



Olimpíadas Nacionais de Informática (ONI'2008)

Relatório de Actividades

Outubro 2008



Patrocínio Principal



Patrocínio

FUNDAÇÃO
CALOUSTE
GULBENKIAN

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR



Apoio



Microsoft



ÍNDICE

1. Introdução	4
2. Desenrolar das provas.....	6
3. Conclusões	8
ANEXOS	9
Anexo I - Regulamento.....	9
Anexo II - Sítio na <i>Web</i> (http://www.dcc.fc.up.pt/oni/2008/)	12
Anexo III - Problemas do concurso de Qualificação.....	15
Anexo IV - Problemas do concurso da Final Nacional.....	22
Anexo V - Classificações do concurso de Qualificação.....	29
Anexo VI - Classificações da Final Nacional	32

1. Introdução

As Olimpíadas Nacionais de Informática são um concurso de âmbito nacional, promovido e organizado pela APDSI – Associação para a Promoção e Desenvolvimento da Sociedade da Informação - em colaboração com a Universidade do Algarve (Departamento de Engenharia Electrónica e Informática da Faculdade de Ciências e Tecnologia) e com a Universidade do Porto (Departamento de Ciência de Computadores da Faculdade de Ciências), destinado aos jovens que frequentam o ensino secundário ou o ensino básico em todo o território nacional, constituído por uma prova nacional realizada em duas fases.

As Olimpíadas Nacionais de Informática têm como objectivo promover o gosto pela programação e pelas tecnologias de informação entre os jovens, constituir um ponto de encontro de âmbito nacional para professores e alunos interessados nestes temas e servir para seleccionar uma equipa para representar Portugal nas Olimpíadas Internacionais de Informática.

Nas Olimpíadas Nacionais de Informática, através da resolução de problemas de programação, os concorrentes podem aplicar os seus conhecimentos de Informática e de Programação, em competição leal e salutar com os seus colegas de outras escolas.

Puderam concorrer, este ano, todos os jovens nascidos após 30 de Junho de 1988 e que, no ano lectivo 2007-2008 frequentassem o ensino básico, secundário ou equivalente (regulamento em anexo).

Nas Olimpíadas Nacionais de Informática os concorrentes representam a escola a que pertencem.

A primeira fase da prova nacional, dita prova preliminar ou de qualificação, consiste na resolução de três problemas apresentados simultaneamente a todos os concorrentes através da Internet por meio de um sistema de gestão de concursos de programação, problemas esses que deverão ser resolvidos individualmente pelos concorrentes, utilizando a linguagem de programação Pascal em ambiente Free Pascal (www.freepascal.org) ou as linguagens C ou C++ em ambiente GCC (<http://gcc.gnu.org>) e submetidos através do mesmo sistema, para avaliação.

Antes da prova preliminar, a organização disponibiliza na Internet um concurso de treino, para que os concorrentes se possam familiarizar com o sistema.

A classificação é calculada automaticamente pelo sistema em função do número de casos de teste que os programas submetidos pelos concorrentes conseguem resolver e é depois validada por um júri nacional, nomeado pela organização, o qual também desempatará os casos em que isso seja necessário, observando a qualidade da programação.

Os melhores concorrentes da primeira fase, até um máximo de 30, participarão na segunda fase do concurso, que constitui a prova final.

A prova final é individual e é realizada usando o mesmo sistema da prova preliminar, mas presencialmente.

A classificação final do concurso, a atribuir pelo júri nacional, é a da prova final, atribuída nos mesmos moldes da prova preliminar.

Este ano de 2008 a prova final realizou-se no dia 16 de Maio nas instalações do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto.

2. Desenrolar das provas

As operações relacionadas com as Olimpíadas Nacionais de Informática foram suportadas por um sítio na Internet, <http://www.dcc.fc.up.pt/oni/2008/>, cuja página de entrada se anexa.

As provas desenrolaram-se de acordo com o planeado: primeiro houve o concurso de treino, com o objectivo de, por um lado, dar a conhecer aos participantes o tipo de problemas que saem nas Olimpíadas e, por outro, de os familiarizar com o sistema de submissão e avaliação automática de problemas.

Dos 207 alunos que se tinham registado, apenas 72 tentaram submeter alguma resolução. Provavelmente isto significa que no momento da inscrição grande parte dos potenciais concorrentes subvaloriza a dificuldade dos problemas, e que, ao tomarem conhecimento deles, imediatamente reconheceram não estar preparados para os resolver. Este fenómeno tem-se repetido todos os anos.

