảng VT lưu vị trí ban đầu của từng phần tử. Sau đó sắp xếp lại mảng A theo giá trị không giảm của các phần tử trong C . Kết quả sắp xếp là mảng B cần tìm. Khi đó phần tử thứ K là $VT[m]$

MIC2C. BÀI 20. Sắp xếp

Cho một dãy X gồm N số nguyên trong phạm vi từ -10000 đến 10000 ($1 \leq N \leq 100000$). Hãy sắp xếp dãy số này theo thứ tự giảm dần.

Dữ liệu vào cho trong file văn bản SORT.INP trong đó dòng đầu chứa số N . Dòng thứ i trong N dòng tiếp theo chứa số thứ i trong dãy X .

Kết quả ghi ra file văn bản với tên SORT.OUT trong đó lần lượt ghi ra các phần tử của dãy X đã được sắp xếp mỗi số trên một dòng.

Ví dụ:

SORT.INP
4
3
4
2
5

SORT.OUT
5
4
3
2

Ý tưởng

Phương pháp sắp xếp mà chúng ta dùng là sắp xếp đếm (counting sort). Phương pháp này tận dụng việc giới hạn của các số cần sắp xếp có thể lưu đủ trong bộ nhớ, trong bài toán này phạm vi các số là -10000..10000. Ta sẽ dùng mảng $dem[-10000..10000]$ trong đó $dem[x]$ lưu số lần xuất hiện của số x trong dãy số.

Bài toán này được ra với yêu cầu làm trên trình biên dịch Borland Pascal. Với trình biên dịch cũ này, quản lý bộ nhớ là một điều quan trọng do dung lượng bộ nhớ bị hạn chế. Chúng tôi sẽ trình bày lời giải bài toán này trên cơ sở bộ nhớ hạn chế đó.

Do mỗi số có thể xuất hiện đến $N \leq 100\,000$ lần nên mảng dem phải mang kiểu dữ liệu longint (số nguyên 32 bit) trong Pascal. Trong Borland Pascal, khi khai báo mảng 20000 phần tử longint sẽ bị báo lỗi là không đủ bộ nhớ. Có một cách để giải quyết điều này:

Khai báo mảng đếm với kiểu dữ liệu word (có giới hạn từ 0..65535) như sau:
var dem: array[-10000..10000] of word;
Ta nhận xét chỉ có nhiều nhất một phần tử của mảng đếm có thể có giá trị lớn hơn hoặc bằng 60000, vì tổng số lượng số nhiều nhất là 100 000. Do đó ta quản lý thêm một biến lưu phần tử đặc biệt có giá trị đếm vượt 60000 này (nếu có). Ta đặt tên biến này là v. Đoạn lệnh dưới đây đọc vào các số và quản lý dữ liệu:

```

for i:=1 to n do
begin
  readln(x);          {đọc vào một số x}
  inc(dem[x]);         {tăng biến đếm số lần xuất hiện của số x}
  if (dem[x]=60000) then {nếu đã có 60000 số x xuất hiện}
  begin
    v:=x;              {lưu lại số x duy nhất này}
    dem[x]:=0;         {gán lại biến đếm bằng 0 để tránh tràn số}
  end;
end;

```

Đoạn lệnh dưới đây in ra các số đã sắp xếp theo thứ tự giảm dần:

```

for i:=-10000 downto 10000 do {duyệt qua phạm vi của các số: [-10000,10000]}
begin
  for j:=1 to dem[i] do {in ra số i với số lần là dem[i]}
    writeln(i);
  if (v=i) then for j:=1 to 60000 do {nếu i là số đặc biệt thì ta cần in thêm 60000 lần xuất hiện nữa}
    writeln(i);
end;

```

M1C2C. BÀI 21. Số tự nhiên nhỏ nhất

(câu 2, HSG Hà Tĩnh, lớp 10, 2012-2013)

Nam một người bạn của Nguyên đang tìm cách giải một bài toán liên quan tới số tự nhiên và cần sự giúp đỡ của Nguyên, nhưng thử thách lần này là một dãy gồm N số tự nhiên bất kì nằm trong đoạn từ 0 tới 10^9 , tìm số tự nhiên nhỏ nhất không có trong dãy số đó. Vì số lượng các số tự nhiên trong dãy số đã cho có thể lên tới 10^6 phần tử nên việc tìm thủ công là không thể mà cần một thuật toán để cài đặt vào máy tính và nhờ máy tính tìm giúp.

Yêu cầu: Cho một dãy A gồm N ($1 \leq N \leq 10^6$) số tự nhiên. Hãy tìm số tự nhiên nhỏ nhất không xuất hiện trong dãy A.

Ví dụ:

Dữ liệu vào	Kết quả
N= 5 Dãy số: 5 4 2 3 1	0
N= 9 Dãy số: 2 4 0 3 1 2 6 2 8	5

M1C2C. BÀI 22. Dãy phân số

Cho hai dãy số nguyên dương a_1, a_2, \dots, a_M và b_1, b_2, \dots, b_N . Từ hai dãy trên tạo ra $M \times N$ phân số với $i = 1, 2, \dots, M, j = 1, 2, \dots, N$. Sắp xếp các phân số vừa tạo theo thứ tự tăng dần sau khi đã tối giản và loại bớt các phân số bằng nhau (các phân số bằng nhau chỉ giữ lại một lần) thu được dãy phân số P.

Ví dụ, dãy thứ nhất gồm 2 phần tử 10, 30; còn dãy thứ 2 gồm 3 phần tử 20, 30, 60 ta tạo được các phân số là:

$\frac{10}{20}, \frac{10}{30}, \frac{10}{60}, \frac{30}{20}, \frac{30}{30}, \frac{30}{60}$ thì dãy phân số P là $\frac{1}{6}, \frac{1}{3}, \frac{1}{2}, 1, \frac{3}{2}$

Yêu cầu: Cho số nguyên dương K, hãy tìm phân số thứ K trong dãy P.

Input: Vào từ file văn bản PS.INP có dạng:

- Dòng đầu tiên ghi 3 số nguyên dương M, N, K ($1 \leq M, N \leq 30$).
- Dòng thứ 2 ghi m số nguyên dương a_1, a_2, \dots, a_M .
- Dòng thứ 3 ghi n số nguyên dương b_1, b_2, \dots, b_N .

($a_i, b_j \leq 10^9$ với $i=1..M, j=1..N$).

Output: Ghi ra file văn bản PS.OUT gồm 2 số nguyên dương là tử số và mẫu số của phân số tìm được (hai số ghi cách nhau một dấu cách).

Chú ý: Có 60% test N=1 và $b_1=1$. Dữ liệu bảo đảm k không vượt quá số lượng phần tử của dãy phân số P.

Ý tưởng

- Liệt kê thành dãy $M \times N$ các phân số dưới dạng tối giản
- Sắp xếp theo thứ tự tăng dần

Ps.inp
2 3 4
10 30
20 30 60

Ps.out
1 1

- Loại các phần tử giống nhau (chỉ giữ lại 1)
- Đưa ra phân số thứ K trong dãy số mới.

MIC2C. BÀI 23. Độ lệch

(câu 2, Hải Dương, lớp 12 thpt năm học 2012 – 2013)

Xét một số N có 4 chữ số và không phải tất cả các chữ số đều giống nhau. Phép tính độ lệch được thực hiện như sau:

Tạo số thứ nhất N1 bằng cách xếp các chữ số theo trình tự giảm dần

Tạo số thứ hai N2 bằng cách xếp các chữ số theo trình tự tăng dần (nếu có chữ số 0 ở đầu thì N2 sẽ không phải là số có 4 chữ số)

Tính hiệu N1-N2 và gán lại cho N.

Các bước trên được thực hiện cho đến khi nhận được số N là 6174 hoặc 0.

Ví dụ: Nếu N=1023

Ở bước 1: N1=3210, N2=123, N=N1-N2=3087

Ở bước 2: N1=8730, N2=378, N=N1-N2=8352

Ở bước 3: N1=8532, N2=2358, N=N1-N2=6174

Vậy ta cần thực hiện 3 lần biến đổi

Yêu cầu: Hãy xác định số lần biến đổi thực hiện theo yêu cầu trên.

Dữ liệu vào: Nhập từ bàn phím số nguyên dương N (N đảm bảo có 4 chữ số, không phải tất cả các chữ số đều giống nhau và N khác 6174. Không cần kiểm tra dữ liệu nhập)

Kết quả ra: Ghi ra màn hình số lần biến đổi tương ứng với số N

Ví dụ:

Dữ liệu nhập: 5364

Kết quả in ra: 3

MIC2C. BÀI 24. Số siêu nguyên tố

(câu 3, Hải Dương, lớp 12 thpt năm học 2012 – 2013)

Số siêu nguyên tố là số nguyên tố mà khi xóa bỏ dần các chữ số bên phải của nó thì phần còn lại vẫn là số nguyên tố. Ví dụ: 2333 là số siêu nguyên tố vì 2333, 233, 23, 2 đều là các số nguyên tố.

Yêu cầu: Cho số nguyên dương M ($M \leq 30000$). Hãy tìm số siêu nguyên tố gần với M nhất, tức là trị tuyệt đối của hiệu giữa số tìm được với M là nhỏ nhất

Dữ liệu: Nhập vào từ bàn phím số nguyên dương M (không cần kiểm tra dữ liệu nhập)

Kết quả: Ghi ra màn hình các số nguyên tố gần M nhất, mỗi số một dòng theo thứ tự số nhỏ hơn ghi trước.

Ví dụ:

Dữ liệu vào	Kết quả ra
Dữ liệu nhập: 30	Kết quả in ra trên 2 dòng: 29 31

MIC2C. BÀI 25. Bình chọn qua điện thoại

(câu 4, Hải Dương, lớp 12 thpt năm học 2012 – 2013)

Trong vòng chung kết cuộc thi “Vietnam Next Top Model” trên VTV3 các thí sinh được đánh số báo danh là một số nguyên dương có giá trị không vượt quá 1000. Khán giả xem truyền hình có thể bình chọn cho thí sinh mình yêu thích bằng cách nhấn tin qua điện thoại di động.

Ban tổ chức nhận được tin nhắn hợp lệ của N khán giả (các khán giả được đánh số từ 1 đến N), khán giả thứ i bình chọn cho thí sinh mang số báo danh a_i .

Hãy liệt kê số báo danh của những thí sinh được nhiều khán giả bình chọn nhất theo thứ tự tăng dần.

Dữ liệu: Vào từ file văn bản VNMODEL.INP

Dòng đầu tiên ghi số nguyên dương N là số lượng khán giả có tin nhắn bình chọn hợp lệ ($N \leq 10^6$)

N dòng tiếp theo, dòng thứ i ghi số nguyên dương a_i là số báo danh của thí sinh mà khán giả thứ i bình chọn.

Kết quả: Ghi ra file văn bản VNMODEL.OUT

Danh sách các thí sinh được nhiều khán giả bình chọn nhất theo thứ tự số báo danh tăng dần

Ví dụ:

VNMODEL.INP	VNMODEL.OUT
5	2
3	3
1	1
3	
2	
2	

Ghi chú: Có ít nhất 50% số điểm của bài tương ứng với các test có $N \leq 1000$.

M1C2C. BÀI 26. Quan hệ huyết thống

(câu 5, Hải Dương, lớp 12 thpt năm học 2012 – 2013)

Trung tâm nghiên cứu gen thu thập N mẫu gen của N cá thể trong cùng một loài. N gen này được mã hoá thành dãy N số nguyên dương a_1, a_2, \dots, a_N . Bộ phận phân tích sau khi xem xét đã đưa ra được kết luận sau:

Hai cá thể là có quan hệ huyết thống gần khi và chỉ khi mã gen của chúng biểu diễn trong cơ số 2 giống nhau hoặc khác nhau đúng 1 bit

Ví dụ: Hai cá thể có mã gen 7 (biểu diễn trong cơ số 2 là 111) và 5 (biểu diễn trong cơ số 2 là 101) là có quan hệ huyết thống gần.

Hãy đếm xem trong số mẫu gen của N cá thể thu thập được có bao nhiêu cặp cá thể có quan hệ huyết thống gần.

Dữ liệu: Vào từ file văn bản GEN.INP

Dòng đầu tiên ghi số nguyên dương N ($N \leq 10^5$)

N dòng tiếp theo, dòng thứ i ghi mã gen của cá thể thứ i là một số nguyên dương trong phạm vi từ 1 đến 1000)

Kết quả: Ghi ra file văn bản GEN.OUT một số nguyên duy nhất là số cặp có quan hệ huyết thống gần đã tìm được.

Ví dụ:

GEN.INP	GEN.OUT
5	4
1	
2	
3	
4	
5	
3	3
10	
10	
10	

Giải thích ví dụ thứ nhất:

Các số 1, 2, 3, 4, 5 biểu diễn trong cơ số 2 lần lượt là 001, 010, 011, 100, 101. Có 4 cặp có quan hệ huyết thống gần là: 1 và 3, 1 và 5, 2 và 3, 4 và 5

M1C2C. BÀI 27. Tìm số

(Câu 1, Quảng Bình 2012-2013)

Cho số nguyên dương X , khi đảo ngược trật tự các chữ số của X ta sẽ thu được một số nguyên dương Y , Y được gọi là số đảo ngược của X .

Ví dụ: $X = 613$ thì $Y = 316$ là số đảo ngược của X .

Số nguyên dương Y được gọi là số nguyên tố nếu nó chỉ có hai ước số là 1 và chính nó, số 1 không phải là số nguyên tố.

Cho hai số nguyên dương P và Q ($1 \leq P \leq Q \leq 2 \times 10^9$; $Q - P \leq 10^5$).

Yêu cầu: Hãy tìm tất cả các số nguyên dương X nằm thỏa mãn $P \leq X \leq Q$ và số đảo ngược của số X là số nguyên tố.

Dữ liệu vào: Cho trong file văn bản TIMSO.INP có cấu trúc như sau:

- Dòng 1: Ghi hai số nguyên dương P, Q , hai số được ghi cách nhau ít nhất một dấu cách.

Dữ liệu ra: Ghi ra file văn bản TIMSO.OUT trên nhiều dòng, mỗi dòng ghi một số nguyên X tìm được.

Ví dụ:

TIMSO.INP	TIMSO.OUT
10 19	11
	13
	14
	16
	17

M1C2C. BÀI 28. Số âm lớn nhất

(câu 1, đề thi Nghệ An 2012-2013, bảng B)

Cho một dãy gồm N số nguyên a_1, a_2, \dots, a_N , mỗi số có giá trị tuyệt đối không vượt quá 10^5 .

Yêu cầu: Hãy tìm số âm lớn nhất X trong dãy.

Dữ liệu vào từ file văn bản SOAM.INP:

Dòng đầu tiên chứa số nguyên dương N ($1 \leq N \leq 10^5$).

N dòng tiếp theo, dòng thứ i chứa số a_i .

Kết quả ghi ra file văn bản SOAM.OUT: Chỉ một dòng duy nhất là số X tìm được. Trong trường hợp không có lời giải thì ghi ra số 0.

Ví dụ:

SOAM.INP	SOAM.OUT
5 -4 3 2 -5 7	-4

MIC2C. BÀI 29. Trò chơi may mắn

(câu 2, đề thi Nghệ An 2012-2013, bảng B)

Nhân dịp lễ giáng sinh, công viên trung tâm tổ chức trò chơi "con số may mắn". Mỗi em nhỏ đến tham dự sẽ được phát một số nguyên dương. Công viên có một thiết bị quay số, mỗi lần quay sẽ tạo ngẫu nhiên một số nguyên dương có giá trị tuyệt đối không vượt quá 32000. Người dẫn chương trình sẽ thực hiện N lần quay số. Số nào xuất hiện nhiều nhất trong N lần quay được gọi là con số may mắn, và em nhỏ nào có con số may mắn thì sẽ được phần thưởng.

Yêu cầu: Cho N con số xuất hiện trong N lần quay. Bạn hãy giúp người dẫn chương trình xác định số lần xuất hiện của con số may mắn.

Dữ liệu vào từ file văn bản MAYMAN.inp:

Dòng đầu là số N ($1 \leq n \leq 100000$)

Dòng tiếp theo có N số là các số xuất hiện trong N lần quay.

Kết quả ghi ra file văn bản MAYMAN.out: Là số lần xuất hiện của con số may mắn.

Ví dụ:

MAYMAN.INP	MAYMAN.OUT	MAYMAN.INP	MAYMAN.OUT
5 4 3 4 4 15	3	7 12 5 10 5 8 10 9	2

MIC2C. BÀI 30. Sắp xếp

Để quản lý nhân sự ở một tỉnh nọ có khoảng 1 triệu người, hãy sắp xếp tuổi của dân cư ở đây theo thứ tự từ nhỏ đến lớn. Biết rằng tuổi ở đây chỉ nằm trong khoảng từ 1 đến 100.

