ToPAS'06
Torneio de Programação para Alunos do Secundário

Departamento de Ciência de Computadores

http://www.dcc.fc.up.pt/topas/

Conjunto de Problemas

U.PORTO

Faculdade de Ciências da Universidade do Porto

13 de Maio de 2006

Este conjunto de problemas deverá conter oito (8) problemas e dezoito (18) páginas.
Se faltar algum problema, por favor avise a organização.

ToPAS'06
Torneio de Programação para Alunos do Secundário

Dep. Ciência de Computadores – FCUP
13 de Maio de 2006

Contents

Problema A: Prendam os suspeitos do costume! 3
Problema B: ET em Estado de Choque 5
Problema C: Cigarras Tontas 7
Problema D: Telefones de Emergência 9
Problema E: Numerais Romanos 11
Problema F: Combate Final 13
Problema G: Jogo dos Segredos 15
Problema H: Onde vem a tartaruga? 17

Problema A

Prendam os suspeitos do costume!

Problema

Uma conhecida multinacional de *software*, que prefere permanecer anónima, foi vítima dum ataque informático ao repositório de código da próxima versão de um dos seus produtos mais difundidos. Os *crackers* introduziram uma série de vírus que assim foram propagados pela primeira vez através de *software* licenciado!

O ataque foi tão bem feito que ainda não foi possível determinar a sua origem. Sabe-se apenas que os autores terão conseguido acesso à rede da empresa usando uma ou mais contas dos funcionários da empresa. Comparando as datas de modificações dos ficheiros alterados determinou-se um intervalo de tempo em que foi realizado o ataque.

Neste momento todos os funcionários da empresa são suspeitos, o clima é insustentável e a investigação deste incidente não avança. Para reduzir o número de suspeitos foi proposto comparar os registos de autenticação dos funcionários. Estes registos indicam a data e hora quer do início, quer do fim de cada sessão de trabalho. Um funcionário será suspeito se uma das suas sessões de trabalho se sobrepuser com período em que ocorreu o ataque. Portanto, uma sessão será suspeita se tiver início antes do fim do ataque e terminar depois do início do ataque.

Tarefa

Escrever um programa para determinar o **número** de suspeitos que seriam localizados deste modo.

Input

Na primeira linha dos dados tem os instantes inicial e final do ataque. Depois segue-se um inteiro n com o número de sessões a processar. As n linhas seguintes descrevem as sessões, ordenadas temporalmente: em cada linha tem o número do funcionário, o início e fim da respectiva sessão.

A empresa não tem mais do que 100 funcionários. Nos dados fornecidos, os momentos de início e fim do ataque e das sessões, são inteiros não negativos (representáveis em 32 bits). Os tempos

estão todos indicados em segundos, correspondendo a um intervalo que decorreu desde um certo instante inicial.

Output

O output do programa resume-se ao número de suspeitos detectados (termina por mudança de linha)

Exemplo de input

```
2574 3390
6
10 1760 2330
2 1761 2574
13 2341 2522
10 3377 4885
3 3393 7811
1 4173 4814
```

Exemplo de output

```
1
```

Problema B

ET em Estado de Choque

Problema

Ontem, dia 12 de Maio, foi encontrado um OVNI, com uma vida extra-terrestre (ET) a bordo, no maravilhoso jardim do futuro edifício do Departamento de Ciência de Computadores. O ET foi retirado da nave sem qualquer ferimento físico, mas em aparente estado de choque. Chamado ao local, o INEM nada conseguiu fazer para avaliar o seu estado, transportando-o imediatamente para o hospital mais próximo.

Reconhecidos cientistas portugueses trabalharam então em conjunto para tentar avaliar o seu estado de saúde. As descobertas foram impressionantes. Segundo o boletim clínico, divulgado após várias horas, tinham conseguido desvendar completamente o código genético do ET. O seu ADN era apenas constituído por 3 elementos distintos: `c`, `t` e `a` (o genoma humano é constituído por 4 elementos). Os cientistas também perceberam que o ADN do ET sofria uma mutação por cada hora que passava, seguindo um padrão constante. O primeiro elemento do ADN era mantido mas os restantes elementos do ADN eram alterados de acordo com a regra seguinte. Para cada par de letras vizinhas, a segunda era alterada se era diferente da anterior. Nesse caso, era substituída pela letra que não estava no par. Por exemplo, para a sequência de ADN `actt`, a mutação seria:

a primeira letra continua `a`
a segunda letra do par `ac` evolui para `t`
a segunda letra do par `ct` evolui para `a`
a segunda letra do par `tt` continua `t`

e o resultado final seria `atat`. As alterações numa sequência ocorriam em simultâneo.