Na prova de qualificação participaram 72 alunos, e desses, 51 tiveram uma submissão aceite. Como a capacidade para a final nacional era de 30 alunos, participaram apenas os 30 melhor classificados

As provas de treino e de qualificação realizaram-se na Internet. A final realizou-se no Porto, nas instalações do Departamento de Ciência de Computadores (DCC) da Faculdade de Ciências da Universidade do Porto. Dos 30 concorrentes seleccionados compareceram 28 acompanhados pelos seus professores. Os dois alunos ausentes foram punidos pela respectiva escola por terem tentado praticar uma fraude durante a prova.

Seguiu-se, num anfiteatro do DCC, a entrega de certificados de participação, medalhas e prémios de participação a todos os finalistas.

A classificação da final foi apresentada no fim de um jantar que teve lugar no hotel Ipanema Porto seguindo-se a entrega dos prémios, das medalhas e dos certificados de classificação aos 4 primeiros classificados.

Os problemas do concurso de qualificação e da final são problemas originais, desenvolvidos pelo júri, e encontram-se também em anexo. Também em anexo, se encontram as classificações dessas duas provas, tal como produzidas pelo sistema de avaliação automática. A classificação das Olimpíadas Nacionais de Informática é a classificação da prova final.

Os oito alunos mais bem classificados foram convidados a realizar um estágio de formação, no final de Julho. No final desse estágio, de entre os oito, serão seleccionados os quatro que formarão a delegação portuguesa às Olimpíadas Internacionais de Informática, que este ano decorrem no Cairo, Egipto.

O júri das Olimpíadas Nacionais de Informática, edição de 2008, teve a seguinte composição:

- Dr.^a Anabela Rodrigues Rocha, Escola Secundária Frei Heitor Pinto (Covilhã)
- Prof. José Maria Fernandes de Almeida, APDSI
- Prof. Pedro João Valente Dias Guerreiro, Universidade do Algarve
- Dr. Pedro Manuel Pinto Ribeiro, Universidade do Porto

3. Conclusões

Inscreeveram-se 207 alunos de 59 escolas distribuídas geograficamente por todo o país.

Verificou-se um acréscimo de 8% no quantitativo dos alunos em relação aos quantitativos no ano anterior. Já desde 2004 se tem verificado um acréscimo no quantitativo de alunos inscritos.

A distribuição geográfica apresenta uma maior tendência para a participação de alunos (76%) e escolas (76%) do litoral com uma participação mais significativa de alunos (64%) e escolas (53%) da zona a Norte do rio Mondego.

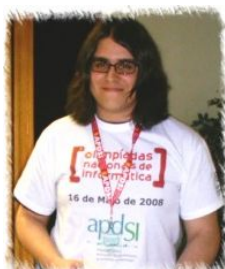
De notar que no ano de 2007 e no ano de 2008 se verificou a inscrição e participação na final de um aluno proveniente de uma escola do Funchal.

Tal como referimos atrás, dos 207 alunos inscritos, 135 não participaram. Este fenómeno repete-se todos os anos provavelmente por grande parte dos potenciais concorrentes subvalorizarem a dificuldade dos problemas e, ao tomarem conhecimento deles, imediatamente reconhecem não estar preparados para os resolver.

Dos 72 participantes apenas 51 tiveram submissões válidas, tendo sido apurados todos os 30.

Dos 30 finalistas, 2 faltaram por punição da escola que representavam.

As provas de treino e selecção foram realizadas de acordo com o calendário previamente estabelecido e culminaram em 16 de Maio de 2007.



O primeiro classificado foi o aluno, Tiago Andrade, do Colégio Internato dos Carvalhos, de Vila Nova de Gaia, que se tinha deslocado daquela cidade acompanhado pela sua professora. Os alunos classificados nas três posições imediatas vinham da Escola S/3 D. Manuel I, em Beja, da Escola Secundária Padre Benjamim Salgado, em Vila Nova de Famalicão e da Escola Secundária Jaime Moniz, no Funchal.