Tuổi của dân cư được cho vào từ file TUOI.INP viết liên tiếp nhau, cách nhau ít nhất một dấu cách hoặc một dấu xuống dòng. Kết quả đưa ra file TUOI.OUT cấu trúc như file vào nhưng đã được sắp xếp.

Ví dụ:

TUOI.INP	TUOI.OUT
1 2 20 4 45 62 3 2 4 55 2 4 5 100	1 2 2 2 3 4 4 4 5 20 45 55 62 100

Ý tưởng

Đối với bài toán này thì ta không thể áp dụng các thuật toán sắp xếp thông thường được vì số lượng phần tử là quá lớn (không thể lưu trữ vào mảng vì sẽ tràn bộ nhớ). Ở đây ta nhận xét tuổi của dân cư là một số nguyên nằm trong khoảng từ 0 đến 100. Do vậy ta áp dụng thuật toán sắp xếp Pigeonhole như sau:

Chuẩn bị một mảng 101 phần tử từ 0 đến 100 kiểu Longint để lưu trữ số người cùng độ tuổi. Tiến hành đọc lần lượt từ file dữ liệu tuổi của dân cư, gặp độ tuổi nào thì tăng phần tử tương ứng của mảng lên 1. Sau khi đọc hết file dữ liệu ta sẽ được một mảng lưu giữ số lượng người cùng độ tuổi. Công việc cuối cùng bây giờ chỉ là ghi ra file kết quả lần lượt các số theo từng độ tuổi được lưu giữ trong mảng trên.

MIC2C. BÀI 31. Xếp việc

Bắt đầu từ thời điểm 0, một người làm N việc với số hiệu từ 1 đến N, $N \leq 200$. Với $1 \leq i \leq N$, việc i cần làm trong $T[i]$ đơn vị thời gian và mỗi đơn vị thời gian từ thời điểm 0 đến lúc bắt đầu làm nó, người đó bị phạt một lượng tiền $C[i]$. Khi đã làm một việc nào thì phải làm xong mới chuyển sang việc làm khác. Hãy thu xếp trình tự các việc làm sao cho tổng số tiền bị phạt là ít nhất.

Dữ liệu vào được cho bởi file XEPVIEC.INP trong đó dòng thứ nhất ghi số nguyên dương N. Với $1 \leq i \leq N$, dòng thứ i+1 ghi hai số thực $T[i]$ và $C[i]$.

Dữ liệu ra ghi trong file XEPVIEC.OUT. Dòng thứ nhất ghi tổng số tiền bị phạt. Từ dòng thứ hai ghi mỗi dòng 1 cặp số: số thứ nhất là số hiệu việc, số thứ hai là thời điểm bắt đầu làm việc. Trình tự từ trên xuống dưới là trình tự lần lượt làm các việc.

Ý tưởng: Sắp xếp các công việc theo trình tự không tăng của tỷ số $T[i] / C[i]$, đó là trình tự thực hiện tốt nhất.

MIC2C. BÀI 32. Dây số

(câu 1, HSG Lào Cai, 2012-2013)

✖

✖

Ví dụ:

DAYSO.INP	DAYSO.OUT
5	2
2 4 6 8 10	2
	2

DAYSO.INP	DAYSO.OUT
5	19
4 7 11 15 19	1
	0

Ý tưởng: Sử dụng các tính chất của đề bài để giải quyết bài toán.

MIC2C. BÀI 33. Số thân thiện

(câu 2, HSG An Giang, 2013-2014, VI)

Viết chương trình nhập vào một số k ($k \leq 200$), tìm tất các số thân thiện từ 1 đến k , biết rằng: **số thân thiện** là số tự nhiên có hai chữ số mà khi đảo trật tự của hai chữ số đó sẽ thu được một số nguyên tố cùng nhau với số đã cho.

*** Ghi chú:** Hai số tự nhiên m và n được gọi là nguyên tố cùng nhau khi và chỉ khi ước số chung lớn nhất của m và n bằng 1.

Ví dụ: $k = 25$, có 8 số thân thiện là: 10, 13, 14, 16, 17, 19, 23, 25,

$k = 34$, có 12 số thân thiện là: 10, 13, 14, 16, 17, 19, 23, 25, 29, 31, 32, 34,

Ý tưởng: Xây dựng CTC đảo ngược số và CTC tìm UCLN của hai số rồi duyệt các số từ 10 đến k để đếm số lượng số thân thiện.

MIC2C. BÀI 34. Sa mạc

Sa mạc là lưới ô vuông cấp $M \times N$ ($1 \leq N, M \leq 100$). Trên mỗi ô của lưới người ta ghi một số nguyên a ($1 \leq a \leq 100$) được gọi là năng lượng của ô đó. Một con lạc đà đang ở ô (i, j) của lưới chỉ được đi đến một trong hai ô $(i+1, j)$ hoặc ô $(i, j+1)$. Lạc đà đi đến ô nào thì hấp thụ được nguồn năng lượng tại ô đó. Hãy tìm cho lạc đà một đường đi từ ô $(1, 1)$ đến ô (M, N) theo nguyên tắc trên và hấp thụ được nhiều năng lượng nhất.

Dữ liệu vào: Từ file văn bản SAMAC.INP, dòng đầu tiên ghi 2 số nguyên dương theo thứ tự M, N .

Dòng thứ i trong M dòng tiếp theo ghi N số nguyên dương, số thứ j là năng lượng trên ô (i, j) của sa mạc (số thứ tự của các số trên một dòng tính từ trái qua phải)

Kết quả: Ghi ra file văn bản SAMAC.OUT, dòng đầu tiên ghi số S là năng lượng mà lạc đà hấp thụ được. Từ dòng thứ 2 trở đi mỗi dòng ghi 2 số nguyên dương là tọa độ các ô theo thứ tự trên đường đi của lạc đà.

Ví dụ:

SAMAC.INP	SAMAC.OUT
4 5	41
1 5 1 3 4	1 1
6 7 9 1 5	2 1
1 1 8 4 1	2 2
1 3 4 3 3	2 3
	3 3
	3 4
	4 4
	4 5

Cả hai file dữ liệu, các số trên một dòng cách nhau một kí tự trắng.

Ý tưởng:

Cách 1: Trong bài này để thuận tiện cho viết chương trình ta sẽ sử dụng kĩ thuật tạo rào khi duyệt mảng như sau:

- Dùng mảng A có kích thước $(m+1) \times (n+1)$, trong đó $A_{m \times n}$ sẽ chứa các giá trị trên lưới. Dòng thứ $m+1$ và cột $n+1$ sẽ lưu các số 0 để làm hàng rào trong quá trình đi của lạc đà.
- Ta thấy, tại ô (i,j) thì lạc đà chỉ đi sang 1 trong 2 ô:
 - o Nếu năng lượng tại ô $(i+1,j) \geq$ ô $(i,j+1)$ thì lạc đà sẽ đi đến ô $(i+1,j)$
 - ngược lại lạc đà sẽ đi đến ô $(i,j+1)$ (xem hình)

		j →				
	i	1			n	n+1
1	1	1	5	1	3	0
		6	7		1	5
		1	1	8	4	1
m		1	3	4	3	3
m+1		0	0	0	0	0

Mảng A

1	0	0	0	0	0
1	1	1	0	0	0
0	0	1	1	0	0
0	0	0	1	1	0
0	0	0	0	0	0

Mảng B

- Do đó nếu đi đến ô nào ta sẽ cộng giá trị tại ô đó vào tổng và ghi nhận lại vị trí của ô (ta sẽ dùng một mảng B có kích thước $m \times n$ để làm điều đó, khởi tạo các giá trị của mảng B đều là 0, sau đó ô nào đi qua thì ta ghi nhận giá trị là 1).
- Như vậy khi duyệt toàn bộ mảng chứa dữ liệu input ta sẽ có năng lượng lớn nhất mà lạc đà hấp thu được.
- Sau đó ta chỉ cần duyệt mảng B, ô nào có giá trị 1 thì ta ghi chỉ số đó vào tệp output.

Cách 2: Sử dụng phương pháp quy hoạch động trên mảng hai chiều

$B[1,i] = B[1,i-1] + A[1,i], i=1, 2, \dots, n$

$B[i,1] = B[i-1,1] + A[i,1], i=1, 2, \dots, m$

$B[i,j] = \max(B[i-1,j], B[i,j-1]) + A[i,j]$, với $i=2, 3, \dots, m$ và $j=2, 3, \dots, n$

MIC2C. BÀI 35. Vườn trường

Vườn trường là một hình chữ nhật gồm một số khoảng trồng các loại cây khác nhau. Nó được mô tả là một lưới ô vuông sao cho mỗi ô của lưới được xem như chỉ có 2 trạng thái: hoặc là diện tích trồng cây, hoặc không phải. Hướng ứng cuộc vận động xây dựng trường "Xanh - Sạch - Đẹp" học sinh khối 12 muốn quét vôi xung quanh các bức tường rào của các khoảng vườn này. Mỗi cạnh ô vuông của lưới được quét vôi nếu nó là cạnh chung của 2 ô khác trạng thái (các cạnh thuộc biên của lưới không được tính). Lập trình tính tổng chiều dài cần quét vôi của các khoảng vườn (theo đơn vị cạnh ô lưới).

Dữ liệu vào: Đọc từ file văn bản VUON.INP gồm:

- Dòng đầu ghi hai số nguyên dương M, N ($M, N \leq 200$ lần lượt là số dòng và cột của lưới).
- Dòng thứ i trong số M dòng tiếp mô tả trạng thái của N ô lưới tương ứng của dòng i gồm N số: 0 (là đất trống) hoặc 1 (là diện tích trồng cây) theo đúng thứ tự các ô trong lưới.

Kết quả: Ghi ra file văn bản VUON.OUT gồm một dòng ghi giá trị tổng chiều dài cần quét vôi.

Ví dụ:



Ý tưởng

Ta dùng một mảng $A[0..m+1, 0..n+1]$, trong đó $A[1..m, 1..n]$ để lưu sơ đồ vườn trường.

Vì xung quanh khu vườn không quét vôi nên ta dùng kĩ thuật tạo rào xung quanh khu vườn để thuận tiện khi xử lí bài toán:

+ Dòng 0 nhận giá trị dòng 1; dòng $m+1$ nhận giá trị dòng m

+ Cột 0 nhận giá trị cột 1; cột $n+1$ nhận giá trị cột n .

Ta duyệt toàn bộ mảng $A[1..m, 1..n]$, ô (i,j) ta kiểm tra điều kiện sau:

+ Nếu ô (i,j) có trạng thái khác ô $(i,j-1)$ thì tăng diện tích cần quét vôi lên 1.

+ Nếu ô (i,j) có trạng thái khác ô $(i-1,j)$ thì tăng diện tích cần quét vôi lên 1.

Khi duyệt xong ta có kết quả của diện tích cần quét vôi.

	0	0	0	1	1	0	0	0	0	0	0	
--	---	---	---	---	---	---	---	---	---	---	---	--

0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	1	0	0	0	0
0	0	0	0	1	0	0	1	1	1	1	0	0
0	0	1	0	0	0	1	1	0	1	1	0	0
0	0	1	1	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	
	0	0	0	0	0	0	1	1	1	1	1	

← Rào

M1C2C. BÀI 36. Kho an toàn

Nhà máy sản xuất vũ khí có nhiều kho hàng và các kho hàng có lính canh gác. Các kho hàng và lính canh được thể hiện trên ma trận m dòng, n cột. Kho hàng được gọi là *Kho an toàn* nếu xung quanh các kho đó có 8 lính canh.

Hãy lập trình đưa ra vị trí các *Kho an toàn*.

Dữ liệu vào: từ tệp văn bản ANTOAN.INP

Dòng đầu tiên là hai số nguyên m và n ($0 < m, n \leq 100$).

m dòng tiếp theo mỗi dòng có n số 0 hoặc 1 thể hiện kho hàng và lính canh (kho hàng kí hiệu: 0, lính canh kí hiệu: 1).

Các số trên cùng một dòng ghi cách nhau một dấu cách.

Kết quả: Ghi ra tệp văn bản ANTOAN.OUT

Nếu có *Kho an toàn* thì ghi vị trí của các kho đó (chỉ số dòng và chỉ số cột).

Nếu không có *Kho an toàn* thì ghi -1.

Các số trên cùng một dòng ghi cách nhau một dấu cách.

Ví dụ:

ANTOAN.INP	ANTOAN.OUT
4 5	2 2
1 1 1 0 1	3 4
1 0 1 1 1	
1 1 1 0 1	
0 0 1 1 1	

Ý tưởng

- Đọc số dòng và số cột của ma trận vào hai biến m, n .
- Dùng mảng A , có kiểu các phần tử là kiểu lô-gic. Vì các phần tử của ma trận là 0 hoặc 1 nên khi đọc nếu giá trị phần tử là 0 thì ta ghi vào mảng giá trị false và nếu là 1 thì ghi là true (lưu như thế để khi xử lý được thuận tiện hơn)
- Ta sẽ duyệt qua các phần tử của ma trận từ ô (2,2) đến ô ($m-1, n-1$) (ta sẽ không xét các ô ở bìa ma trận vì chắc chắn các kho (nếu có) sẽ là kho không an toàn), với mỗi ô có giá trị false (kho) thì ta sẽ kiểm tra các ô xung quanh, nếu các ô xung quanh đều cho giá trị true thì kho đó là an toàn. Khi đó ta sẽ ghi chỉ số của ô vào tệp output.

T	T	T	F	T
T	F	T	T	T
T	T	T	F	T
F	F	T	T	T

Ghi chú: T: true; F: false

M1C2C. BÀI 37. Bố trí xe

Vừa qua kì thi máy tính cầm tay cấp quốc gia được tổ chức tại nam Định, Ban tổ chức đã bố trí xe ô tô để đưa mỗi đoàn học sinh của mỗi tỉnh đi tham quan các địa điểm khác nhau. Có tất cả n đoàn học sinh đánh số từ 1 đến n , đoàn thứ i cần đi tới địa điểm cách nơi ở là d_i đơn vị (coi như d_i là khoảng cách tính theo chiều cả đi lẫn về). Có m chiếc xe ô tô sử dụng được đánh số từ 1 đến m ($m \geq n$), có thể dùng để phục vụ đưa các đoàn đi tham quan. Được biết xe thứ j có mức tiêu thụ xăng là v_j đơn vị thể tích xăng trên một đơn vị độ dài. Hãy lựa chọn và bố trí n xe, mỗi xe chỉ phục vụ một đoàn theo yêu cầu sao cho tổng thể tích xăng cần thiết là ít nhất.

Dữ liệu: Vào từ tệp văn bản XE.INP có cấu trúc:

- Dòng 1: Chứa 2 số nguyên dương n, m ($1 \leq n \leq m \leq 200$);
- Dòng 2 chứa n số d_1, d_2, \dots, d_n ;
- Dòng 3: Chứa m số v_1, v_2, \dots, v_m . (Các giá trị d_i, v_j đều nguyên dương và không quá 32000).

Kết quả: Đưa ra tệp văn bản XE.OUT, chứa 2 dòng

- Dòng 1: Chứa số lượng xăng cần ít nhất theo yêu cầu;

- Dòng 2: chứa n số x_1, x_2, \dots, x_n thể hiện đoàn thứ i được bố trí đi xe x_i ;

Ví dụ:

XE.INP	XE.OUT
3 5	29
3 2 4	2 5 3
2 7 1 4 2	

Ý tưởng

1. Dùng mảng $Val[1..2, 1..200]$, trong đó dòng 1 lưu khoảng cách từng đoàn cần tham quan, dòng 2 lưu lượng xăng tương ứng của từng xe.
2. Dùng mảng $Pos[1..2, 1..200]$, trong đó dòng 1 lưu vị trí ban đầu của các đoàn, dòng 2 lưu vị trí ban đầu của từng xe.
3. Sắp xếp dòng 1: tăng dần theo khoảng cách các đoàn cần tham quan ($Val[1, i], i=1, \dots, n$)
4. Sắp xếp dòng 2: tăng dần theo lượng xăng tiêu thụ của từng xe ($Val[2, i], i=1, \dots, m$)
5. Tổng mức xăng tiêu thụ:
6. $T = \sum_{i=1}^n Val[1, n-i+1] * Val[2, i]$ (đoàn đi xa nhất sẽ đi xe tốn ít xăng nhất)
7. Các xe đã được sử dụng là: $Pos[2, i]$ với $i=n, n-1, \dots, 1$

MIC2C. BÀI 38. Kết bạn

Tại cuộc cắm trại, Tuấn muốn tìm hiểu một số người bạn nữa nhưng không phải người nào Tuấn cũng muốn kết bạn. Quy tắc đặt ra của Tuấn là người mà Tuấn muốn kết bạn phải là bạn của ít nhất 2 người bạn mà Tuấn đã chơi cùng. Ví dụ Nam và Hùng là bạn của Tuấn, Tuấn sẽ kết bạn với Bảo nếu Bảo là bạn của Hùng và Nam

Yêu cầu: Cho biết danh sách của các người bạn. Hãy cho biết Tuấn có thể kết bạn với nhiều nhất bao nhiêu người tại cuộc cắm trại đó

Dữ liệu vào: Từ tệp văn bản **KETBAN.INP**

- Dòng 1 ghi N-người tại buổi cắm trại ($0 < N < 10000$) mỗi người được đánh số từ 1 đến N. Tuấn được đánh số 1

- Dòng thứ i ghi trong N dòng tiếp theo ghi danh sách những người bạn của người thứ i

Kết quả: ghi ra tệp văn bản **KETBAN.OUT** gồm 1 số duy nhất là số bạn nhiều nhất mà Tuấn có thể kết bạn.