Tarefa

Escrever um programa que, dado um estado do ADN do ET, calcule qual será o seu estado passado uma hora.

Input

O input é uma sequência de letras (`c`, `t` ou `a`), arbitrariamente longa, representando o ADN do ET, e que terminará por `#`. Nos dados, tem um símbolo por linha. Todas as linhas têm um caracter de mudança de linha.

Output

O output é a sequência do ADN depois de uma mutação. Terminará por mudança de linha.

Exemplo 1

Input

```
a
t
c
c
a
#
```

Output

```
a
c
a
c
t
```

Exemplo 2

Input

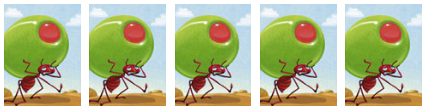
```
t
t
a
a
c
#
```

Output

```
t
t
c
a
t
```

Problema C

Cigarras Tontas



Problema

Fartas de ouvirem dizer que passavam o dia a cantar, algumas cigarras resolveram imitar as suas rivais. Não percebendo bem a lógica da deslocação em carreiros, fomos encontrá-las atarantadas: tendo saído dum certo local em fila indiana, começaram a andar às voltas, passando sistematicamente por locais onde anteriormente já tinham estado, como se andassem perdidas num labirinto. Finalmente encontraram o local onde pretendiam chegar. De qualquer modo, gostariamos de saber o que poderiam ter feito se tivessem tido um melhor sentido de orientação e não tivessem "visitado" os mesmos locais repetidamente.

Tarefa

Escrever um programa que analise a sequência de locais que foram "visitando", a qual inclui o ponto de partida e no fim o de chegada, e encontre um trajecto que respeite essa sequência mas retire todas as voltas desnecessárias. Só a última passagem em cada um desses locais poderá ser considerada correcta. Todas as outras constituem enganos. Em caso de engano, toda a volta que as cigarras deram, desde que passaram num dado local até que lá voltaram, será considerada desnecessária. O local de chegada não foi visitado duas vezes, mas qualquer um dos restantes, incluindo a origem, pode ter sido.

Input

É dada apenas a sequência de locais onde as cigarras foram passando, um local por cada linha de dados. A última linha tem o valor 0. O número de locais distintos não excede 30. Cada local é identificado por um inteiro positivo inferior a 10000.

Output

A sequência de locais que visitariam se não se tivessem enganado, cada local numa linha.

Exemplo 1

Input

```
1600
15
2315
15
1315
0
```

Output

```
1600
15
1315
```

Exemplo 2

Input

```
1513
3171
178
1316
1600
1513
1774
178
3171
672
1315
0
```

Output

```
1513
1774
178
3171
672
1315
```

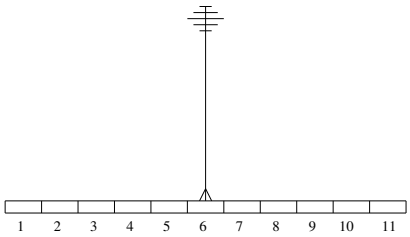
Problema D

Telefones de Emergência

Problema

Uma determinada concessionária de auto-estradas, pretende instalar uma rede de telefones de emergência. Contudo, devido ao preço exorbitante dos cabos eléctricos nos últimos tempos, a concessionária está a tentar resolver o problema de enviar electricidade aos telefones de uma forma que não a obrigue a colocar cabos ao longo de toda a auto-estrada. Embora sejam amplamente adoptados noutras auto-estradas, nesta zona os painéis solares não são uma opção, pois está permanentemente mau tempo.

Contudo nem tudo está perdido. Recentemente foi inventada uma nova tecnologia de transmissão de electricidade sem fios que, recorrendo a sofisticadíssimas antenas direccionais permite o envio directo de todo o sinal para um dado telefone, sempre que este iniciar uma chamada. Com esta tecnologia de transmissão de electricidade, a potência eléctrica decresce apenas uma unidade por cada quilómetro que o sinal percorre.



Apesar de entusiasmados com a ideia, os responsáveis, zelosos pelas questões de segurança, pretendem confirmar que todos os telefones de emergência terão electricidade suficiente para funcionar.

Tarefa

Escrever um programa que dadas as posições das antenas e dos telefones e as suas potências (de emissão e de funcionamento), verifique se os telefones poderão ou não funcionar. Um telefone poderá receber simultaneamente sinais de várias antenas, sendo a potência total recebida igual ao somatório das potências recebidas de cada antena. Assuma que não será utilizado mais do que um telefone em simultâneo.