ANEXOS

Anexo I- Regulamento



20^a Olimpíadas Nacionais de Informática 2008

ONI'2008

Regulamento

1. As Olimpíadas Nacionais de Informática são um concurso de âmbito nacional, promovido e organizado pela APDSI – Associação para a Promoção e Desenvolvimento da Sociedade da Informação –, em colaboração com o Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto, destinado aos jovens que frequentam o ensino secundário ou o ensino básico em todo o território nacional, constituído por uma prova nacional realizada em duas fases.
2. As Olimpíadas Nacionais de Informática têm como objectivo promover o gosto pela programação e pelas tecnologias de informação entre os jovens, constituir um ponto de encontro de âmbito nacional para professores e alunos interessados nestes temas e servir para seleccionar uma equipa para representar Portugal nas Olimpíadas Internacionais de Informática.
3. Nas Olimpíadas Nacionais de Informática, através da resolução de problemas de programação, os concorrentes podem aplicar os seus conhecimentos de Informática e de Programação, em competição leal e salutar com os seus colegas de outras escolas.
4. Podem concorrer todos os jovens nascidos após 30 de Junho de 1988 e que no ano lectivo 2007-2008 frequentem o ensino básico, secundário ou equivalente.
5. A apresentação a concurso deverá ser feita individualmente, por meio de ficha de inscrição própria, disponível no sítio do concurso, em <http://www.dcc.fc.up.pt/oni/>.
6. Cada inscrição deverá incluir a identificação de um professor responsável que apoia a participação do aluno concorrente nas diversas fases do concurso e que deve ser mandatado pela escola a que pertencem o professor e o aluno concorrente.
7. Os contactos entre a organização do concurso e o concorrente serão feitos através do professor responsável ou com conhecimento deste.
8. Nas Olimpíadas Nacionais de Informática os concorrentes representam a escola a que pertencem.

9. A primeira fase da prova nacional, dita prova preliminar, consiste em três problemas apresentados simultaneamente a todos os concorrentes através da Internet por meio de um sistema de gestão de concursos de programação, problemas esses que deverão ser resolvidos individualmente pelos concorrentes, utilizando a linguagem de programação Pascal em ambiente Free Pascal (www.freepascal.org) ou as linguagens C ou C++ em ambiente GCC (<http://gcc.gnu.org>) e submetidos através do mesmo sistema, para avaliação.
10. Antes da prova preliminar, a organização disponibilizará na Internet um concurso fictício para que os concorrentes se possam familiarizar com o sistema.
11. A classificação é atribuída automaticamente pelo sistema em função do número de testes avaliados correctamente pelos programas submetidos pelos concorrentes e será validada por um júri nacional, nomeado pela APDSI, o qual também desempatará os casos em que isso seja necessário, observando a qualidade da programação.
12. Os melhores concorrentes da primeira fase, até um máximo de 30, participarão na segunda fase do concurso, que constitui a prova final.
13. A prova final é individual e é realizada usando o mesmo sistema da prova preliminar, mas presencialmente, em local ou locais a anunciar.
14. A classificação final do concurso, a atribuir pelo júri nacional, é a da prova final, atribuída nos mesmos moldes da prova preliminar.
15. O calendário das provas será publicado no sítio do concurso, assim como o local ou locais das provas presenciais.
16. As fraudes ou tentativas de fraude em qualquer fase do concurso serão punidas com a exclusão do concorrente.
17. Os concorrentes mais bem classificados na prova final, em número que será determinado pelo júri nacional, serão seleccionados para participarem num estágio de formação sobre resolução de problemas de programação em ambiente de concursos, a realizar em Julho ou início de Agosto de 2008, no final da qual serão escolhidos, de entre os elementos desse grupo, em função do seu desempenho durante o estágio, quatro, para participarem, em representação de Portugal, nas Olimpíadas Internacionais de 2008 (<http://www.ioi2008.org/>) que se realizarão no Cairo, Egipto, de 16 a 23 de Agosto de 2008.
18. A resolução dos casos omissos neste regulamento será da competência do júri nacional.
19. As decisões do júri nacional são irrevogáveis e delas não há recurso.

Anexo II - Sítio na Web (<http://www.dcc.fc.up.pt/oni/2008/>)

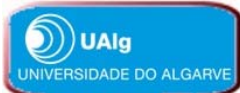
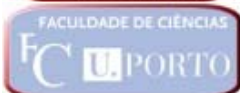
olimpiadas nacionais de informática



20^{as} Olimpíadas Nacionais de Informática - Edição 2008

- > [Início](#)
- > [Notícias](#)
- > [Final 2008](#)
- > [Informação](#)
- > [Regulamento](#)
- > [Problemas](#)
- > [Inscrições](#)
- > [Calendário](#)
- > [Organização](#)
- > [Liações](#)

- > [Treino/Mooshak](#)
- > [Fórum](#)



Patrocínios/Apoios



Bem-vindo ao site oficial das Olimpíadas Nacionais de Informática, um concurso de programação destinado a alunos do Secundário. As Olimpíadas são um concurso de carácter individual e as linguagens de programação permitidas para resolver os problemas propostos são o Pascal, C ou C++.