Ví dụ

KETBAN.INP	KETBAN.OUT
8	2
3 4 6 7	
3 7	
1 2 8	
1 5	
4 6	
1 5	
1 2	
3	

Giải thích: Tuấn kết bạn thêm với 2 người là người số 5 và số 2

Ý tưởng:

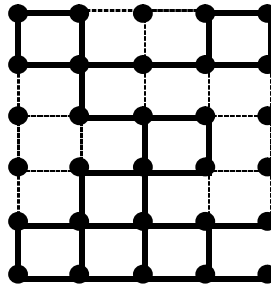
- Đọc danh sách bạn vào mảng $A[i, j]$
- Tạo danh sách bạn của Tuấn là $B[j] = A[1, j]$ (với $A[1, j] > 0$)
- Kiểm tra danh sách từ bạn 2 đến bạn thứ N. Nếu có ít nhất 2 bạn nào có mặt trong danh sách $B[j]$ thì ta lưu vào mảng C và đếm cho người này

MIC2C. BÀI 39. Đếm số ô vuông đơn vị

Cho một bảng lưới hình chữ nhật gồm $m \times n$ điểm (m hàng ngang, n cột dọc) nằm trên các mắt lưới ô vuông. Các điểm liên kề trên cùng một hàng hoặc một cột có thể nối với nhau bởi một đoạn thẳng có độ dài bằng 1 đơn vị.

Trên mỗi hàng có nhiều nhất $n-1$ đoạn thẳng nối các điểm kề nhau theo hàng ngang, trên mỗi cột có nhiều nhất là $m-1$ đoạn thẳng nối các điểm kề nhau theo cột dọc. Các đoạn nối đó có thể sẽ tạo ra các ô vuông trên bảng lưới (chỉ xét đến các ô vuông có diện tích bằng 1 đơn vị vuông).

Ví dụ, bảng lưới bên dưới có 6 hàng ngang, mỗi hàng 5 mắt lưới (5 điểm) và 5 cột dọc, mỗi cột 6 mắt lưới (6 điểm). Trong trường hợp này, các đoạn thẳng nối các điểm đã tạo nên 7 ô vuông.



Để mô tả bảng lưới, người ta dùng hai mảng nhị phân: một mảng (*mảng ngang*) diễn tả các đoạn thẳng nằm ngang, một mảng (*mảng dọc*) diễn tả các đoạn thẳng thẳng đứng.

1	0	0	1
1	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1
1	1	1	1

Mảng ngang

1	1	0	1	1
0	1	0	0	0
0	0	1	1	0
0	1	1	0	0
1	1	1	1	0

Mảng dọc

Trong các mảng, số 1 diễn tả có đoạn thẳng nối hai điểm kề nhau, số 0 diễn tả không có đoạn thẳng nối hai điểm liền kề.

Lập trình đếm số các ô vuông (chỉ xét đến các ô vuông có diện tích bằng 1 đơn vị vuông) được tạo bởi các đoạn nối có trên bảng lưới đã cho.

Dữ liệu vào: 3 file, gồm: KTHUOC.INP, NGANG.INP, DOC.INP.

Dữ liệu ra: file OVUONG.OUT chứa số ô vuông.

Ví dụ: Như hình vẽ trên ta có

KTHUOC.INP	NGANG.INP	DOC.INP	OVUONG.OUT
6 5	1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1	1 1 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0	7

M1C2C. BÀI 40. Chuyển vị ma trận

Cho bảng vuông A có kích thước $n \times n$ ($1 < n \leq 40$) có giá trị nguyên. Viết chương trình: chuyển hàng thành cột và ngược lại.

Dữ liệu vào: cho trong tập tin văn bản HV.INP

+ Dòng 1: Chứa số nguyên n

+ n dòng tiếp theo: mỗi dòng chứa n số nguyên, mô tả một hàng của bảng A, các số trên cùng dòng cách nhau ít nhất 1 khoảng trắng.

Kết quả ra: ghi vào tập tin văn bản HV.OUT, bảng A đã được chuyển vị với cách tổ chức thông tin giống như tệp dữ liệu vào HV.INP

Ví dụ:

HV.INP	HV.OUT	HV.INP	HV.OUT
4		3	
1 2 3 4	1 5 9 13	1 5 9	1 2 4
5 6 7 8	2 6 10 14	2 6 10	5 6 8
9 10 11 12	3 7 11 15	4 8 12	9 10 12
13 14 15 16	4 8 12 16		

Ý tưởng: chuyển hàng thành cột và ngược lại, công thức: $B_{ij} = A_{ji}$ ($i=1,...,m; j=1,...,n$)

M1C2C. BÀI 41. Sắp xếp trên ma trận

Trên bảng chữ số gồm M dòng N cột các phần tử nguyên $a[i,j]$. Hãy sắp xếp các phần tử theo chiều tăng dần trên mỗi hàng từ trái qua phải, sau đó sắp xếp các phần tử theo chiều tăng dần trên mỗi cột từ trên xuống dưới.

Ví dụ:

SORT.INP	SORT.OUT
4 5	
3 9 1 36 12	1 3 7 12 15
6 4 7 34 12	4 6 8 12 32
6 7 22 32 8	4 7 9 13 34
7 4 9 15 13	6 7 9 22 36

Dữ liệu vào: Từ tệp văn bản SORT.INP gồm:

- Dòng 1: Gồm hai số nguyên M và N ($3 \leq M, N \leq 100$).
- Mỗi dòng trong M dòng tiếp theo có N số nguyên $a[i,j]$, mỗi số ghi cách nhau một dấu cách.

Dữ liệu ra: Ghi ra tệp văn bản SORT.OUT gồm M dòng, N cột là các phần tử được sắp xếp như trên.

MIC2C. BÀI 42. Bài toán cái túi

(câu 3, HSG Bạc Liêu, 2011-2012, bảng A, ngày 2)

Cho n đồ vật. Trọng lượng và giá trị của vật i lần lượt là $A[i]$, $B[i]$. Hãy chọn ra một số vật sao cho tổng trọng lượng của chúng không vượt quá trọng lượng M cho trước và tổng giá trị là lớn nhất. Cho biết $0 < n \leq 20$, $0 < M \leq 100$, $0 < A[i], B[i] < 256$

Dữ liệu vào: Cho trong file NHAP.INP gồm n + 1 dòng:

- Dòng đầu là 2 số n, M;
- Dòng i + 1 ($1 \leq i \leq n$) ghi 2 số nguyên dương $A[i]$, $B[i]$.

Dữ liệu ra: Ghi vào file XUAT.OUT gồm:

- Mỗi dòng ghi 3 số: i, $A[i]$, $B[i]$;
- Dòng cuối là 3 số: Tổng số vật, tổng trọng lượng và tổng giá trị của các vật được chọn. Các số ghi trên cùng một dòng được cách ít nhất một dấu cách.

Ý tưởng

Hàm mục tiêu f: tổng giá trị của vali.

Nhận xét: giá trị của vali phụ thuộc vào 2 yếu tố: có bao nhiêu vật đang được xét và trọng lượng của các vật. Do đó bảng phương án sẽ là bảng 2 chiều. $L[i,j]$: tổng giá trị lớn nhất của vali khi xét từ vật 1...vật i và trọng lượng của vali chưa vượt quá j. Chú ý rằng khi xét đến $L[i,j]$ thì các giá trị trên bảng phương án đều đã được tối ưu.

Cách tính: Tính $C[i,j]$: vật đang xét là ai với trọng lượng của vali không được quá j. Có 2 khả năng xảy ra:

- + Nếu chọn a_i đưa vào vali, trọng lượng vali trước đó phải $\leq j - a[i]$. Vì mỗi vật chỉ được chọn 1 lần nên giá trị lớn nhất của vali lúc đó là $C[i-1, j-a[i]] + b[i]$
 - + Nếu không chọn $a[i]$, trọng lượng của vali là như cũ (như lúc trước khi chọn a_i): $C[i-1, j]$
- Tóm lại ta có $C[i,j] = \max(C[i-1, j-a[i]] + b[i], C[i-1, j])$

MIC2C. BÀI 43. Trò chơi (dạng bài toán cái túi - chọn 1 hoặc nhiều vật)

Trong cuộc thi SV2012, các nhà tài trợ trao cho ban tổ chức N phần thưởng. Mỗi phần thưởng thuộc loại thứ i có khối lượng là $A[i]$ và giá trị $C[i]$. Số lượng các phần thưởng của mỗi loại không hạn chế. Đội chơi nào chiến thắng trong cuộc thi sẽ được ban tổ chức tặng một cái túi dùng để đựng các phần thưởng và cái túi có thể mang được tối đa khối lượng W. Hỏi các bạn trong đội chơi cần chọn những phần thưởng nào để cho tổng giá trị của các phần thưởng đã chọn là lớn nhất nhưng khối lượng của chúng không vượt quá khối lượng W. Biết mỗi loại phần thưởng có thể hoặc không chọn phần thưởng nào, hoặc chọn một phần thưởng, hoặc chọn nhiều phần thưởng.

Dữ liệu vào: Từ tệp văn bản BAG.INP gồm:

- Dòng 1: Ghi hai số nguyên N và W ($1 \leq N, W \leq 1000$).
- N dòng tiếp theo, dòng thứ i ghi hai số nguyên $A[i]$ và $C[i]$ ($1 \leq A[i], C[i] \leq 1000$).

Dữ liệu ra: Ghi ra tệp văn bản BAG.OUT gồm:

- Dòng 1: Ghi tổng giá trị phần thưởng lớn nhất được chọn.
- Các dòng tiếp theo ghi chỉ số và số lượng của các loại phần thưởng đã chọn.

Ví dụ:

BAG.INP	BAG.OUT
5 13	19
3 4	1 1
4 5	4 5
5 6	
2 3	
1 1	

Giải thích ví dụ: Chọn 1 phần thưởng loại 1, chọn 5 phần thưởng loại 4. Tổng giá trị phần thưởng lớn nhất được chọn là:
 $1 \times 4 + 3 \times 5 = 19$.

MIC2C. BÀI 44. Tổng các hàng của ma trận

(Câu 1 - đề thi HSG tỉnh Bạc Liêu năm học 2011-2012, bảng A, ngày 1)

Đọc từ một ma trận A cấp $m \times n$ các số nguyên, với các phần tử được cho từ một tệp văn bản MATRAN.INP, với mỗi phần tử cách nhau một dấu cách. Trong đó, hàng đầu tiên ghi hai số m, n lần lượt là số hàng và số cột của ma trận, m hàng tiếp theo ghi các hàng của ma trận.
 Tính tổng các số hạng trên mỗi hàng của ma trận và ghi kết quả vào tệp văn bản MATRAN.OUT

Ví dụ:

MATRAN.INP	MATRAN.OUT
3 4	3 4
1 3 5 7	1 3 5 7 16
2 4 6 8	2 4 6 8 20
0 3 2 6	0 3 2 6 11

Ý tưởng: Ta chỉ cần duyệt từng hàng và tính tổng các phần tử.

M1C2C. BÀI 45. Ma trận số

(Câu 1 - đề thi HSG tỉnh Bạc Liêu năm học 2011-2012, bảng B, ngày 2)

Cho một số nguyên dương n bất kì ($n \leq 20$).

Hãy viết chương trình tạo mảng $N \times N$ phần tử nguyên dương (giá trị lớn nhất trong bảng là $2 \cdot N$) theo quy luật cho trong ví dụ sau với $N = 7$

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	2	4	6
4	8	12	2	4	6	8
5	10	2	4	6	8	10
6	12	2	4	6	8	10
7	14	2	4	6	8	10

- Dữ liệu vào: Giá trị của N được cho trong tệp tin MATRAN.TXT

- Kết quả ra: Ghi trong tệp tin MATRAN.TXT mảng kết quả

M1C2C. BÀI 46. Sắp xếp ma trận

(Câu 2, HSG Đà Nẵng 2001-2002, vòng 1)

Cho ma trận $N \times M$ chứa các số tự nhiên.

Hãy sắp xếp lại các phần tử trong ma trận sao cho:

$A[i, 1] \leq A[i, 2] \leq \dots \leq A[i, M]$

$A[1, j] \leq A[2, j] \leq \dots \leq A[N, j]$

Dữ liệu vào: trong file INP.TXT gồm $N + 1$ dòng:

+ Dòng đầu tiên ghi hai số N và M ($N, M \leq 20$)

+ Dòng $i + 1$ ($1 \leq i \leq N$) ghi M số $A[i, 1], A[i, 2], \dots, A[i, M]$

Các số ghi trên cùng một dòng cách nhau ít nhất một khoảng trắng

Dữ liệu ra: In ra màn hình ma trận đã được sắp xếp.

Ví dụ:

INP.TXT	KẾT QUẢ RA MÀN HÌNH
5 8	
1 3 9 8 3 2 4 5	1 1 1 1 1 1 1 1
5 2 4 1 6 1 7 9	2 2 2 2 2 3 3
4 3 3 4 1 2 3 2	3 3 3 3 3 4 4
5 3 8 1 6 3 5 4	4 4 4 4 5 5 5 5
8 2 1 2 1 1 3 4	6 6 7 8 8 8 9 9

Ý tưởng: ta có thể chuyển mảng hai chiều về mảng một chiều, sắp xếp dữ liệu tăng dần sau đó ghi kết quả vào tệp dưới dạng mảng hai chiều.

M1C2C. BÀI 47. Sắp xếp mảng hai chiều tăng theo hàng, cột

(Câu 2, HSG Long An, lớp 12 Bảng B, vòng 1 2012-2013)

Cho mảng 2 chiều $A[1..N, 1..M]$ chứa các số nguyên. Hãy sắp xếp lại các giá trị của các ô trong A sao cho:

$A[i, 1] \leq A[i, 2] \leq \dots \leq A[i, M]$

$A[1, j] \leq A[2, j] \leq \dots \leq A[N, j]$

Và tính tổng giá trị các phần tử trên cùng một hàng.

Dữ liệu: cho file SORT.INP gồm $n+1$ dòng

Dòng 1: chứa 2 số M và N

Dòng $i+1$ ($1 \leq i \leq N$): ghi M số $A[i, 1], A[i, 2], \dots, A[i, M]$

Các số ghi trên cùng một dòng cách nhau ít nhất một dấu cách.

Kết quả: ghi vào file SORT.OUT có dạng:

Dòng 1: chứa 2 số M và N

Dòng i+1 (1 ≤ i ≤ N): ghi M số A[i,1], A[i,2], ..., A[i,N] sau khi đã sắp xếp

Ví dụ

SORT.INP	SORT.OUT
<pre> 5 8 1 3 5 7 9 2 4 6 1 2 3 4 5 6 7 8 2 4 6 8 10 12 14 16 9 8 7 6 5 4 3 2 1 2 1 3 1 4 1 5 </pre>	<pre> 5 8 1 1 1 1 1 2 2 2 2 2 3 3 3 3 4 4 4 4 4 5 5 5 5 6 6 6 6 7 7 7 8 8 8 9 9 10 12 14 16 tong gia tri dong 1 la : 10 tong gia tri dong 2 la : 22 tong gia tri dong 3 la : 36 tong gia tri dong 4 la : 53 tong gia tri dong 5 la : 86 </pre>

Ý tưởng

Thực hiện tương tự bài sắp xếp ma trận, cần ghi dữ liệu theo đúng dạng bài toán yêu cầu.

MIC2C. BÀI 48. Rút tiền từ máy ATM

Một điểm giao dịch của ngân hàng X có N loại tiền mệnh giá từ A[1], A[2], A[3], . . . , A[N] (đơn vị ngàn đồng) với số lượng tiền mỗi loại không giới hạn. Một khách hàng cần rút với số tiền là M (ngàn đồng). Hãy cho biết cần bao nhiêu tiền mỗi loại để chi trả sao cho số tờ là ít nhất.

Cho biết: $N \leq 9$; $A[i] \leq 500$; $M \leq 10000$

Dữ liệu vào: Cho trong file **INP.TXT** gồm 2 dòng:

- Dòng đầu là 2 số N, M;
- Dòng thứ hai ghi N số nguyên dương A[1], A[2], A[3], . . . , A[N]

Dữ liệu ra: Ghi vào file **OUT.TXT** gồm:

- Dòng đầu ghi số lượng tờ phải trả;
- Dòng thứ hai ghi N số nguyên không âm ứng với số tờ cần trả cho mỗi loại tiền. Các số ghi trên cùng một dòng được cách ít nhất một dấu cách.