Input

Corresponde a um plano de instalação de N antenas e de localização de M telefones de emergência. Na primeira linha tem um inteiro L que indica a extensão em Km (medida em linha recta) da região em que serão instaladas as antenas, $1 \leq L \leq 1000$. Nas linhas seguintes tem o valor de N ($1 \leq N \leq 100$) e depois N pares de linhas: a primeira linha de cada par tem a posição x da antena ($1 \leq x \leq L$) e a segunda a sua potência p ($1 \leq p \leq 64$), medida numa unidade especialmente desenvolvida para estas antenas. Depois encontrará o número de telefones M ($1 \leq M \leq 500$) e M pares de linhas: a primeira linha de cada par tem a posição x de cada telefone ($1 \leq x \leq L$) e a segunda, a potência p necessária para o seu funcionamento ($1 \leq p \leq 64$).

Output

É composto por M linhas. A i-ésima linha indica se o i-ésimo telefone de emergência fica funcional ou não: terá `SIM` se ficar e `NAO` caso contrário. Todas terminam por mudança de linha.

Exemplo - input

```
10
2
1
5
10
5
4
5
2
5
1
10
6
8
3
```

Exemplo - output

```
NAO
SIM
NAO
SIM
```

Problema E

Numerais Romanos

Problema

A civilização clássica romana desenvolveu um sistema de numeração que perdurou durante muito séculos e que ainda hoje é usado em referências particulares: "D. Manuel II", "séc. XXI", "ano MCMXCVIII", por exemplo.

Na notação romana usam-se as letras I, V, X, L, C, D, M para representar quantidades de 1 a 1000:

I	1
V	5
X	10
L	50
C	100
D	500
M	1000

A principal diferença entre a notação romana e a árabe (que usamos hoje em dia) é que os valores dos símbolos não são posicionais, mas sim adicionados ou subtraídos conforme a ordem por que ocorrem. Primeiramente, os símbolos repetidos são agrupados e o seu valor adicionado: III é $1+1+1=3$, XX é $10+10=20$, MMM é $1000+1000+1000=3000$. Seguidamente, analisamos os grupos de símbolos da direita para a esquerda:

- O grupo mais à direita é sempre **positivo**.
- Se um grupo vale mais em módulo que o seu vizinho à direita, o valor tem sinal **positivo**: XII é $(1+1)+10=12$, LVIII é $(1+1+1)+5+50=58$.
- Se o grupo vale menos em módulo que o vizinho à direita, o seu valor tem sinal **negativo**: IX é $10+(-1)=9$, XXM é $1000+(-10-10)=980$.
- Não podem ocorrer grupos de símbolos distintos com valores igual (correspondem a numerais inválidos como "XVV" ou "VIIIII").

Assim, o numeral romano MCMXCVIII vale $(1+1+1)+5+100-10+1000-100+1000=1998$.

Tarefa

Escrever um programa que lê um numeral romano e escreve o seu valor em decimal.

Input

O input é um numeral romano numa única linha, em carácteres maiúsculas, de comprimento inferior a 20 carácteres. Pode assumir que o numeral é sempre válido e que o seu valor não excede 5000.

Output

O output deve ter o valor do numeral romano em decimal (seguido de mudança de linha).

Exemplo 1

Input
```
CCCLXXIX
```

Output
```
379
```

Exemplo 2

Input
```
CDXLVII
```

Output
```
447
```

Exemplo 3

Input
```
MMMDCCCXLVIII
```

Output
```
3848
```

Problema F

Combate Final

Problema

Dois amigos encontram-se para um torneio de cartas mágicas. Cada um tem um baralho de cartas com monstros, contendo cada carta um número inteiro que indica a força/poder de combate do personagem. Cada jogador empilha "estrategicamente" n cartas (número igual para os dois jogadores). O desenvolvimento de um jogo ocorre numa sequência de combates da seguinte forma. O jogador A e o jogador B tiram a carta de topo do seu baralho.

- Se a carta do jogador A for mais poderosa do que a do jogador B, então o jogador A ganha esse combate, e a carta do jogador B fica para o jogador A. Coloca na base do seu baralho primeiro a sua carta e depois a de B.
- Se a carta do jogador B tiver mais poder de ataque que a carta do jogador A, então B ganha esse combate. Neste caso, é o jogador B que fica com a carta do jogador A. Coloca na base do seu baralho primeiro a sua carta e depois a de A.
- Se ambas as cartas tiverem o mesmo poder de ataque então ocorre um empate. Neste caso, cada jogador coloca a sua carta na base do seu baralho.

O jogo vai prosseguindo de modo análogo. A particularidade e interesse do jogo advém do facto das cartas serem mágicas. De tal forma que quando duas cartas se encontram, independentemente do resultado do combate, o poder da carta é decrementado de um (porque o monstro fica cansado do combate e perde energia). Se o valor da carta chegar a 0 ela é retirada de jogo. Esta regra introduz interesse no jogo, pois caso contrário o jogador com a carta mais alta nunca poderia perder o jogo.