Os melhores classificados terão a oportunidade de representar Portugal nas [Olimpíadas Internacionais de Informática](#), que este ano serão realizadas de 16 a 23 de Agosto na cidade do Cairo, no Egipto.

Consulta a secção [informação](#) para saberes mais. Podes também descobrir o que se passou nas [edições passadas](#) das Olimpíadas..

Fotos da final

13:48 23/07/2008

Já estão disponíveis fotos da final nacional.

Podes ver algumas imagens do registo das equipas, dos laboratórios de competição, da cerimónia decorrida no anfiteatro e do jantar/convívio.

O álbum inclui fotos individuais dos concorrentes durante a prova e ainda fotos da entrega de prémios para os 4 primeiros lugares.

[Ver álbum de fotos da Final Nacional 2008.](#)

Se alguém tiver mais fotos que gostasse de ver publicadas, por favor entre em contacto conosco.



Classificação da final

12:17 17/05/2008

Realizou-se ontem a grande Final Nacional das ONI'2008 e já está disponível a classificação oficial desse mesma final.

[Classificação oficial da Final Nacional.](#)

O grande vencedor desta 20a edição das Olimpíadas Nacionais de Informática, "o campeão nacional", foi o **Tiago Andrade** (Colégio Internato dos Carvalhos), com uns fantásticos **225** pontos (quase o dobro do segundo classificado), resolvendo dois dos problemas na totalidade e obtendo 25 pontos no terceiro problema. Parabéns a todos os que participaram, mas de modo mais especial aos 8 primeiros, que vão agora ter um estágio de preparação para concursos de programação (de onde sairão os nome dos 4 que irão representar Portugal no Egipto, nas Olimpíadas Internacionais).

Já estão também disponíveis os enunciados dos 3 problemas da final:

- **Problema A:** [Controle de Tráfego Aéreo](#)
- **Problema B:** [Números Fáceis de Escrever](#)
- **Problema C:** [Vigiando a Fronteira](#)

Brevemente serão colocadas aqui algumas fotos da prova.



Final Nacional

Treino para a final 2008

12:31 08/05/2008

Com a final a chegar (é já próximo no dia 16 de Maio) damos-te agora a possibilidade de te treinares, colocando os **12 problemas das últimas 4 finais disponíveis para submissão** (nota que os testes usados no treino não necessariamente iguais aos das finais reais).

Podes registar-te (com um email válido) e começar a submeter. E a inscrição está aberta a todos, mesmo os que não são finalistas.

- [Entrar no servidor de treino](#)

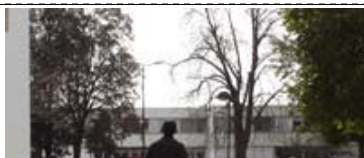


Informações sobre a final

11:13 08/05/2008

A final nacional irá realizar-se no dia **16 de Maio de 2008** (sexta-feira), no [Departamento de Ciência de Computadores](#) (DCC) da [Faculdade de Ciências](#) da [Universidade do Porto](#).

Não deixes de espreitar a [secção de informações sobre essa mesma final](#), que contém dados como o programa, como chegar ao




local da prova e os recursos computacionais (*hardware* e *software*) disponíveis.

É esperado que os participantes da final cheguem entre as **12:00** e **as 12:45**. Podes por isso começar já a planear a tua viagem. Estamos à tua espera para o grande dia!

Aproveita também para conheceres os [patrocinadores](#) que tornam possível a realização das Olimpíadas Nacionais de Informática'2008.



12 notícias em arquivo | [Ver todas](#) | 

Última alteração: **13:48 23/07/2008**

Comentários/Sugestões: [primeiro_at_dcc.fc.up.pt](#)

19 225 visitantes únicos 

Anexo III - Problemas do concurso de Qualificação

Problema A - Matrículas



Em Portugal, as matrículas dos carros começaram por ter o formato XX-NN-NN, onde X era uma das 23 letras do nosso alfabeto entre A e Z (nota que as letras K, W e Y não são permitidas) e N era um dígito entre 0 e 9. Um exemplo de uma matrícula válida é QX-79-84. As matrículas são atribuídas a novos veículos de forma sequencial. Os novos valores na sequência são obtidos primeiro incrementando a parte numérica da matrícula (onde o dígito menos significativo está à direita), e depois a parte alfabética (tal como nos números, a letra menos significativo está à direita). Por exemplo, a seguir à matrícula CC-00-00 vem a matrícula CC-00-01, a seguir a CC-99-99 vem CD-00-00 e a seguir a CZ-99-99 vem a matrícula DA-00-00.