Ý tưởng:

- Ghi nhận vị trí ban đầu của các loại mệnh giá tiền (mảng vt)
- Sắp xếp mảng A giảm dần theo giá trị các loại tiền. (Dùng Quicksort)
- Sử dụng giải thuật tham lam, với số tiền lớn nhất chưa lấy ta sẽ lấy tối đa số tờ có thể lấy được.

$$soto = \frac{sotien}{giatri}$$

MIC2C. BÀI 49. Đóng gói sản phẩm

(Câu 2, đề thi HSG tỉnh Đồng Tháp năm học 2012-2013)

Một nhà máy xay xát cần đóng gói gạo vào các loại bao 100kg, 50kg, 20kg, 10kg, 5 kg. Với mỗi loại bao chỉ được đóng gói khi đủ số lượng quy định cho loại bao đó. Tuy nhiên công ty sản xuất các loại bao thì cung cấp số lượng theo từng đợt với số lượng từng loại khác nhau.

Yêu cầu: Với mỗi khối lượng gạo và số lượng bao của từng loại cho trước, hãy tìm một phương án đóng gói sao cho số lượng gạo thừa không được đóng gói (nếu có) là ít nhất.

Dữ liệu vào: cho từ file văn bản DONGGOL.INP gồm 2 dòng.

- Dòng đầu tiên ghi số lượng gạo N (đơn vị kg) cần phải đóng gói ($0 < N < 1000000$).
- Dòng thứ hai ghi 5 số tự nhiên cho biết số lượng bao của từng loại theo thứ tự 100kg, 50kg, 20kg, 10kg, 5 kg (số lượng bao mỗi loại không vượt quá 200000)

Kết quả ra: ghi ra file văn bản DONGGOL.OUT gồm 2 dòng

- Dòng đầu tiên ghi số gạo thừa không được đóng gói. Nếu không còn thừa thì ghi số 0.
- Dòng thứ hai ghi 5 số nguyên dương tương ứng với số lượng bao đã sử dụng của từng loại theo thứ tự 100kg, 50kg, 20kg, 10kg, 5 kg.

Ví dụ:

DONGGOL.INP	DONGGOL.OUT
1000	0
5 0 20 30 45	5 0 20 10 0
DONGGOL.INP	DONGGOL.OUT
950	10
10 0 20 0 0	9 0 2 0 0

Ý tưởng: Thực hiện tương tự bài toán “Rút tiền từ máy ATM”

MIC2C. BÀI 50. Hệ thống cảnh báo thảm họa

(DẠNG TOÁN HCN TỐI ĐẠI TRONG MA TRẬN)

Để giảm bớt thiệt hại khi có rủi ro do động đất và sóng thần gây ra, Trung tâm dự báo thảm họa thiên tai X muốn xây dựng hệ thống cảnh báo động đất và sóng thần ở các vùng biển. Vệ tinh VNSAT đã cung cấp bản đồ khảo sát độ sâu đáy biển của các vùng biển. Bản đồ là lưới ô vuông gồm M dòng và N cột, mỗi ô trên bản đồ ghi một số nguyên dương cho biết độ sâu của đáy biển tại điểm khảo sát. Trung tâm X cần tìm trên bản đồ một hình chữ nhật có các cạnh song song với các cạnh của bản đồ và các ô trong hình chữ nhật đó có cùng một độ sâu để xây dựng hệ thống cảnh báo thảm họa.

Yêu cầu: Hãy tìm trên bản đồ một hình chữ nhật có số lượng ô nhiều nhất và có cùng một độ sâu.

Dữ liệu vào: Cho trong file văn bản HTCB.INP có cấu trúc như sau:

- Dòng 1: Ghi hai số nguyên dương M N ($0 < M \leq 65535$; $0 < N \leq 100$).

- M dòng tiếp theo: Mỗi dòng ghi N số nguyên dương là giá trị của các ô trong bản đồ. Các số được ghi cách nhau ít nhất một dấu cách.

Dữ liệu ra: Ghi ra file văn bản HTCB.OUT theo cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương K là số lượng ô của hình chữ nhật tìm được.

Ví dụ:

HTCB.INP	HTCB.OUT
4 5 1 2 1 3 2 3 2 2 2 1 1 2 2 2 1 1 3 3 2 2	6

Giải thích:

1	2	1	3	2
3	2	2	2	1
1	2	2	2	1
1	3	3	2	2

Giới hạn thời gian thực hiện chương trình không quá 1 phút đối với trường hợp của dữ liệu vào có $M \geq 60000$ và $N \geq 70$ (khoảng 40% số Test).

Chúng ta sẽ xây dựng một thuật toán giải bài toán tổng quát hơn như sau:

Sân bay vũ trụ

Người ta cần xác định một vùng hình chữ nhật ABCD có diện tích lớn nhất và bằng phẳng trên hành tinh Vega để xây dựng sân bay vũ trụ. Bạn được cung cấp một mảnh bản đồ hành tinh Vega, nơi cần xác định vị trí xây dựng một sân bay. Mảnh bản đồ có dạng hình chữ nhật gồm nhiều dòng điểm mã số từ 1 trở đi, mỗi dòng có đúng M điểm mã số từ 1 đến M. Mỗi điểm được tô một màu thể hiện độ cao của điểm đó. Yêu cầu: xác định hình chữ nhật ABCD chứa nhiều điểm đồng màu nhất.

Dữ liệu vào: Tập văn bản CNMAX . INP.

- Dòng đầu tiên: số tự nhiên M, $3 \leq M \leq 70$.
- Tiếp đến là các dòng dài bằng nhau thể hiện một xâu gồm M kí tự là các chữ cái a..z viết liền nhau. Mỗi kí tự biểu diễn cho một màu thể hiện độ cao của điểm tương ứng trên mảnh bản đồ hành tinh Vega. Hai kí tự khác nhau thể hiện hai độ cao khác nhau. Hai điểm cùng độ cao được biểu diễn với cùng một kí tự.
- Số dòng của tập input có thể lên tới 60 nghìn.

Ví dụ:

CNMAX.INP	CNMAX.OUT
20 bccccdddeabcvvvvvvvvb bbbbbbccccccccccbbbbb vvvvvcccccccccccccbb vvcccccccccccccbbbbb pppppccccccccccabbbb pppppcccccccccczzzzz ssccccccccccccczzzzz sssssccccccccccccczz hhhhhcccccccccczzzzz uuuuuuuuczzzzzzzzzzz	80 2 6 9 15

Dữ liệu ra: tập văn bản CNMAX . OUT:

- Dòng đầu tiên: Diện tích của hình chữ nhật tối đại ABCD tìm được.

- Dòng thứ hai: Tọa độ dòng và cột của đỉnh A (ô Tây-Bắc).
- Dòng thứ ba: Tọa độ dòng và cột của đỉnh C (ô Đông-Nam).

Ý tưởng

Do không thể đọc toàn bộ dữ liệu từ tệp **CNMAX.INP** vào một mảng để xử lý nên chúng ta sẽ đọc mỗi lần một dòng vào biến kiểu chuỗi ký tự (string) y . Vì hình chữ nhật ABCD cần tìm chứa cùng một loại ký tự cho nên các dòng của hình sẽ liên thông nhau. Để phát hiện tính liên thông chúng ta cần dùng thêm một biến kiểu chuỗi ký tự x để lưu dòng đã đọc và xử lý ở bước trước. Tóm lại là ta cần xử lý đồng thời hai dòng: x là dòng trước và y là dòng đang xét.

Nếu xét các cột trong hình chữ nhật cần tìm ta thấy chúng phải chứa cùng một loại ký tự. Ta dùng một mảng h với ý nghĩa phần tử $h[i]$ của mảng cho biết tính từ vị trí thứ i của dòng y trở lên có bao nhiêu ký tự giống nhau (và giống với ký tự $y[i]$). Ta gọi $h[i]$ là độ cao của cột i và mảng h khi đó sẽ được gọi là độ cao của dòng đang xét.

Thí dụ, mảng tích lũy độ cao của dòng thứ 5 trong thí dụ đã cho sẽ là:

$h = (1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4, 5, 4, 4, 4, 1, 2, 2, 4, 5)$

Biết độ cao, với hai dòng x và y chúng ta dễ dàng tính được diện tích của mỗi hình chữ nhật ABCD chứa phần tử $y[i]$ trên cạnh DC. Thật vậy, giả sử ta đang xét ký tự thứ $i = 8$ trên dòng thứ 5 như đã nói ở phần trên. Ta có $h[8] = h[7] = h[6] = 4$; $h[9] = h[10] = h[11] = 4$; $h[12] = 5$; $h[13] = h[14] = h[15] = 4$; $h[16] = 1, \dots$ Vậy thì, khi ta đi từ i về hai phía, trái và phải, nếu gặp các ký tự giống ký tự $y[i]$ còn độ cao thì không nhỏ hơn $h[i]$ ta sẽ thu được hình chữ nhật lớn nhất chứa ký tự $y[i]$.

Với dòng thứ 5 là y đang xét, ta có:

$x = \text{'vvccccccccccccccbbbbb'}$; {dòng thu 4 }

$y = \text{'ppppppccccccccccabbbb'}$; {dòng thu 5 }

dòng		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
4	x	v	v	c	c	c	c	c	c	c	c	c	c	c	c	c	b	b	b	b	b
5	y	p	p	p	p	p	c	c	c	c	c	c	c	c	c	c	a	b	b	b	b
	h	1	1	1	1	1	4	4	4	4	4	4	5	4	4	4	1	2	2	4	5

trong đó x chứa dữ liệu của dòng thứ 4, y chứa dữ liệu của dòng thứ 5 trong tệp **CNMAX.INP**.

Từ điểm $i = 8$ dịch chuyển về bên trái ta thu được điểm $c1 = 6$; dịch chuyển về bên phải ta thu được điểm $c2 = 15$. Điều kiện để dịch chuyển là:

- ♦ $(y[c1] = y[i]) \text{ and } (h[c1] \geq h[i])$, nếu qua trái và
- ♦ $(y[c2] = y[i]) \text{ and } (h[c2] \geq h[i])$, nếu qua phải.

Hai thao tác trên được đặt trong hàm tính diện tích của hình chữ nhật ABCD lớn nhất chứa điểm $y[i]$. Hàm cho ra ba giá trị:

- ♦ $c1$: điểm trái nhất hay là tọa độ cột của đỉnh D.
- ♦ $c2$: điểm phải nhất hay là tọa độ cột của đỉnh C.
- ♦ Diện tích của hình.

Thủ tục **Xuli** được thiết kế theo sơ đồ sau:

- Mở tệp dữ liệu **CNMAX.INP**;
- Đọc giá trị chiều dài mỗi dòng vào biến m ;
- Khởi trị:
 - ♦ Con đếm dòng $d := 0$;
 - ♦ Dòng trước x toàn dấu cách;
 - ♦ Mảng chiều cao h toàn 0;
 - ♦ $dtmax := 0$; {diện tích max}
- Lặp cho đến khi hết tệp **CNMAX.INP**:
 - Đọc dòng y ;
 - Tăng con đếm dòng d ;
 - Chỉnh độ cao $h[1..m]$;
 - Xử lý mỗi ký tự $y[i]$ của dòng y ; $i = 1..m$.
 - Tìm diện tích dt của hình chữ nhật lớn nhất chứa phần tử $y[i]$; cho giá trị ra là dt và hai chỉ số đầu trái $c1$ và đầu phải $c2$ thể hiện cạnh đáy của hình chữ nhật.
 - Nếu $dt > dtmax$: chỉnh lại các giá trị

Diện tích max: $dtmax := dt$;

Tọa độ đỉnh A ($Axmax, Aymax$):

$Axmax := d - h[i] + 1$;

$Aymax := c1$;

Tọa độ đỉnh C ($Cxmax, Cymax$):

$Cxmax := d$;

Cymax := c2;

4.5. Sao chép dòng y sang x: **x := y;**

5. Đóng tệp **CNMAX.INP**.

6. Thông báo kết quả.

M1C2C. BÀI 51. Các điểm cực tiểu.

Cho một ma trận A có m hàng và n cột ($3 \leq m, n \leq 100$), một phần tử trên ma trận A được gọi là phần tử cực tiểu nếu nó không lớn hơn các phần tử kề nó, nghĩa là:

$$A[i, j] = \text{Min} \{ A[k, t] / k \in [i-1, i+1], t \in [j-1, j+1] \}$$

Yêu cầu: Hãy viết chương trình tìm tất cả các điểm cực tiểu của ma trận A.

Dữ liệu vào: Cho bởi file text CUCTIEU.INP trong đó:

- Dòng đầu là 2 số m và n.
- M dòng tiếp theo, mỗi dòng i ghi các phần tử trên dòng i của ma trận A ($1 \leq i \leq m$). Các số cách nhau ít nhất một dấu cách.

Kết quả ra: Ghi ra file CUCTIEU.OUT gồm

- Dòng đầu: Ghi số lượng phần tử cực tiểu.
- Các dòng tiếp theo, mỗi dòng i: ghi các vị trí và giá trị của các phần tử cực tiểu trên hàng i.

Ví dụ:

CUCTIEU.INP	CUCTIEU.OUT
4 5	4
1 2 3 4 5	A[1,1]=1
6 7 8 2 4	A[2,4]=2
11 3 13 14 15	A[3,2]=3
16 17 18 4 20	A[4,4]=4

Ý tưởng: Duyệt các phần tử A[i,j] trong ma trận A. So sánh A[i,j] với các phần tử lân cận. Nếu A[i,j] nhỏ hơn tất cả các phần tử lân cận thì đó là phần tử nhỏ nhất.

+ Để làm được việc này thì trước hết phải đọc ma trận A ra mảng 2 chiều m hàng n cột.

* Đọc dòng đầu tiên, số đầu chính là số hàng của mảng A; số tiếp theo là số cột của mảng A.

* Đọc từng dòng trong tệp văn bản. Mỗi dòng trong tệp là một dòng trong mảng.

+ Sau khi đọc từ tệp ra mảng. Ta tiến hành tìm các phần tử cực tiểu từng hàng theo các bước. Nếu nó là cực tiểu thì tăng biến đếm lên 1 (đếm khởi tạo ban đầu có giá trị bằng 0)

Lưu lại chỉ số i, j và A[i,j] trong 3 mảng 1 chiều là *hang*, *cot*, *giatri*

M1C2C. BÀI 52. Chữ số thứ N

Khi viết các số tự nhiên tăng dần từ 1, 2, 3... liên tiếp nhau, ta nhận được một dãy các chữ số thập phân vô hạn, ví dụ: 1234567891011121314151617181920...

Yêu cầu: Hãy tìm chữ số thứ N của dãy số vô hạn trên.

Dữ liệu vào: từ file NUMBER.INP gồm một số dòng, mỗi dòng ghi một số nguyên dương N ($N < 10^9$)

Kết quả ra: ghi ra file NUMBER.OUT, với mỗi số N đọc được từ file number.inp, ghi trên dòng tương ứng chữ số thứ N của dãy.

Ví dụ:

NUMBER.INP	NUMBER.OUT
5	5
10	1
54	3

M1C2C. BÀI 53. Lá bài

Có N lá bài với các số hiệu là 1, 2, ..., N, $N \leq 50$, trên lá bài i ghi một số nguyên dương $F[i]$, $1 \leq F[i] \leq N$.

Yêu cầu: Hãy tìm một số nhiều nhất các lá bài sao cho tập các số hiệu của chúng trùng với tập các số ghi trên các lá bài đó.

Dữ liệu vào: cho bởi file LABAI.INP trong đó dòng thứ nhất ghi số N nguyên dương. Trong các dòng tiếp theo mỗi dòng ghi 10 số (cho tới khi N hết số) lần lượt là các số ghi trên các lá bài 1 đến lá bài N.

Dữ liệu ra: ghi ra file LABAI.OUT, trong đó dòng thứ nhất ghi số lượng bài. Trong các dòng tiếp theo, mỗi dòng ghi 10 số là số hiệu của các lá bài được chọn, các số hiệu ghi theo thứ tự tăng dần cho đến hết.

Ví dụ:

LABAI.INP	LABAI.OUT
14	10

6 5 1 3 4 2 8 10 7 9 3 2 1 9	1 2 3 4 5 6 7 8 9 10
---------------------------------	----------------------

Thực chất bài toán có thể phát biểu lại như sau:

Cho 1 tệp (labai.inp) có dòng đầu chứa số nguyên N. Các dòng tiếp theo 1 dòng bao gồm 10 số nguyên i (cho tới khi hết số) có giá trị nằm trong đoạn 1 đến N.

Hãy tìm các số nguyên khác nhau từng đôi một (không trùng nhau) trong tệp đó rồi sắp xếp chúng theo thứ tự tăng dần và ghi ra tệp khác có tên là labai.out

MIC2C. BÀI 54. Trộn hai tệp

Cho hai tệp văn bản data1.inp và data2.inp chứa các số nguyên được sắp tăng. Viết chương trình trộn hai dãy dữ liệu trong hai tệp này thành một dãy dữ liệu sắp tăng duy nhất và ghi trong tệp văn bản data.out.