O jogador que durante uma sequência de combates perder todas as cartas perde o jogo. Para evitar jogos muito longos, se ao fim de $n^3 + n$ jogadas nenhum dos jogadores tiver ganho, o jogo é considerado como empatado. Se ambos os jogadores ficarem sem cartas simultaneamente o jogo também fica empatado.

Tarefa

Implementar um programa que, dados os baralhos dos jogadores, simule o jogo para determinar qual dos dois jogadores ganha.

Input

Na primeira linha tem o número n de cartas com que jogarão ($1 \leq n \leq 20$). Nas linhas seguintes tem a descrição das n cartas do baralho do jogador A (isto é, n inteiros que representam o poder das cartas escolhidas, do topo até à base). Nas n restantes, tem a descrição das n cartas do jogador B.

Output

O programa deve escrever `A`, `B` ou `E`, para indicar o jogador que ganha o jogo. Escreverá `E` se tiver havido empate. A linha termina por mudança de linha.

Exemplo 1

Input

```
3
3
5
2
2
1
9
```

Output

```
B
```

Exemplo 2

Input

```
5
4
2
1
6
8
3
3
4
2
1
```

Output

```
A
```

Problema G

Jogo dos Segredos

Problema

No intervalo de aulas alguns alunos estão a jogar aos Segredos, agrupados em círculos com pelo menos três alunos. Um certo elemento num círculo diz um segredo ao seu vizinho esquerdo. Este transmite-o ao seguinte e assim sucessivamente até o segredo chegar primeiro, por vezes verdadeiramente distorcido! Quanto maior é o número de elementos do círculo mais distorcido chega o segredo.

Tarefa

Escrever um programa para determinar o tamanho do maior círculo num cenário que envolve n alunos, numerados de 1 a n, possivelmente agrupados num ou mais círculos (de tamanhos variados). É dada uma sequência $a_1, a_2, \ldots, a_n$ em que a_1 é o aluno que estava à esquerda do aluno 1, a_2 é o aluno que estava à esquerda do aluno 2, e assim sucessivamente.

Input

Na primeira linha tem o número de alunos (sabe-se que $n \leq 100$). Nas linhas seguintes tem a sequência $a_1\, a_2\, \ldots\, a_n$ que define o cenário.

Output

Escreverá o maior número de alunos num círculo, seguido de mudança de linha.

Exemplo 1

Input

```
3
2
3
1
```

Ouput

```
3
```

Exemplo 2

Input

```
9
3
4
2
8
6
7
9
1
5
```

Ouput

```
5
```

Problema H

Onde vem a tartaruga?

Problema

Numa prova de atletismo, ainda não introduzida nos concursos mundiais por falta de consenso, a classificação de cada equipa é dada pelo número de elementos que chegaram à meta e pelo tempo de chegada do último. É usado um critério de desempate quando atletas de equipas diferentes cruzam a meta ao mesmo tempo e as equipas a que pertencem também terminaram a prova com o mesmo número de atletas. Se, a imagem gravada não permitir ver qual o atleta que esticou mais o pescoço, o peso total dos elementos da equipa é usado para as desempatar. Assim, não haverá empates!

Cada equipa é constituída por três atletas. É possível que alguns atletas desistam. Ficam classificadas nos lugares mais cimeiros as equipas em que mais atletas conseguiram terminar a prova. Uma equipa não é classificada se nenhum dos seus atletas cruzar a meta.

Tarefa

Escrever um programa que dada uma sequência de chegadas determina a ordem de classificação das equipas (em que pelo menos um atleta terminou a prova).

Input

A primeira linha tem o número de equipas inscritas na prova. As restantes linhas contêm a sequência de chegada dos atletas (já desempatados). Cada valor identifica apenas o número da equipa a que pertence o atleta que chegou na posição correspondente. Os dados terminam por -1.

As equipas são identificadas por inteiros consecutivos maiores ou iguais a 1. Não há mais do que 200 equipas.

Output

Tantas linhas quanto o número de equipas classificadas: a primeira linha indica o número da equipa que ficou em primeiro lugar, a segunda será o da classificada em segundo e assim su-

cessivamente. No fim escreve o número de equipas não classificadas, precedido da mensagem "`No. abandonos (equipa toda): `". Todas as linhas terminam por mudança de linha.

Exemplo: input

```
8
1
2
5
5
2
3
4
5
2
3
4
3
7
1
-1
```

Exemplo: output

```
5
2
3
1
4
7
No. abandonos (equipa toda): 2
```