A última matrícula válida neste sistema era o ZZ-99-99, mas rapidamente o número de veículos em circulação ultrapassou o permitido pelo esquema de matrículas e desse modo novos formatos foram adicionados ao sistema. A primeira extensão foi o formato NN-NN-XX. Quando estas matrículas esgotaram (a última foi 99-99-ZZ) foi introduzido o mais recente formato NN-XX-NN. Em todos estes esquemas, a parte dos números é incrementada antes da parte das letras. Por exemplo, a seguir a 00-AA-99 vem a matrícula 01-AA-00.

É possível definir a distância entre duas matrículas observando a ordem em que elas foram sendo atribuídas. Por exemplo, a distância entre QX-79-84 e QX-79-86 é de 2. Já a distância entre ZZ-90-00 e ZZ-80-00 é de 1000, tal como a distância entre ZZ-90-00 e 00-00-AA.

O Problema

Dadas duas matrículas em qualquer um dos três formato possíveis, a tua tarefa é calcular e imprimir a distância entre elas.

Input

O *input*

é constituído por duas linhas, cada uma delas contendo uma única matrícula. As matrículas vêm na forma XX-NN-NN, NN-NN-XX ou NN-XX-NN, onde X é uma letra portuguesa entre 'A' e 'Z' (não podem aparecer 'K', 'W' e 'Y') e N é um dígito entre 0 e 9. As duas matrículas são diferentes uma da outra.

Output

O *output*

é constituído por uma única linha, contendo um número inteiro positivo que indica a distância entre as duas matrículas dadas.

Exemplo de Input 1

ZZ-90-00
00-00-AA

Exemplo de Output 1

1000

Exemplo de Input 2

01-AA-00
00-AA-99

Exemplo de Output 2

1

Exemplo de Input 3

AA-00-00
00-00-AA

Exemplo de Output 3

5290000

Exemplo de Input 4

23-AC-47
41-86-ZZ

Exemplo de Output 4

28161

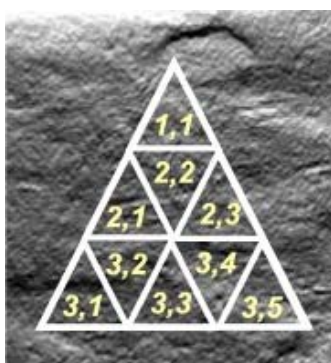
Qualificação para a final das ONI'2008
(17/04 a 19/04 de 2008)

Problema B - Indiana Jones e o Triângulo Perdido

Toda a humanidade quer descobrir o triângulo perdido da Atlântida, um tesouro de valor incalculável. Indiana Jones está à sua procura e descobriu um templo abandonado há milhares de anos que pode conter pistas importantes. Como não podia deixar de ser, todas as informações estão de algum modo relacionadas com triângulos. Por todo o templo Indiana Jones encontrou triângulos dentro de triângulos em imagens como estas:



Indiana notou como os números se organizavam. Começavam de cima para baixo e da esquerda para a direita. Mais à frente no templo, os triângulos continuavam, mas pareciam ter outro formato:



Indiana percebeu facilmente que neste caso o primeiro número indicava a "linha" de triângulos (a contar de cima para baixo) e o segundo número indicava o número do triângulo (a contar da esquerda para a direita).

Um pouco mais à frente uma misteriosa porta surgiu, com milhares de inscrições como $14-??$, $??$ ou $??-29, 17$. Indiana pensou e chegou à conclusão que para a abrir a porta teria de resolver o enigma dos pontos de interrogação. Depois de experimentar um bocado, percebeu que tudo estava relacionado com os misteriosos triângulos que vira atrás! Assim, para abrir a porta, teria de preencher os pontos de interrogação com o resultado de converter entre a notação usada nos dois tipos de triângulos. Por exemplo, ao número 4 corresponde a coordenada 2, 3. De igual modo, à coordenada 3, 5 corresponde o número 9. Indiana reparou que existiam também números inscritos na porta muito maiores do que os dos triângulos que vira, mas raciocinou rapidamente e percebeu que se baseavam no esquema usado nos triângulos. Por exemplo, ao número 10 corresponderia a coordenada 4, 1, ao número 30 corresponderia a coordenada 6, 5 e por aí adiante.

Serás capaz de ajudar Indiana a decifrar todas as inscrições, fazendo as conversões entre os dois tipos de triângulos?