Chú ý:

- Với dữ liệu đã cho trong tệp thứ nhất là 5 số, tệp thứ hai là 6 số thì tệp kết quả sẽ chứa 11 số.
- Số lượng các số trong mỗi tệp tối đa là 100 và *không biết trước*.
- Các số có giá trị kiểu nguyên, được tách nhau bởi dấu cách và có thể nằm trên nhiều dòng.

Thí dụ:

data1.inp	data2.inp	data.out
	3	2
2	3	3
3	4	3
5	7	3
5	12	4
10	20	5
		5
		7
		10
		12
		20

Ý tưởng: Ta có thể sử dụng cách sau để xử lý bài toán:

- + Đọc tất cả dữ liệu trong hai tệp vào hai biến mảng.
- + Sử dụng thuật toán trộn hai mảng để có mảng kết quả.
- + Ghi mảng kết quả vào tệp data.out

Tuy nhiên cách giải này không hay bằng cách dưới đây:

Ta dùng phương pháp cân. Gọi hai tệp chứa dữ liệu cần trộn là f và g, tệp chứa kết quả trộn là h. Hãy tưởng tượng, ta dùng tay trái lấy lần lượt, mỗi lần một phần tử của tệp f (ghi vào biến t) và dùng tay phải lấy lần lượt mỗi lần một phần tử của tệp g (ghi vào biến p). So sánh vật nặng trên hai tay t và p. Tay nào cầm phần tử nhẹ hơn thì đặt phần tử đó vào tệp kết quả h và do tay đó rồi nên lấy tiếp phần tử từ tệp tương ứng. Quá trình này kết thúc khi nào một trong hai tệp f hoặc g được duyệt xong. Cuối cùng ta chuyển nốt các phần tử còn lại của tệp chưa duyệt hết (tệp f hoặc g) vào tệp kết quả h.

Ta cần lưu ý mấy điểm sau đây:

Khi đọc xong phần tử cuối cùng của một tệp thì tệp đó chuyển sang trạng thái kết thúc (EOF), do đó nếu ta tổ chức vòng lặp WHILE trong thủ tục trộn hai tệp theo điều kiện (NOT EOF(f)) AND (NOT EOF(g)) thì phần tử cuối của các tệp đó sẽ chưa kịp được so sánh, trong khi ta muốn tôn trọng nguyên tắc: sau khi so sánh t và p thì một trong hai biến, t hoặc p phải được giải phóng. Có thể thực hiện nguyên tắc này bằng kỹ thuật săn đuôi như sau: dùng biến logic ef ghi nhận trạng thái hết tệp f sớm hơn một nhịp. Điều đó có nghĩa khi ef=FALSE biến t vẫn đang chứa giá trị chưa xử lý (chưa so sánh với p và do đó chưa được ghi vào tệp h). Chú ý rằng dù ef = FALSE nhưng có thể ta vẫn có EOF(f)=TRUE. Một biến eg tương tự cũng được tạo cho tệp g. Về bản chất có thể hiểu các biến ef và eg khi chúng nhận giá trị TRUE là thực sự đã đọc được 1 đơn vị dữ liệu từ file.

Thủ tục Seekeof(f) dùng để kiểm tra con trỏ tệp đến vị trí trước vị trí kết thúc tệp hay chưa.

MIC2C. BÀI 55. Trộn hai tệp thành tệp tăng dần

Hai file SN1.TXT, SN2.TXT chứa các số nguyên bất kỳ. Hãy viết chương trình tạo file KQ.OUT chứa các số nguyên được sắp xếp theo thứ tự tăng dần, mỗi số trong file KQ.OUT là tổng của hai số nguyên lấy từ file SN1.TXT và SN2.TXT.

Ví dụ: File SN1.TXT chứa các số nguyên: 1 9 7 12; SN2.TXT chứa các số nguyên: 31 5 2 1 thì file KQ.OUT có kết quả: 9 13 14 32

Ý tưởng: đọc dữ liệu từ 2 file vào mảng, sắp xếp tăng dần, duyệt qua hai mảng và cộng các phần tử tương ứng rồi ghi vào tệp.

MIC2C. BÀI 56. Tìm số lần lặp lại nhiều nhất của một số trong dãy số

Cho trước dãy n số nguyên a_1, a_2, \dots, a_n không giảm và hai chỉ số i và j ($1 \leq i \leq j \leq n$).

Hãy tìm số lần lặp lại nhiều nhất của các số trong dãy số a_i, a_2, \dots, a_j

Dữ liệu vào từ file văn bản DAYSO.INP gồm:

- Dòng đầu tiên chứa số nguyên n
- Dòng thứ hai chứa dãy số a_1, a_2, \dots, a_n .
- Dòng thứ ba chứa 2 số i, j .

(Các số trên cùng một dòng cách nhau ít nhất một dấu cách)

Kết quả ghi ra file DAYSO.OUT có

- Dòng đầu tiên chứa số lần lặp lại nhiều nhất của các số trong dãy a_1, a_2, \dots, a_n .
- Dòng thứ hai chứa số lần lặp lại nhiều nhất của các số trong dãy a_i, \dots, a_j .

Ví dụ:

DAYSO.INP	DAYSO.OUT
10	4
-1 -1 1 1 1 3 10 10 10	3
5 10	

Hạn chế: $1 \leq n \leq 100; -10000 \leq a_i \leq 10000$

Bài này là dạng bài cơ bản, giáo viên hướng dẫn hs đếm từng loại giá trị để đưa ra giá trị xuất hiện nhiều nhất.

MIC2C. BÀI 57. Xếp khách

Một khách sạn có N phòng đôi (phòng cho 2 người) được đánh số từ 1 đến N . Khi có một đoàn khách đến thuê phòng, tiếp tân của khách sạn sẽ xếp khách vào phòng theo quy tắc sau: Mỗi cặp khách sẽ được xếp vào phòng có chỉ số nhỏ nhất trong số các phòng trống. Nếu số lượng khách của đoàn là số lẻ thì người cuối cùng của đoàn khách sẽ được xếp vào phòng có chỉ số nhỏ nhất trong số các phòng trống. Nếu không còn phòng trống, thì số khách chưa có phòng sẽ được xếp tuần tự từng người một vào phòng có chỉ số nhỏ nhất trong số các phòng mới có một khách ở (của đoàn khách đến trước). Đầu tiên tất cả các phòng của khách sạn là trống.

Yêu cầu: Cho trước trình tự đến của các đoàn khách và số lượng khách của mỗi đoàn. Hãy xác định số lượng khách trong mỗi phòng của khách sạn.

Dữ liệu vào: từ file văn bản ROOM.INP:

- Dòng đầu tiên chứa 2 số nguyên dương N ($1 \leq N \leq 100$) và G được ghi cách nhau bởi dấu cách. N là số phòng của khách sạn, G là số lượng đoàn khách.
- Dòng thứ i trong số G dòng tiếp theo chứa số lượng khách của đoàn khách thứ i (các đoàn khách được đánh số theo thứ tự đến khách sạn bắt đầu từ 1).

Giả thiết rằng không có 2 đoàn khách nào đến khách sạn vào cùng một thời điểm và tổng số khách của tất cả các đoàn không vượt quá sức chứa của khách sạn ($\leq 2N$).

Kết quả: Ghi ra file văn bản ROOMS.OUT N dòng: Dòng thứ i chứa số lượng khách của phòng i ($1 \leq i \leq N$).

Ví dụ:

ROOMS.INP	ROOMS.OUT
7 3	2
3	1
1	1
4	2
	2
	0
	0

MIC2C. BÀI 58. Mã số sách

Để đánh mã số cho từng quyển sách, một cán bộ thư viện dự định dùng một từ có 5 kí hiệu liên tiếp nhau gồm: 2 kí tự (A.....Z) tiếp theo là 3 kí số (001...999). Em hãy lập trình tạo tất cả từ có thể dùng làm mã số.

Yêu cầu kĩ thuật

Kết quả: ghi ra tệp văn bản MASO.INP

Mỗi dòng có 8 từ, mỗi từ cách nhau một khoảng trống.

Ghi chú: Mã số sách không có 3 kí hiệu cuối là 000

Ý tưởng: Duyệt tất cả các khả năng và kiểm tra điều kiện đề bài để ghi kết quả.

MIC2C. BÀI 59. Tam giác số

Hình bên mô tả một tam giác số có số hàng $N=5$. Đi từ đỉnh (số 7) đến đáy tam giác bằng một đường gấp khúc, mỗi bước chỉ được đi từ số ở hàng trên xuống một trong hai số đứng kề bên phải hay bên trái ở hàng dưới, và tính tích các số trên đường đi lại ta được một tích.

Ví dụ: đường đi 7 8 1 4 có tích là $S=224$, đường đi 7 3 1 7 có tích là $S=147$.

Yêu cầu: Cho tam giác số, tìm tích của đường đi có tích lớn nhất.

Dữ liệu vào: cho trong tệp văn bản TGS.INP:

+ Dòng đầu tiên chứa số nguyên n , ($0 < N < 101$).

+ N dòng tiếp theo, từ dòng thứ 2 đến dòng thứ $N+1$: dòng thứ i có $(i-1)$ số cách nhau bởi dấu cách (các số có giá trị tuyệt đối ≤ 100).

Kết quả ra: ghi vào tệp văn bản TGS.OUT một số nguyên – là tích lớn nhất tìm được.

Chú ý:

- 30 %, số chữ số của $s \leq 18$.
- 30 %, các số trong tam giác nguyên dương, số chữ số của $s > 18$.
- 40 %, các số có thể ≤ 0 hoặc > 0 , số chữ số của $s > 18$.

Ý tưởng: Sử dụng QHĐ để tìm tích đường đi lớn nhất.

Gọi $M[i,j]$ là tích lớn nhất đến ô (i,j) .

Trường hợp đơn giản:

Nếu $i=1$ và $j=1$ thì $M[i,j] = a[i,j]$

(Với a là mảng giá trị ban đầu của tam giác).

Với $j=1$ hoặc $j=i$ tức ở bìa tam giác (chỉ có một đường duy nhất đi đến đó).

Khi đó:

$M[i,j] := a[i,j] * M[i-1,j]$ (nếu $j=1$)

$M[i,j] := a[i,j] * M[i-1,j-1]$ (nếu $j=i$)

Với $j \neq 1$ hoặc $j \neq i$, có thể đi đến ô (i,j) từ một trong hai ô $(i-1,j)$ hoặc $(i-1,j-1)$, do đó:

nếu $a[i,j] < 0$ thì $M[i,j] := a[i,j] * \max(\text{abs}(M[i-1,j]), \text{abs}(M[i-1,j-1]))$

ngược lại thì $M[i,j] := a[i,j] * \max(M[i-1,j], M[i-1,j-1])$;

TGS.INP

4

7

3 8

8 1 0

2 7 4 4

TGS.OUT

1176

Bài tập tương tự: tìm đường đi có tổng lớn nhất

MIC2C. BÀI 60. Thuê máy tính

Sau khi vào đội tuyển, Long bỗng học sút hần đi. Thám tử của khối điều tra và biết được Long mở một công ty Tin học với tên rất hấp dẫn: AttoSoft (Micro = 10^{-6} , Pico = 10^{-12} , Atto = 10^{-18}). Long chỉ có một máy tính.

Một khách hàng thuê Long viết N chương trình. Long thuê luôn N nhà lập trình mỗi người viết một chương trình.

N nhà lập trình đều có mặt lúc 0 giờ. Mỗi nhà lập trình cần một số nguyên giờ viết chương trình và đòi trả tiền theo số giờ từ lúc bắt đầu đến công ty cho tới khi chương trình đưa vào máy tính. Nói chung, mỗi nhà lập trình đòi thù lao khác nhau.

Hãy giúp Long thu xếp trình tự cho các nhà lập trình đưa chương trình vào máy tính sao cho tổng chi phí là ít nhất.

Dữ liệu vào được cho bởi file ATTO1.INP trong đó dòng thứ nhất ghi số nguyên dương N không lớn hơn 10000. Trong N dòng tiếp theo, dòng thứ U ghi hai số nguyên dương X, Y không lớn hơn 30000 với ý nghĩa nhà lập trình thứ U cần trả X đồng cho một giờ làm việc và cần Y giờ để viết chương trình.

Kết quả ghi ra file ATTO1.OUT N dòng, dòng thứ I ghi số hiệu nhà lập trình là người thứ I vào máy tính.

Ví dụ

ATTO.INP	ATTO.OUT
3	1
100 2	3
20 5	2
500 20	

Ý tưởng: Sắp xếp các nhà lập trình theo các tỷ số X/Y giảm dần và theo thứ tự đó lần lượt chọn các nhà lập trình.

MIC2C. BÀI 61. Hàng đợi

Có N người hâm mộ (vì không biết tên họ nên tạm đặt tên họ từ 1 đến N tính từ đầu hàng) đứng trước quầy bán vé để mua vé cho một kỳ EURO 2012. Để có thể mua một chiếc vé là không hề dễ dàng.

Họ phải xếp hàng từ tối hôm trước đến sáng sớm hôm sau, và theo tự nhiên một vài người trong số họ có nhu cầu sử dụng nhà vệ sinh công cộng. Mỗi khi có nhu cầu, người đó sẽ bước ra khỏi hàng đợi, và sau khi

hoàn thành nhiệm vụ, bước trở lại hàng, mặc dù không nhất thiết phải là vị trí trước đó.

Vì chỉ có 1 nhà vệ sinh, nên không ai bước ra khỏi hàng đợi trước khi người trước đó trở lại hàng (như vậy tại bất kì thời điểm nào thì trong hàng chỉ có nhiều nhất 1 người vắng mặt).

Suốt đêm hôm trước, có tổng cộng K cuộc viếng thăm nhà vệ sinh. Mỗi cuộc viếng thăm được mô tả bởi hai số nguyên A và B, biểu thị rằng người có tên A bước ra khỏi hàng đợi và trở lại hàng đợi ngay trước mặt người có tên B. Bây giờ tất cả các cuộc viếng thăm đã hoàn thành, thứ tự của N người đã bị đảo lộn trong hàng.

Yêu cầu: cho biết trước các cuộc viếng thăm nhà vệ sinh, sau khi kết thúc k cuộc viếng thăm, hãy cho biết người đứng trước và đứng sau của mỗi người trong hàng.

Dữ liệu vào cho trong tệp văn bản QUEUE.INP:

- + Dòng đầu tiên chứa số nguyên N ($2 \leq N \leq 105$) – số lượng người trong hàng
- + Dòng thứ 2 chứa số nguyên K ($1 \leq K \leq N$) – tổng số các cuộc viếng thăm nhà vệ sinh.
- + K dòng tiếp theo mỗi dòng chứa hai số nguyên khác nhau A và B ($1 \leq A, B \leq 105$), mô tả một cuộc viếng thăm nhà vệ sinh.

Kết quả ghi ra tệp văn bản QUEUE.OUT: ghi ra N dòng:

Dòng thứ i : ghi hai số X Y thể hiện người đứng trước và sau người tên là i. Nếu người i là người đầu hàng thì người đứng trước người i là 0, người thứ i là người cuối hàng thì người đứng sau người i là 0.

Ví dụ:

QUEUE.INP	QUEUE.OUT
9	2 9
5	0 1
6 3	7 8
9 6	5 7
3 8	6 4
4 7	9 5
2 1	4 3
	3 0
	1 6

* Ghi chú: Có 60% test $N \leq 10^3$

Ý tưởng: Dùng hai mảng T và S để lưu vị trí trước và sau của mỗi phần tử, ban đầu ta có mảng như sau:

Mảng T	1	2	3	4	..		N
	0	1	2	3	N-1

Mảng S	1	2	3	4		N-1	N
	2	3	4	5	...	N	0

Ta xác định vị trí của A và B ban đầu như sau:

Vị trí ban đầu	T[A]	A	S[A]	...		T[B]	B	S[B]
----------------	------	---	------	-----	--	------	---	------

Vị trí sau khi thay đổi	T[A]	S[A]	...	T[B]	A	B	S[B]
-------------------------	------	------	-----	------	---	---	------

Sau khi A thay đổi vị trí thì ta cập nhật lại vị trí mới như sau:

- + Vị trí trước của phần tử sau A(ban đầu) bằng vị trí trước của A: $T[S[A]] = T[A]$
- + Vị trí sau của phần tử trước A(ban đầu) bằng vị trí sau A: $S[T[A]] = S[A]$
- + Vị trí trước A (mới) bằng vị trí trước của B: $T[A] = T[B]$
- + Vị trí sau của A là B: $S[A] = B$
- + Vị trí sau của phần tử trước B (ban đầu) là A(mới): $S[T[B]] = A$
- + Vị trí trước B là A: $T[B] = A$

M1C2C. BÀI 62. Bản tin bóng đá

Sau cuối mùa giải bóng đá, căn cứ vào bảng điểm của tất cả các trận đấu, Ban tổ chức biết được số trận thắng, thua, hòa và tổng số điểm của mỗi đội. Từ đó, Ban tổ chức biết được đội bóng đá mạnh nhất trong mùa giải.