O Problema

Dado um conjunto de inscrições que correspondem a um número ou a uma coordenada, a tua tarefa é converter de uma representação para outra todas as inscrições. Os números deverão ser convertidos em coordenadas e vice-versa, de acordo com o indicado atrás.

Input

Na primeira linha do *input* vem um número inteiro **NUM**, que indica o número de inscrições a considerar ($1 \leq \text{NUM} \leq 50000$).

Seguem-se **NUM**

linhas cada uma contendo uma inscrição, que pode ser um número ou uma coordenada. As linhas que representam um número começam por ter o carácter 'N', seguido de um espaço e depois vem o número correspondente (por exemplo, o número 42 seria indicado como "N 42", numa linha, sem as aspas). As linhas que representam coordenadas começam por ter o carácter 'C' seguido de um espaço e das duas coordenadas correspondentes separadas por um espaço (por exemplo, a coordenada 10,5 seria indicada como "C 10 5"). Os números usados são sempre maiores que zero e menores que 200000000. As coordenadas são sempre entre 1 e 40000 (inclusive).

Output

O *output* é constituído por exactamente **NUM**

linhas, cada uma contendo o resultado da conversão correspondente, sempre no formato $N-C_1, C_2$ (onde N é o número e C_1, C_2 a coordenada). Estas conversões devem vir pela mesma ordem que são pedidas no *input*. Vê o exemplo para perceberes melhor o formato pedido.

Exemplo de Input

```
5
N 4
C 3 5
C 4 1
N 11
N 30
```

Exemplo de Output

```
4-2,3
9-3,5
10-4,1
11-4,2
30-6,5
```

Qualificação para a final das ONI'2008
(17/04 a 19/04 de 2008)

Problema C - O Carteiro Paulo

O Carteiro Paulo e o seu gato preto e branco vão todos os dias de manhã distribuir o correio na sua carrinha vermelha. Todos na cidade acenam à sua passagem.

A carrinha do Paulo tem uma capacidade limitada e por vezes os sacos de correio ultrapassam o limite de carga. Quando isso acontece, o Paulo quer encher o máximo possível a sua carrinha, levando tanto peso quanto possível. No entanto, o Paulo não pode retirar cartas e encomendas de um saco para outro e apenas pode decidir se um determinado saco vai ou não ser transportado na carrinha.



Imagina por exemplo que o Paulo tinha no correio 4 sacos com os pesos de 4Kg, 5Kg, 7Kg e 8Kg. Se a capacidade de carga é de 10Kg de correio, qual é o máximo peso de correio que o Paulo consegue levar numa viagem? Neste caso, o melhor que consegue fazer é levar 9Kg, que correspondem a transportar os dois sacos de 4Kg e 5Kg. E no caso geral? Tens de ajudar o Paulo!

O Problema

Dado um conjunto de sacos de correio e os seus respectivos pesos, bem como o limite de carga da carrinha vermelha do Paulo, a tua tarefa é calcular o máximo peso que o Paulo consegue levar na carrinha, sabendo que um saco de correio não poder ser dividido (ou é transportado na carrinha ou fica na estação de correios).

Input

Na primeira linha do *input* vem uma única linha contendo dois inteiros **S** e **C**, separados por um espaço, sendo que **S** indica o número de sacos de correio e **C** indica a capacidade de carga máxima da carrinha ($1 \leq S \leq 5000$, $1 \leq C \leq 10000$).

Seguem-se exactamente **S** linhas, cada uma contendo um número inteiro contendo o peso **P_i** de um saco de correio ($1 \leq P_i \leq 500$). Os sacos podem vir em qualquer ordem e podem existir vários sacos com o mesmo peso.

Output

O *output*

é constituído uma única linha, indicando o peso máximo que a carrinha consegue transportar, ou seja, qual a maior soma de pesos de um conjunto de sacos de correio que é inferior ou igual à capacidade de carga de carrinha.

Exemplo de Input 1

```
4 10
4
5
7
8
```

Exemplo de Output 1

```
9
```

Exemplo de Input 2

```
6 15
10
2
4
2
1
1
```

Exemplo de Output 2

```
15
```

Qualificação para a final das ONI'2008
(17/04 a 19/04 de 2008)

Anexo IV - Problemas do concurso da Final Nacional

Problema A - Controle de Tráfego Aéreo

Finalmente, o local do novo aeroporto de Lisboa está escolhido: Alcochete. Construído de raiz a pensar no futuro, terá um imenso volume de aviões sempre a descolar e a aterrar. Gerir todo este "engarrafamento" aéreo não é nada fácil. A missão dos controladores de tráfego aéreo é precisamente garantir um fluxo de aviões seguro. São eles que, a partir da torre de controlo, fornecem indicações e autorizações de voo, de acordo com as características da



aeronave e do contexto do momento. Podem pedir aos pilotos para alterar factores como a rota, a altitude ou a velocidade. Como deves imaginar, todo este trabalho não é completamente manual. Programas de computador muito elaborados auxiliam os controladores e garantem a eficácia do sistema.

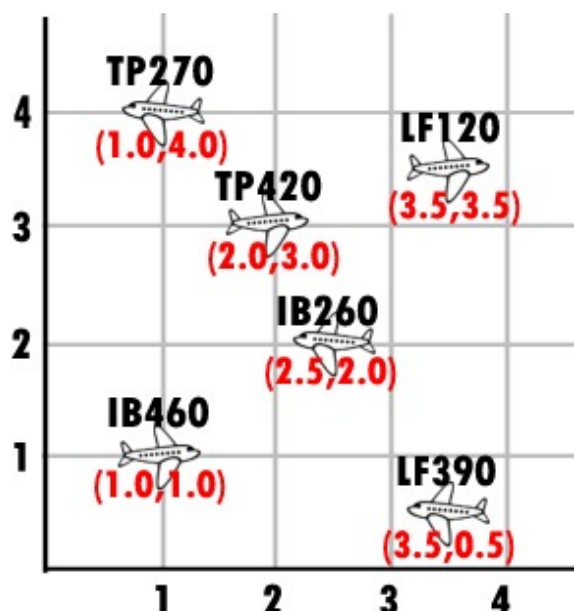
Uma empresa que faz *software*

de controlo aéreo contratou-te a ti para ajudar. Um dos muitos sistemas instalados avisa do perigo de colisão entre dois aviões. Se dois aviões estiverem demasiado próximos, um alarme dispara imediatamente.

Basicamente, a empresa quer que programes uma maneira eficiente de detectar isto, descobrindo no meio de todo o tráfego aéreo quais são os dois aviões mais próximos um do outro. Como simplificação, as posições dos aviões são dadas como em coordenadas (X,Y), em relação a uma determinada origem. A altitude, direcção e velocidade do avião pode ser ignoradas. Se dois aviões tiverem posições (X₁,Y₁) e (X₂,Y₂), a

distância entre eles é dada pela fórmula $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Cada avião é identificado por um código alfanumérico. Um exemplo de um tráfego aéreo seria o que é dado na figura seguinte.



Neste caso, os dois aviões mais próximos seriam o TP420 e o IB260, a uma distância de

$$\sqrt{(2.0 - 2.5)^2 + (3.0 - 2.0)^2}$$

O Problema

Escreve um programa que dadas as localizações (X,Y) de um conjunto de aviões, calcula qual o par de

aviões que está mais perto um do outro.

Input

Na primeira linha de *input* vem um número inteiro A , indicando o número de aviões a considerar ($2 \leq A \leq 50\ 000$).

Seguem-se exactamente A linhas, cada uma delas indicando um avião no formato "**CODIGO** **x** **y**" onde **CODIGO** é uma palavra contendo apenas letras e/ou números (de tamanho máximo 10) e **x** e **y** são as coordenadas da localização do avião ($0 \leq x, y \leq 40\ 000$). É garantido que nunca estarão dois aviões exactamente no mesmo ponto.

Output

O *output*

é constituído por uma única linha contendo os códigos do par de aviões que estão mais próximos um do outro, no formato "**CODIGO1** **CODIGO2**". Deve sempre vir primeiro o código do avião que aparecer também primeiro no *input*.

É garantido que nos casos de teste dados a solução é única, ou seja, existe apenas um par de aviões que está à distância mínima.

Exemplo de Input

```
6
TP270 1.0 4.0
TP420 2.0 3.0
LF120 3.5 3.5
LF390 3.5 0.5
IB460 1.0 1.0
IB260 2.5 2.0
```

Exemplo de Output

```
TP420 IB260
```

Final Nacional das ONI'2008
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
(16 de Maio de 2008)

Problema B - Números Fáceis de Escrever



Sabias que o número de telemóveis existentes no mundo ultrapassa já a metade do número de seres humanos que nele habitam? São mais de 3,3 mil milhões de telemóveis! Claro que isto não quer dizer que metade dos habitantes do nosso planeta tem telemóvel. É que muitas pessoas têm dois, três ou mais. De facto, estima-se que em cerca de 60 países o número de telemóveis existentes é superior à sua própria população. O telemóvel mudou definitivamente a nossas vidas...