Quy ước: Trận thắng được 3 điểm, trận hòa được 1 điểm, thua là 0 điểm.

Yêu cầu: Người lập trình giúp Ban tổ chức, thống kê số trận thắng, hòa, thua, tổng điểm của mỗi đội, và tìm được đội bóng đá mạnh nhất.

Dữ liệu vào: cho trong tệp văn bản BANGDIEM.INP

- Dòng đầu tiên ghi hai số nguyên dương n, m tương ứng là n đội và m trận đấu của mỗi đội, giữa 02 số cách nhau bởi dấu cách.

- Dòng thứ i trong n dòng tiếp theo chứa m số nguyên thuộc một trong các số 0, 1, 3 ứng với điểm mỗi trận đấu, và cách nhau bởi dấu cách.

Kết quả: Ghi ra tệp văn bản TKEGIAI.OUT: Ghi n dòng, mỗi dòng là bốn chữ số, theo thứ tự là số lượng trận thắng, hòa, thua và số cuối là tổng điểm của mỗi đội. Dòng tiếp theo, ghi số thứ tự của đội bóng trong danh sách ban đầu là đội bóng mạnh nhất. Nếu trong trường hợp có nhiều đội bóng có cùng tổng số điểm thì ta liệt kê hết các số thứ tự của đội bóng đó và cách nhau bởi dấu cách.

Ví dụ:

BANGDIEM.INP	TKEGIAI.OUT
3 4	1 1 2 4
1 3 0 0	3 1 0 10
3 3 3 1	0 2 2 2
1 1 0 0	2

MIC2C. BÀI 63. Nhà chung cư

Một khu chung cư được xây dựng nhằm đáp ứng nhu cầu nhà ở đối với những người có thu nhập thấp. Do nhu cầu nhà ở rất lớn và nhu cầu cũng rất khác nhau, nên người kinh doanh nhà ở đã xây dựng với số lượng căn hộ rất lớn và giá trị cho thuê của mỗi căn hộ cũng khác nhau. Mỗi căn hộ được trang bị khá đầy đủ tiện nghi, đảm bảo cho một gia đình sinh hoạt hàng ngày.

Yêu cầu: Người lập trình hãy giúp cho người kinh doanh thống kê có bao nhiêu loại căn hộ ứng với các mức giá trị cho thuê. Đồng thời cho biết số lượng căn hộ có giá trị cho thuê bao nhiêu là nhiều nhất?

Dữ liệu vào: cho trong tệp văn bản GIACANHO.INP

- Dòng đầu tiên ghi số nguyên dương N là số lượng căn hộ của khu chung cư.
- Dòng thứ hai trở đi, mỗi dòng chứa 10 số (cho đến khi hết N số), mỗi số cách nhau một dấu cách, dòng cuối cùng có thể ít hơn 10 số, số nhỏ nhất là 100 (đơn vị tính là triệu đồng), số lớn nhất là 800.

Kết quả: Ghi ra tệp văn bản LOAIGIA.OUT: dòng đầu ghi M là số lượng loại căn hộ khác nhau ứng với mỗi giá trị cho thuê; dòng thứ hai ghi 2 chữ số cách nhau bởi dấu cách, số đầu ghi số lượng căn hộ có mức giá cho thuê nhiều nhất, số thứ hai là giá trị căn hộ đó.

Ví dụ:

GIACANHO.INP	LOAIGIA.OUT
15	10
100 150 150 200 250 300 250 150 400 150	5 150
500 150 600 700 800	

MIC2C. BÀI 64. Hành tinh XYZ

Hành tinh XYZ là một hành tinh có nền kinh tế, khoa học rất phát triển. Hành tinh có n người sinh sống, mỗi người trên hành tinh có 1 mã số là số nguyên dương. Không có 2 mã số nào giống nhau. Độ phù hợp giữa 2 người được tính như sau: biểu diễn mã dưới dạng nhị phân, bổ sung các số 0 vào đầu nếu cần thiết để 2 số có cùng độ dài, viết 1 số dưới số kia và tạo ra số nhị phân mới theo nguyên tắc: nếu 2 bit của các toán hạng giống nhau thì bit kết quả là 0, trong trường hợp ngược lại bit kết quả là 1, sau đó kết quả được đổi trở lại hệ 10.

Ví dụ 2 người có mã là 19 và 10 sẽ có độ phù hợp là 25

$$10011 = 19$$

$$01010 = 10$$

$$\overline{11001} = 25$$

Độ phù hợp của hành tinh là tổng độ phù hợp của tất cả các cặp 2 người.

Hãy tính độ phù hợp của hành tinh.

Dữ liệu vào: cho trong văn bản XYZ.INP gồm :

Dòng chứa số nguyên n ($2 \leq n \leq 10^6$)

Mỗi dòng trong n dòng sau chứa một mã số, mã có giá trị không vượt quá 10^6

Kết quả: ghi ra tệp văn bản XYZ.OUT một số nguyên là độ phù hợp của hành tinh.

Ví dụ

XYZ.INP	XYZ.OUT
3	12
7	
3	
5	

Chú ý: 50% số test có $n \leq 1000$ ứng với 50% số điểm của bài.

MIC2C. BÀI 65. Bảng số

Cho một bảng số gồm n số nguyên, mỗi số được viết trên một ô. Hãy cắt bảng số này thành nhiều đoạn nhất sao cho tổng các phần tử trong các đoạn là bằng nhau.

Dữ liệu vào: DIV.INP + Dòng đầu ghi n ($n \leq 1000$)

+ Dòng tiếp theo ghi n số nguyên a_1, a_2, \dots, a_n

(các số nằm trên một dòng cách nhau bởi một dấu cách $a_i \leq 1000$)

Dữ liệu ra: DIV.OUT Ghi K là số đoạn cần chia.

Ví dụ:

10	-1	12	-1	5	2	1	2
----	----	----	----	---	---	---	---

10	-1	12	-1	5	2	1	2
----	----	----	----	---	---	---	---

DIV.INP	DIV.OUT	Giải thích
8 10 -1 12 -1 5 2 1 2	3	Đoạn 1: 10 Đoạn 2: -1 + 12 + -1 = 10 Đoạn 3: 5 + 2 + 1 + 2 = 10

MIC2C. BÀI 66. Tìm chữ số thứ M

Xét dãy số tự nhiên $\{a_n\}$ được xây dựng theo quy tắc sau:

- Cho trước số a_0 là một số tự nhiên có tối đa 10 chữ số.
- Số a_i ($i > 0$) là một số tự nhiên nhận được từ a_{i-1} bằng cách viết thêm vào sau các chữ số của a_{i-1} chính a_{i-1} nhưng viết theo thứ tự ngược lại.

Ví dụ:

Với $a_0 = 123$ thì $a_1 = 123321$, $a_2 = 123321123321$, $a_3 = 123321123321123321$

Với hai số N và M cho trước, hãy tìm chữ số thứ M trong a_N .

Dữ liệu cho trong file văn bản với tên là CHUSO.INP trong đó dòng đầu chứa số a_0 , dòng thứ hai chứa hai số N và M .

Kết quả ghi ra file văn bản với tên là CHUSO.OUT. Trong trường hợp có lời giải, file này sẽ chứa số tìm được, ngược lại file này chứa số -1.

Ví dụ:

CHUSO.INP
123
3 7

CHUSO.OUT
1

Giới hạn: $1 \leq N \leq 25$, $1 \leq M \leq 1\ 000$

000 000.

Ý tưởng:

Trước tiên ta nhận xét mặc dù đề bài cho a_0 là số tự nhiên, nhưng vì bài toán không sử dụng tính chất của số nên ta có thể xem a_0 như một xâu kí tự.

Gọi L là số kí tự của a_0 , ta thấy a_0 có L kí tự, a_1 có $2L$ kí tự, a_2 có $4L$ kí tự, ..., a_N có $2^N L$ kí tự.

Kí hiệu s^R là chuỗi kí tự đảo ngược của chuỗi s . Ví dụ: nếu $s = "abca"$ thì $s^R = "acba"$, các chữ cái được viết theo thứ tự ngược lại. Để ý là với hai chuỗi a, b bất kỳ ta có $(ab)^R = b^R a^R$. Theo đề bài, ta có $a_1 = a_0 a_0^R$, $a_2 = a_1 a_1^R = a_0 a_0^R (a_0 a_0^R)^R = a_0 a_0^R a_0 a_0^R$, ..., $a_N = a_0 a_0^R a_0 a_0^R \dots a_0 a_0^R$, nghĩa là xâu a_N sẽ ghép thành từ các xâu a_0 và a_0^R xen kẽ nhau.

Từ nhận xét này ta có thuật toán sau:

Nếu vị trí M nằm ngoài xâu kí tự a_N , hay nói cách khác nếu $M < 1$ hoặc $M > 2^N L$ thì in ra -1. Đoạn lệnh sau đây thể hiện điều này:

```
if (m < 1) or (m > 1 * (1 shl n)) then {chú ý: 1 shl n = 2^n}
begin
    timchuso := -1;
    exit;
end;
```

Còn nếu không, ta xem vị trí M sẽ thuộc về một xâu a_0 hay một xâu đảo ngược a_0^R , điều này được chỉ ra bởi giá trị của biểu thức $((M-1) \text{ div } L) \bmod 2$ bằng 0 hay 1. Biết được điều này, sử dụng phép lấy số dư, ta sẽ tìm được vị trí của kí tự cần tìm trong chuỗi a_0 :

$i := (m-1) \text{ mod } l+1$; { i là vị trí tương ứng với vị trí M trong chuỗi a_0 }

if $((m-1) \text{ div } l) \bmod 2 = 1$ then $i := l-i+1$;

{nếu vị trí M thuộc xâu đảo ngược a_0^R thì ta đảo ngược vị trí i }

timchuso:=ord(a0[i])-ord('0'); {kết quả bằng chữ số a0[i]}

M1C2C. BÀI 67. Diện tích miền phủ

(câu 4, HSG Thanh Hóa, 2011-2012)

Một trường học X cần làm mái che nằm ngang. Trường học có N tấm bạt hình chữ nhật, các tấm bạt làm mái che có thể chồng lên nhau. Tính diện tích bị che phủ.

Dữ liệu vào: Từ file MIENPHU.INP gồm:

- Dòng 1: Số nguyên N ($1 \leq N \leq 10$).

- Dòng 2...N+1: Mỗi dòng chứa bốn số nguyên là x1, y1, x2 và y2, trong đó (x1, y1) là điểm trái trên của tấm bạt hình chữ nhật còn (x2, y2) là điểm phải dưới của tấm bạt hình chữ nhật ($-104 \leq x1, y1, x2, y2 \leq 104$). Các số trên một dòng cách nhau ít nhất một dấu cách trống.

Kết quả: Ghi ra file MIENPHU.OUT gồm: Một dòng duy nhất là kết quả của bài toán.

Ví dụ:

MIENPHU.INP	MIENPHU.OUT
3	20
0 5 4 1	
2 4 6 2	
0 5 4 1	

Ý tưởng: Vì dữ liệu bài toán nhỏ nên ta có thể sử dụng một mảng hai chiều A[-104..104,-104..104] để thể hiện cho diện tích cần làm mái che.

Ban đầu ta gán toàn bộ các phần tử của mảng A giá trị 0 (chưa che các tấm bạt)

Đọc tọa độ các tấm bạt, với mỗi tấm ta sẽ gán giá trị 1 cho phần mà tấm bạt này phủ

Readln(f,x1,y1,x2,y2);

For i:= x1 to x2 do

For j:= y1 to y2 do

A[i,j]:=1;

Sau cùng ta duyệt toàn bộ mảng A, đếm số lượng phần tử có giá trị là 1. Đó chính là kết quả bài toán.

(Cách này chưa tối ưu)

Cách tối ưu là áp dụng thuật toán tính diện tích miền phủ bởi N hình chữ nhật.

M1C2C. BÀI 68: Tính diện tích

Trên mặt phẳng tọa độ cho N ($N \leq 10\,000$) hình chữ nhật với các cạnh song song với các trục tọa độ. Các hình chữ nhật được đánh số từ 1 tới N. Hình chữ nhật thứ i được cho bởi tọa độ đỉnh trái dưới (x_{i1}, y_{i1}) và tọa độ đỉnh phải trên (x_{i2}, y_{i2}). Các số $x_{i1}, y_{i1}, x_{i2}, y_{i2}$ là các số nguyên trong phạm vi từ -100 đến 100.

Hãy lập trình tính:

1. Diện tích của phần mặt phẳng mà N hình chữ nhật này phủ.
2. Tính diện tích phần chung của N hình chữ nhật này.

Dữ liệu cho trong file HCN.INP trong đó dòng đầu chứa số N. Dòng thứ i trong N dòng tiếp theo chứa 4 số số

$x_{i1}, y_{i1}, x_{i2}, y_{i2}$.

Kết quả ghi ra file HCN.OUT gồm 2 dòng, trong đó dòng đầu chứa số S₁ là kết quả của câu 1. Dòng thứ hai chứa số S₂ là kết quả của câu 2.

Ví dụ:

HCN.INP	HCN.OUT
2	4
0 0 1 1	1
-1 -1 1 1	

Hướng dẫn:

Ý tưởng:

- Lập mảng X[1..2n], Y[1..2n] lần lượt chứa hoành độ, tung độ các hình chữ nhật

- Lưu tọa độ ban đầu các hình chữ nhật vào mảng A

- Sắp xếp mảng X,Y tăng dần

- Lần lượt kiểm tra các hình chữ nhật có tọa độ đỉnh trên bên phải (x_{i+1}, y_{i+1}) và tọa độ đỉnh dưới bên

phải là (x_i, y_i) với $1 \leq i \leq n-1$. Nếu hình chữ nhật này thuộc một trong các hình chữ nhật ban đầu thì cộng thêm vào phần diện tích đang cần tìm diện tích của hình chữ nhật con này.

Const Maxn=1000;

Type Toa_do = record x1,y1,x2,y2:integer; end;

mang=array[1..2*maxn] of integer;

var N,s:integer;

a:array[1..maxn] of toa_do;

X,Y:mang;

{=====Doc du lieu=====}

```

Procedure doc_dl;
var f:text; i:integer;
begin
  assign(f,'HCN.inp'); reset(f);
  readln(f,n);
  for i:=1 to n do
    readln(f,a[i].x1,a[i].y1,a[i].x2,a[i].y2);
  close(f);
end;
{===Sap xep mang tang dan===}
Procedure Sapxep(Var T:Mang);
Var i,j,TG,m:integer;
Begin
  m:=2*n;
  for i:=1 to m-1 do
    for j:=i+1 to m do
      if T[i]>T[j] then
        begin
          TG:=T[i];
          T[i]:=T[j];
          T[j]:=TG;
        end;
    end;
end;
{====Kiem tra HCN moi thuoc HCN da cho====}
function Kiemtra_thuoc(i,j:integer):boolean;
var k:integer;
begin
  Kiemtra_thuoc:=true;
  for k:=1 to N do
    if (a[k].x1<=x[i-1]) and (X[i]<=a[k].x2)
    and (a[k].y1<=y[j-1]) and (Y[j]<=a[k].y2) then exit;
  Kiemtra_thuoc:=false;
end;

Procedure Xuly;
var i,j:integer;
Begin
  for i:=1 to N do
    begin
      X[i*2-1]:=a[i].x1;
      X[i*2]:=a[i].x2;
      Y[i*2-1]:=a[i].y1;
      Y[i*2]:=a[i].y2;
    end;
  Sapxep(X); Sapxep(Y);
  for i:=2 to 2*N do
    for j:=2 to 2*N do
      if Kiemtra_thuoc(i,j) then S:=s+(X[i]-X[i-1])*(Y[j]-Y[j-1]);
    end;
end;

Begin
  doc_dl;
  s:=0;
  Xuly;
  Write('Dien tích là: ', S);
  readln;
End.

```

Câu b) tương tự bài tìm điểm thuộc tất cả N hình chữ nhật (đề thi HSG tỉnh Cà Mau 2013-2014)

M1C2C. BÀI 69. Tạo bảng

Cho một bảng A gồm $N \times N$ số nguyên ($N \leq 100$), các dòng được đánh số từ trên xuống dưới bắt đầu từ 1, các cột được đánh số từ trái qua phải cũng bắt đầu từ 1. Mỗi số trong bảng có giá trị tuyệt đối không vượt quá 30000. Bảng B được tạo ra từ bảng A theo qui tắc sau:

Phần tử của B nằm ở dòng I, cột j có giá trị bằng tổng của các số nằm trong ô (i,j) và các ô kề nó trong bảng A: $B_{ij} = A_{ij} + A_{(i+1)j} + A_{(i-1)j} + A_{i(j+1)} + A_{i(j-1)}$

Chú ý: Các phần tử nằm ngoài bảng được xem như có giá trị bằng 0.

Bài toán: Cho bảng A. Hãy tạo ra bảng B tương ứng.

Dữ liệu vào cho trong file văn bản TABLE.INP trong đó :

- Dòng đầu chứa số N.
- Dòng thứ i trong N dòng tiếp theo chứa N số nguyên lần lượt ứng với các phần tử nằm trên dòng thứ i của bảng A.
- Các số trên cùng một dòng cách nhau bởi khoảng trắng.

Kết quả ghi ra file văn bản TABLE.OUT cho biết bảng B tạo được có định dạng cùng một qui cách với file input, nghĩa là:

Dòng đầu chữ số N.

Dòng thứ i trong N dòng tiếp theo chứa N số nguyên lần lượt ứng với các phần tử nằm trên dòng thứ i của bảng B.

Các số trên cùng một dòng cách nhau bởi khoảng trắng.

Ví dụ:

TABLE.INP
4
1 2 3 4
5 6 7 8
9 8 7 6
5 4 3 2

TABLE.OUT
4
8 12 16 15
21 28 31 25
27 34 31 23
18 20 16 11

Ý tưởng

Ta in ra các phần tử của mảng B theo công thức như đề bài nêu:

```
for i:=1 to n do
begin
  for j:=1 to n do
    write(a[i,j]+a[i,j-1]+a[i-1,j]+a[i,j+1]+a[i+1,j], ' ');
  writeln;
end;
```

Khi cài đặt chú ý để khai báo dư ra các phần tử biên trên mảng A và đặt chúng bằng giá trị 0:

```
var
  a: array[0..MAXN+1, 0..MAXN+1] of longint; {0,n+1 là chỉ số của các hàng, cột biên}
...
fillchar(a,sizeof(a),0); {tất cả các phần tử, bao gồm phần tử biên sẽ bằng 0}
```

MIC2C. BÀI 70. Tìm số âm lớn nhất

(câu 2, đề thi Nghệ An 2012-2013, bảng B)

Cho một dãy gồm N số nguyên a_1, a_2, \dots, a_N , mỗi số có giá trị tuyệt đối không vượt quá 10^5 .

Yêu cầu: Hãy tìm số âm lớn nhất X trong dãy.

Dữ liệu vào: cho trong tệp văn bản SOAM.INP:

- Dòng đầu tiên chứa số nguyên dương N ($1 \leq N \leq 10^5$).
- N dòng tiếp theo, dòng thứ i chứa số a_i .

Kết quả: Ghi ra tệp văn bản SOAM.OUT trên một dòng duy nhất số X tìm được. Trong trường hợp không có lời giải, ghi ra số 0.

Ví dụ:

SOAM.INP	SOAM.OUT
5	-4
-4	
3	
2	
-5	
7	

MIC2C. BÀI 71. Trạm canh

(câu 2, HSG An Giang 2009)

Tại một vùng đất nọ, người ta bố trí các trạm canh để cảnh báo nguy cơ cho toàn vùng. Mỗi trạm canh gồm có một đài lửa và một cung thủ được trang bị một số lượng tên nhất định. Khi có nguy cơ xuất hiện, một trạm được chỉ định làm trạm xuất phát sẽ tự đốt lửa báo hiệu, sau đó người cung thủ của trạm sẽ bắn tên để thấp lửa cho các trạm khác theo những chỉ thị cho sẵn. Đến lượt mình, mỗi trạm sau khi được thấp lửa, người cung thủ của trạm lại

bắn tên để thả lửa cho các trạm khác, cũng theo những chỉ thị cho sẵn. Việc bắn tên của mỗi cung thủ chỉ kết thúc khi đã sử dụng hết số tên được cấp hoặc khi tất cả các trạm đã được thả lửa báo hiệu.

Xem vùng đất như một mặt phẳng tọa độ và mỗi trạm canh được đặt tại một tọa độ nhất định. Viết chương trình xác định thời điểm mà mỗi trạm được thả lửa báo hiệu.

Giả sử thời gian để một cung thủ bắn tên thả lửa cho trạm kế tiếp bằng khoảng cách giữa hai trạm (thời gian giữa các loạt bắn là không đáng kể) và trạm số 1 được chỉ định là trạm xuất phát. Ngoài ra, ta cũng giả sử mỗi cung thủ luôn bắn trúng đích.

Dữ liệu vào: chứa trong tệp văn bản SPARK.INP có dạng như sau:

- + Dòng đầu chứa số nguyên dương N ($1 \leq N \leq 100$) cho biết số trạm canh.
- + N dòng tiếp theo mỗi dòng chứa $N+2$ số nguyên theo thứ tự cho biết các thông tin sau:

Hai số nguyên X, Y ($1 \leq X, Y \leq 1000$) cho biết tọa độ của trạm.

Số nguyên S ($1 \leq S \leq 100$) cho biết lượng cung tên được cấp cho trạm.

$N-1$ số nguyên cho biết thứ tự các trạm được chỉ định cho người cung thủ phải bắn tên để thả lửa tiếp theo.

Dữ liệu ra: ghi vào tệp văn bản SPARK.OUT gồm N dòng, mỗi dòng ghi thời điểm mà trạm tương ứng được thả lửa báo hiệu. Các thời điểm ghi dưới dạng số thập phân với độ chính xác đến ± 0.001

Ví dụ:

SPARK.INP	SPARK.OUT
4	0.000000
1 1 1 2 3 4	1.000000
1 2 1 4 1 3	3.000000
2 1 1 2 1 4	2.000000
2 2 1 3 2 1	

M1C2C. BÀI 72. Hình chữ nhật

(câu 1, Hải Dương, lớp 12 thpt năm học 2012 – 2013)

Trong dịp nghỉ hè, bé Mai được bố mẹ cho đi tắm biển. Trên bờ biển bé nhặt được N viên đá cuội rất đẹp mắt. Mai quyết định vẽ trên cát một lưới hình chữ nhật kích thước $a \times b$ (a, b nguyên dương) được chia thành $a \times b$ ô vuông bằng các đường ngang dọc sao cho có thể rải N hòn sỏi này vào các ô vuông sao cho mỗi ô vuông có nhiều nhất một viên sỏi.

Hãy giúp bé Mai chọn kích thước của hình chữ nhật sao cho chu vi của nó là nhỏ nhất. In ra màn hình giá trị chu vi này.

Dữ liệu: Nhập vào từ bàn phím số nguyên dương N ($N \leq 10^9$).

Kết quả: Ghi ra màn hình chu vi của hình chữ nhật tìm được

Ví dụ:

Dữ liệu nhập vào: 15

Kết quả in ra: 16

Giải thích: Hình chữ nhật tìm được có kích thước 4×4

M1C2C. BÀI 73. Dây con chung dài nhất¹

(Câu 3, Quảng Bình 2012-2013)

Cho dãy số nguyên A gồm N phần tử a_1, a_2, \dots, a_N và dãy số nguyên B gồm M phần tử b_1, b_2, \dots, b_M . Các phần tử trong một dãy số có giá trị khác nhau từng đôi một. ($1 \leq a_i, b_j \leq 2 \times 10^9$; $1 \leq N \leq 100$; $1 \leq i \leq N$; $1 \leq M \leq 100$; $1 \leq j \leq M$). Dây C được gọi là dây con của dãy A nếu dây C nhận được từ dãy A bằng cách xóa đi một số phần tử và giữ nguyên thứ tự của các phần tử còn lại. Nếu dây C là dây con của dãy A và cũng là dây con của dãy B thì dây C được gọi là dây con chung của hai dãy A và B .

Yêu cầu: Hãy tìm dây C là dây con chung của hai dãy A và B sao cho số lượng phần tử của dây C là lớn nhất.

Dữ liệu vào: Cho trong file văn bản DAYCON.INP có cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương N là số lượng phần tử của dãy A .
- Dòng 2: Ghi N số nguyên là giá trị của các phần tử trong dãy A , các số được ghi cách nhau ít nhất một dấu cách.
- Dòng 3: Ghi số nguyên dương M là số lượng phần tử của dãy B .
- Dòng 4: Ghi M số nguyên là giá trị của các phần tử trong dãy B , các số được ghi cách nhau ít nhất một dấu cách.

Kết quả ra: Ghi ra file văn bản DAYCON.OUT theo cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương K là số lượng phần tử của dây C .
- Dòng 2: Ghi K số nguyên là giá trị của các phần tử trong dây C , các số được ghi cách nhau một dấu cách.
- Dòng 3: Ghi K số nguyên dương lần lượt là chỉ số của các phần tử trong dãy A tương ứng với các giá trị của phần tử đó trong dây C , các số được ghi cách nhau một dấu cách.
- Dòng 4: Ghi K số nguyên dương lần lượt là chỉ số của các phần tử trong dãy B tương ứng với các giá trị của phần tử đó trong dây C , các số được ghi cách nhau một dấu cách.

¹ Tương tự bài xâu con chung dài nhất

Ví dụ:

DAYCON.INP	DAYCON.OUT
6	4
9 3 1 12 6 15	3 12 6 15
5	2 4 5 6
3 12 7 6 15	1 2 4 5

Ý tưởng:

Thực hiện tương tự bài toán xâu con chung dài nhất

MIC2C. BÀI 74. Số lượng nhóm đề tài

(câu 3, HSG Hà Tĩnh, lớp 10, 2012-2013)

Nhà trường phát động phong trào đăng kí làm sáng tạo khoa học kỹ thuật, tất cả các bạn trong lớp của Nguyễn đều tích cực tham gia và được phân công vào các nhóm đề tài. Mỗi nhóm đề tài được kí hiệu: <Tên nhóm> <Số thành viên>, ví dụ Nguyễn được phân công vào nhóm TIN gồm 3 thành viên thì kí hiệu nhóm là TIN 3. Danh sách được lập ra gồm kí hiệu nhóm và tên thành viên, nhưng trong quá trình in ấn cột kí hiệu nhóm bị mờ <tên nhóm> và không đọc được chỉ còn lại <số thành viên>.

Ví dụ:

Kí hiệu	Thành viên
TIN 3	Việt
TOAN 2	Tuấn
TIN 3	Thái
TIN 3	Anh
TOAN 2	Chính

Do lỗi in ấn \rightarrow

ịệu	Thành viên
3	Việt
2	Tuấn
3	Thái
3	Anh
2	Chính

Yêu cầu: Cho danh sách gồm n học sinh và số thành viên của nhóm tương ứng với từng học sinh. Hãy xác định số lượng nhóm đề tài đã được phân công. Dữ liệu đảm bảo bài toán có nghiệm.

Ví dụ:

Dữ liệu vào	Kết quả
N= 5 3 2 3 3 2	2
N= 10 5 1 2 5 5 2 5 5 2 2	4

MIC2C. BÀI 75. Siêu nguyên tố

(câu 2, HSG Đắk Lắk, 2011-2012)

Số siêu nguyên tố là số nguyên tố mà khi bỏ một số tùy ý các chữ số bên phải của nó thì phần còn lại vẫn tạo thành một số nguyên tố.

Ví dụ 7331 là một số siêu nguyên tố có 4 chữ số vì 733, 73, 7 cũng là các số nguyên tố.

Viết chương trình nhập dữ liệu vào là một số nguyên N ($0 < N < 10$) và đưa ra kết quả là các số siêu nguyên tố có N chữ số cùng số lượng của chúng.

MIC2C. BÀI 76. Cấp số cộng

(câu 1, HSG Thanh Hóa, 2010-2011)

Cho dãy gồm n số nguyên a_1, a_2, \dots, a_n .

Yêu cầu: cho biết dãy số trên có lập thành cấp số cộng hay không?

Dữ liệu vào: từ file CSC.INP gồm:

- Dòng đầu tiên ghi số n .

- Dòng tiếp theo lần lượt ghi n số a_1, a_2, \dots, a_n .

Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách trống.

Kết quả: ghi ra file CSC.OUT như sau:

- Số 1 - Nếu dãy số là cấp số cộng.

- Số 0 - Nếu dãy số không phải là cấp số cộng.

Ví dụ:

CSC.INP	CSC.OUT
4 5 6	1

CSC.INP	CSC.OUT
-3 4 5	0

MIC2C. BÀI 77. Đếm nhóm bạn trong hội trại

(câu 3, HSG Quảng Bình, 2012-2013)

Trong một Hội trại hè do Tỉnh Đoàn tổ chức, có N học sinh tham gia, trong đó, có một số học sinh quen nhau. Một số học sinh được gọi là cùng 1 nhóm bạn, nếu bất kì một học sinh nào thuộc nhóm đều có quen ít nhất 1 học sinh khác trong cùng nhóm đó.

Yêu cầu: Hãy đếm xem có bao nhiêu nhóm bạn trong N học sinh tham gia Hội trại.

Dữ liệu vào: Cho trong file văn bản NHOMBAN.INP, có cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương N, là số lượng học sinh tham gia Hội trại.

($1 \leq N \leq 100$).

- Trong N dòng tiếp theo: Mỗi dòng ghi N số nguyên dương $a[i,j]$ với ý nghĩa:

$a[i,j] = 1$ nếu học sinh i quen học sinh j (với $i \neq j$).

$a[i,j] = 0$ nếu học sinh i không quen học sinh j (với $i \neq j$).

$a[i,i] = 1$ (học sinh i được xem là quen bản thân nó).

Các số trên cùng một dòng được ghi cách nhau ít nhất một dấu cách.

Dữ liệu ra: Ghi ra file văn bản NHOMBAN.OUT, theo cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương K, là số lượng nhóm bạn tìm được trong N học sinh tham gia Hội trại.

Ví dụ:



M1C2C. BÀI 78. Tổng các chữ số

(câu 1, Khu Vực Tây Hồ - Hà Nội 2011-2012)

Viết chương trình nhập vào số nguyên dương N và tính tổng các chữ số là số lẻ của số N?

Dữ liệu vào: cho trong tệp văn bản SUM.INP có dạng:

Một dòng duy nhất chứa số N (N tối đa 100 chữ số).

Kết quả ra: ghi vào tệp văn bản SUM.OUT có dạng:

Một dòng duy nhất chứa tổng các số là số lẻ của số N.

Ví dụ:

SUM.INP	SUM.OUT
29475	21

(Có 50% số test $N \leq 2$ tỷ)

M1C2C. BÀI 79. Phân tích một số thành tổng các số

Cách 1

1. Ta sẽ lưu nghiệm trong mảng x, ngoài ra có một mảng t. Mảng t xây dựng như sau: t_i sẽ là tổng các phần tử trong mảng x từ x_1 đến x_i : $t_i = x_1 + x_2 + \dots + x_i$.

2. Ta sẽ liệt kê các dãy x có tổng các phần tử đúng bằng n, để tránh sự trùng lặp ta đưa thêm ràng buộc $x_{i-1} \leq x_i$.

3. Vì số phần tử thực sự của mảng x là không cố định nên thủ tục PrintResult dùng để in ra 1 cách phân tích phải có thêm tham số cho biết sẽ in ra bao nhiêu phần tử.

4. Thủ tục đệ quy Try(i) sẽ thử các giá trị có thể nhận của x_i ($x_i \geq x_{i-1}$)

5. Khi nào thì in kết quả và khi nào thì gọi đệ quy tìm tiếp?

Lưu ý rằng t_{i-1} là tổng của tất cả các phần tử từ x_1 đến x_{i-1} do đó

• Khi $t_i = n$ tức là ($x_i = n - t_{i-1}$) thì in kết quả

• Khi tìm tiếp, x_{i+1} sẽ phải lớn hơn hoặc bằng x_i . Mặt khác t_{i+1} là tổng của các số từ x_1 tới x_{i+1} không được vượt quá n. Vậy ta có $t_{i+1} \leq n \Leftrightarrow t_{i-1} + x[i] + x[i+1] \leq n \Leftrightarrow x[i] + x[i+1] \leq n - t_{i-1}$ tức là $x_i \leq (n - t_{i-1})/2$. Ví dụ đơn giản khi $n = 10$ thì chọn $x_1 = 6, 7, 8, 9$ là việc làm vô nghĩa vì như vậy cũng không ra nghiệm mà cũng không chọn tiếp x_2 được nữa.

Một cách dễ hiểu ta gọi đệ quy tìm tiếp khi giá trị x_i được chọn còn cho phép chọn thêm một phần tử khác lớn hơn hoặc bằng nó mà không làm tổng vượt quá n. Còn ta in kết quả chỉ khi x_i mang giá trị đúng bằng số thiếu hụt của tổng $i-1$ phần tử đầu so với n.

6. Vậy thủ tục Try(i) thử các giá trị cho x_i có thể mô tả như sau: (để tổng quát cho $i = 1$ ta đặt $x_0 = 1$ và $t_0 = 0$).

- Xét các giá trị của x_i từ x_{i-1} đến $(n - t_{i-1}) \div 2$, cập nhật $t_i := t_{i-1} + x_i$ và gọi đệ quy tìm tiếp.
- Cuối cùng xét giá trị $x_i = n - t_{i-1}$ và in kết quả từ x_1 đến x_i .

```
program Analyses;
var
```

```

n: Integer;
x: array[0..100] of Integer;
t: array[0..100] of Integer;
Count: Longint;

procedure Init;
begin
  Write('n = '); Readln(n);
  x[0] := 1;
  t[0] := 0;
  Count := 0;
end;

procedure PrintResult(k: Integer);
var
  i: Integer;
begin
  Inc(Count);
  Write(Count:10, '. ', n, ' = ');
  for i := 1 to k - 1 do Write(x[i], '+');
  Writeln(x[k]);
end;

procedure Try(i: Integer);
var
  j: Integer;
begin
  for j := x[i - 1] to (n - t[i - 1]) div 2 do
    begin
      x[i] := j;
      t[i] := t[i - 1] + j;
      Try(i + 1);
    end;
  x[i] := n - t[i - 1];
  PrintResult(i);
end;

begin
  Init;
  Try(1);
end.

```

{Tru

Cách 2: Giải bằng quy hoạch động

Nếu bài toán chỉ yêu cầu đưa ra số cách phân tích thì ta có thể làm theo cách sau:

Để giải bài toán này, trong cách 1 ta đã dùng phương pháp liệt kê tất cả các cách phân tích và đếm số cấu hình. Bây giờ ta thử nghĩ xem, **có cách nào tính ngay ra số lượng các cách phân tích mà không cần phải liệt kê hay không?** Bởi vì khi số cách phân tích tương đối lớn, phương pháp liệt kê tỏ ra khá chậm. ($n = 100$ có 190569292 cách phân tích).

Nhận xét:

Nếu gọi **$F[m, v]$ là số cách phân tích số v thành tổng các số nguyên dương $\leq m$** . Khi đó: Các cách phân tích số v thành tổng các số nguyên dương $\leq m$ có thể chia làm hai loại:

Loại 1: Không chứa số m trong phép phân tích, khi đó số cách phân tích loại này chính là số cách phân tích số v thành tổng các số nguyên dương $< m$, tức là số cách phân tích số v thành tổng các số nguyên dương $\leq m - 1$ và bằng $F[m - 1, v]$.

Loại 2: Có chứa ít nhất một số m trong phép phân tích. Khi đó nếu trong các cách phân tích loại này ta bỏ đi số m đó thì ta sẽ được các cách phân tích số $v - m$ thành tổng các số nguyên dương $\leq m$ (Lưu ý: điều này chỉ đúng khi không tính lặp lại các hoán vị của một cách). Có nghĩa là về mặt số lượng, số các cách phân tích loại này bằng $F[m, v - m]$

Trong trường hợp $m > v$ thì rõ ràng chỉ có các cách phân tích loại 1, còn trong trường hợp $m \leq v$ thì sẽ có cả các cách phân tích loại 1 và loại 2. Vì thế:

$$F[m, v] = F[m - 1, v] \text{ nếu } m > v$$

$$F[m, v] = F[m - 1, v] + F[m, v - m] \text{ nếu } m \leq v$$

Ta có công thức xây dựng $F[m, v]$ từ $F[m - 1, v]$ và $F[m, v - m]$. Công thức này có tên gọi là **công thức truy hồi** đưa việc tính $F[m, v]$ về việc tính các $F[m', v']$ với dữ liệu nhỏ hơn. Tất nhiên cuối cùng ta sẽ quan tâm đến $F[n, n]$: Số các cách phân tích n thành tổng các số nguyên dương $\leq n$.

Với $n = 5$, bảng F sẽ là:

F	0	1	2	3	4	5
0	1	0	0	0	0	0
1	1	1	1	1	1	1
2	1	1	2	2	3	3
3	1	1	2	3	4	5
4	1	1	2	3	5	6
5	1	1	2	3	5	7

Nhìn vào bảng F , ta thấy rằng $F[m, v]$ được tính bằng tổng của:

Một phần tử ở hàng trên: $F[m - 1, v]$ và một phần tử ở cùng hàng, bên trái: $F[m, v - m]$.

$F[5, 5]$ sẽ được tính bằng $F[4, 5] + F[5, 0]$, hay $F[3, 5]$ sẽ được tính bằng $F[2, 5] + F[3, 2]$.

Chính vì vậy để tính $F[m, v]$ thì $F[m - 1, v]$ và $F[m, v - m]$ phải được tính trước. Suy ra thứ tự hợp lý để tính các phần tử trong bảng F sẽ phải là theo thứ tự từ trên xuống và trên mỗi hàng thì tính theo thứ tự từ trái qua phải.

Điều đó có nghĩa là ban đầu ta phải tính hàng 0 của bảng: $F[0, v] =$ số dãy có các phần tử ≤ 0 mà tổng bằng v , theo quy ước ở đề bài thì $F[0, 0] = 1$ còn $F[0, v]$ với mọi $v > 0$ đều là 0.

Vậy giải thuật dựng rất đơn giản: Khởi tạo dòng 0 của bảng F : $F[0, 0] = 1$ còn $F[0, v]$ với mọi $v > 0$ đều bằng 0, sau đó dùng công thức truy hồi tính ra tất cả các phần tử của bảng F . Cuối cùng $F[n, n]$ là số cách phân tích cần tìm

PROG01_1.PAS * Đếm số cách phân tích số n

program Analyse1; {Bài toán phân tích số}

const

max = 100;

var

F: array[0..max, 0..max] of LongInt;

n, m, v: Integer;

begin

Write('n = '); ReadLn(n);

FillChar(F[0], SizeOf(F[0]), 0); {Khởi tạo dòng 0 của bảng F toàn số 0}

F[0, 0] := 1; {Duy chỉ có $F[0, 0] = 1$ }

for m := 1 to n do {Dùng công thức tính các dòng theo thứ tự từ trên xuống dưới}

for v := 0 to n do {Các phần tử trên một dòng thì tính theo thứ tự từ trái qua phải}

if v < m then $F[m, v] := F[m - 1, v]$

else $F[m, v] := F[m - 1, v] + F[m, v - m]$; **WriteLn(F[n, n], ' Analyses');** {Cuối cùng $F[n, n]$ là số cách phân tích}

end.

Cải tiến thứ nhất

Cách làm trên có thể tóm tắt lại như sau:

Khởi tạo dòng 0 của bảng, sau đó dùng dòng 0 tính dòng 1, dùng dòng 1 tính dòng 2 v.v... tới khi tính được hết dòng n . Có thể nhận thấy rằng khi đã tính xong dòng thứ k thì việc lưu trữ các dòng từ dòng 0 tới dòng $k - 1$ là không cần thiết bởi vì việc tính dòng $k + 1$ chỉ phụ thuộc các giá trị lưu trữ trên dòng k . Vậy ta có thể dùng hai mảng một chiều: Mảng **Current** lưu dòng hiện thời đang xét của bảng và mảng **Next** lưu dòng kế tiếp, đầu tiên mảng **Current** được gán các giá trị tương ứng trên dòng 0. Sau đó dùng mảng **Current** tính mảng **Next**, mảng **Next** sau khi tính sẽ mang các giá trị tương ứng trên dòng 1. Rồi lại gán mảng **Current** := **Next** và tiếp tục dùng mảng **Current** tính mảng **Next**, mảng **Next** sẽ gồm các giá trị tương ứng trên dòng 2 v.v... Vậy ta có cải đặt cải tiến sau:

*** Đếm số cách phân tích số n**

program Analyse2;

const max = 100;

var

Current, Next: array[0..max] of LongInt;

n, m, v: Integer;

begin

```

Write('n = '); ReadLn(n); FillChar(Current, SizeOf(Current), 0);
Current[0] := 1; {Khởi tạo mảng Current tương ứng với dòng 0 của bảng F}
for m := 1 to n do
begin {Dùng dòng hiện thời Current tính dòng kế tiếp Next → Dùng dòng m - 1 tính
dòng m của bảng F}
for v := 0 to n do
if v < m then Next[v] := Current[v]
else Next[v] := Current[v] + Next[v - m];
Current := Next; {Gán Current := Next tức là Current bây giờ lại lưu các phần
tử trên dòng m của bảng F}
end;
WriteLn(Current[n], ' Analyses');
end.

```

*** Cách làm trên đã tiết kiệm được khá nhiều không gian lưu trữ, nhưng nó hơi chậm hơn phương pháp đầu tiên vì phép gán mảng (Current := Next). Có thể cải tiến thêm cách làm này như sau:**

Đếm số cách phân tích số n

```

program Analyse3;
const
max = 100;
var
B: array[1..2, 0..max] of LongInt; {Bảng B chỉ gồm 2 dòng thay cho 2 dòng liên
tiếp của bảng phương án}
n, m, v, x, y: Integer;
begin
Write('n = '); ReadLn(n);
{Trước hết, dòng 1 của bảng B tương ứng với dòng 0 của bảng phương án F, được
điền cơ sở quy hoạch động}
FillChar(B[1], SizeOf(B[1]), 0); B[1][0] := 1;
x := 1; {Dòng B[x] đóng vai trò là dòng hiện thời trong bảng phương án}
y := 2; {Dòng B[y] đóng vai trò là dòng kế tiếp trong bảng phương án}
for m := 1 to n do begin
{Dùng dòng x tính dòng y
→
Dùng dòng hiện thời trong bảng phương án để tính dòng kế tiếp}
for v := 0 to n do
if v < m then B[y][v] := B[x][v]
else B[y][v] := B[x][v] + B[y][v - m];
x := 3 - x; y := 3 - y; {Đảo giá trị x và y, tính xoay lại}
end;
WriteLn(B[x][n], ' Analyses');
end.

```

Cải tiến thứ hai

Ta vẫn còn cách tốt hơn nữa, tại mỗi bước, ta chỉ cần lưu lại một dòng của bảng F bằng một mảng 1 chiều, sau đó dùng mảng đó tính lại chính nó để sau khi tính, mảng một chiều sẽ lưu các giá trị của bảng F trên dòng kế tiếp.

```

PROG01_4.PAS * Đếm số cách phân tích số n
program Analyse4;
const
max = 100;
var
L: array[0..max] of LongInt; {Chỉ cần lưu 1 dòng}
n, m, v: Integer;
begin
Write('n = '); ReadLn(n); FillChar(L, SizeOf(L), 0);
L[0] := 1; {Khởi tạo mảng 1 chiều L lưu dòng 0 của bảng}
for m := 1 to n do {Dùng L tính lại chính nó}
for v := m to n do
L[v] := L[v] + L[v - m]; WriteLn(L[n], ' Analyses');
end.

```

M1C2C. BÀI 80. Phân phối hàng cứu trợ

(Câu 3, HSG Cà Mau, 2012-2014)

Hai trận bão vừa qua đã tàn phá nặng nề khu vực miền trung của nước ta. Đồng bào vùng chịu ảnh hưởng của bão đang gặp rất nhiều khó khăn, thiếu thốn. Công tác cứu trợ dù có nhiều trở ngại nhưng đang được thực hiện hết sức khẩn trương và bài bản. Một ủy ban cứu trợ địa phương nhận được N thùng hàng cứu trợ. Tại địa phương có rất nhiều khu vực bị thiệt hại, ủy ban đã đánh giá lập danh sách các khu vực đó theo mức thiệt hại giảm dần. Để đảm bảo công bằng, ủy ban sẽ phân chia hết số hàng viện trợ cho các khu vực sao cho số thùng hàng mỗi khu vực nhận được không nhiều hơn của khu vực bị thiệt hại nặng hơn. Ngoài ra mỗi khu vực không được nhận quá K thùng hàng.

Yêu cầu: Do số người làm công tác cứu trợ không nhiều nên ủy ban muốn xác định xem có bao nhiêu phương án gửi hàng cứu trợ cho các khu vực theo yêu cầu trên.

Dữ liệu vào: cho trong file STORM.INP gồm một dòng chứa hai số N và K ($N < 10000$, $K < 100$)

Dữ liệu ra: Ghi ra file STORM.OUT gồm một số duy nhất là phần dư khi chia cho 10^9 của số cách phân chia hàng cứu trợ.

Ví dụ: nếu có 5 thùng hàng ($N=5$) và mỗi khu vực nhận không quá 3 thùng ($K=3$) thì có 5 cách phân chia sau (ở mỗi cách lần lượt gửi hàng cho các khu vực có thiệt hại giảm dần bắt đầu từ nơi bị thiệt hại nặng nhất)

$$5=3+2$$

$$5=3+1+1$$

$$5=2+1+1+1$$

$$5=2+2+1$$

$$5=1+1+1+1+1$$

Ví dụ

STORM.INP	STORM.OUT
5 3	5

Hướng dẫn: Đây là dạng bài toán đếm số cách phân tích một số N thành tổng các số, ta có thể vận dụng một trong hai cách của bài M1C2C. Bài 79 để giải.

M1C2C. BÀI 81. Tìm điểm thuộc tất cả N hình chữ nhật

(Câu 2, HSG Cà Mau, 2013-2014)

Cho số tự nhiên N và mảng số nguyên M kích thước $4 \times n$ chứa dữ liệu về n hình chữ nhật ($m[1,i], m[2,i]$) và ($m[3,i], m[4,i]$) là các tọa độ góc trái trên và góc phải dưới của hình chữ nhật thứ i . Hỏi có điểm nào trong mặt phẳng tọa độ thuộc tất cả các hình chữ nhật đã cho không? Hãy lập chương trình để giải bài toán đó.

Dữ liệu vào trong tệp DIEM_HCN.INP có quy cách như sau:

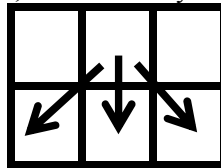
+ Dòng đầu tiên là số n ;

+ n dòng tiếp theo, mỗi dòng gồm 4 số là tọa độ $m[1,i]$, $m[2,i]$, $m[3,i]$, $m[4,i]$ tương ứng, các số phân cách nhau bởi dấu cách.

Kết quả thể hiện trên màn hình.

M1C2C. BÀI 82. Con thạch sùng

Sau trận mưa tối hôm qua, một bức tường trong nhà có rất nhiều muỗi đậu trên đó. Bức tường có kích thước $h \times w$ ô vuông, trong đó h là số hàng được đánh số từ 1 đến h theo chiều từ trên xuống dưới và w là số cột được đánh số từ 1 đến w theo chiều từ trái qua phải. Tại mỗi ô vuông đã có khoảng từ 1 đến 1000 con muỗi đậu ở đó. Một con thạch sùng muốn ăn nhiều nhất muỗi có thể, tùy thuộc vào hạn chế như sau: Nó bắt đầu chọn một ô nhiều muỗi nhất ở hàng trên cùng và ăn số muỗi tại đây. Sau đó chuyển xuống một ô vuông ở hàng bên dưới, tiếp tục ăn muỗi tại đó, con thạch sùng cứ chuyển như vậy đến hết hàng ngang cuối cùng để ăn số muỗi trên mỗi ô. Khi chuyển xuống ô tiếp theo của hàng dưới, nó có thể chuyển theo một trong 3 hướng như hình vẽ.



Yêu cầu: Cho h và w và số muỗi trong mỗi ô, con thạch sùng có thể ăn nhiều nhất là bao nhiêu con muỗi trong một lần duy nhất di chuyển từ hàng trên cùng xuống hàng cuối cùng?

Dữ liệu vào: Dòng đầu ghi hai số nguyên, số đầu tiên là h – số hàng, số nguyên thứ hai là w – số cột. Dòng thứ i trong h dòng tiếp theo ghi w số nguyên m là số muỗi trong mỗi ô. Tất cả các số nguyên đều cách nhau một dấu cách.

Dữ liệu ra: Một dòng duy nhất ghi một số nguyên duy nhất là số muỗi lớn nhất mà con thạch sùng có thể ăn được trong một lần di chuyển duy nhất từ hàng trên cùng xuống hàng dưới cùng.

Ví dụ:

Input	Output
6 5	32
3 1 7 4 2	
2 1 3 1 1	
1 2 2 1 8	
2 2 1 5 3	
2 1 4 4 4	
5 7 2 5 1	

Hạn chế: $1 \leq h \leq 500$.

$1 \leq w \leq 500$.

$1 \leq m \leq 1000$.

Giải thích ví dụ: Con thạch sùng chọn như sau: hàng 1 chọn 7, hàng 2 chọn 1, hàng 3 chọn 8, hàng 4 chọn 5, hàng 5 chọn 4, hàng 6 chọn 7. Tổng: $7 + 1 + 8 + 5 + 4 + 7 = 32$.

Bài toán tương tự: di chuyển từ cột trái qua cột phải sao cho tổng chi phí di chuyển là lớn (nhỏ) nhất.



HẾT

Nguyễn Văn Mộng
 Trường THPT Cà Mau
 Liên hệ Email: nvmong.camau@gmail.com
 SĐT: 01292451547