A propósito de telemóveis, certamente já reparaste na rapidez com que os jovens escrevem no teclado de um telemóvel. Sejam palavras ou números, tudo é digitado com uma rapidez assombrosa. Concentremo-nos agora nos números. Será que todos se prestam a serem escritos com a mesma rapidez? Comecemos por analisar um teclado clássico de um telemóvel, que pode ser visto na figura seguinte:

```

1 2 3
4 5 6
7 8 9
* 0 #

```

O número 23698, por exemplo, é fácil de escrever, uma vez que os vários dígitos consecutivos são adjacentes no teclado. Vamos concretizar melhor esta definição, dizendo que um número é "fácil de escrever" se um par de dígitos adjacentes no próprio número é constituído pelo mesmo dígito ou por dígitos adjacentes (vertical ou horizontalmente) no teclado clássico do telemóvel. Por exemplo, 1225 é um número "fácil de escrever" uma vez que 1 e 2 são adjacentes no teclado, 2 e 2 são o mesmo dígito e finalmente 2 e 5 são adjacentes. Já 126 não é "fácil de escrever" uma vez que 2 e 6 não são adjacentes pela definição dada.

Se pensarmos na sequência crescente de números "fáceis de escrever", chegamos ao seguinte:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 14, 21, 22, 23, 25, 32, 33, 36, 41, 44, 45, 47, ...

Se associarmos números de ordem à sequência, podemos por exemplo dizer que o 3 é o quarto número da sequência, ou que 36 é o vigésimo número. Por outras palavras o número de ordem 4 na sequência é 3; o número de ordem 20 é 36.

Ordem:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Número:	0,	1,	2,	3,	4,	5,	6,	7,	8,	9,	11,	12,	14,	21,	22,	23,	25,	32,	33,	36,	41,	44,	45,	47,

Dado um número de ordem, serás capaz de descobrir qual o número da sequência a que corresponde?

O Problema

Escreve um programa que dado um número de ordem, calcula qual o número da sequência crescente de números "fáceis de escrever" que tem esse número de ordem. Um número é "fácil de escrever" se todos os seus dígitos consecutivos são iguais ou adjacentes (vertical ou horizontalmente) no teclado do telemóvel.

Input

Na primeira linha de *input* vem um número inteiro c indicando o número de casos a considerar ($1 \leq c \leq 500$).

Seguem-se exactamente c linhas, cada uma delas contendo um número de ordem n da sequência ($1 \leq n \leq 15000$). Estes números podem vir por qualquer ordem.

Output

O *output* é constituído por exactamente c linhas, cada uma no formato " $n: F_n$ ", onde n é o número de ordem e F_n é o n -ésimo número da sequência crescente de números "fáceis de escrever". As linhas devem vir na mesma ordem do *input*.

Exemplo de Input

```
5
4
20
1
23
30
```

Exemplo de Output

```
4: 3
20: 36
1: 0
23: 45
30: 63
```

Final Nacional das ONI'2008
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
(16 de Maio de 2008)

Problema C - Vigiando a Fronteira

A companhia de desenvolvimento de jogos OGPro (Olympiad Game Programmers) está a desenvolver um novo jogo de estratégia em tempo real com tecnologia de inteligência artificial do mais recente que existe. O jogo envolve estratégia militar e muitos pequenos subproblemas precisam de ser resolvidos. Tu fazes parte da equipa de desenvolvimento e estás encarregado de distribuir automaticamente as torres de vigia que irão controlar uma importante linha fronteiriça.



A fronteira pode ser pensada como uma linha recta. Todas as torres de vigia têm um igual alcance pré-determinado que é igual a um número inteiro de quilómetros. Certos pontos da fronteira (os chamados pontos críticos) são mais importantes e por isso mesmo têm de ser mais fortemente vigiados. Estes pontos são indicados por uma única coordenada, indicando a distância em quilómetros do início da fronteira. A figura seguinte exemplifica uma fronteira com 6 pontos críticos a 1, 2, 3, 5, 8 e 10 quilómetros.



O objectivo é colocar as torres de vigia de modo a maximizar o número de pontos críticos vigiados. Por exemplo, se as torres de vigia tiverem um alcance de 1 quilómetro (significa que controlam todos os pontos a distância menor ou igual a 1quilómetro da sua localização), e se apenas tiveres uma torre para colocar, esta deveria ficar na posição 2quilómetro, controlando assim 3 pontos críticos, como podes ver na figura seguinte:



Se, nas mesmas circunstâncias, pudesses colocar 2 torres de vigia, já seria possível controlar 5 pontos críticos, como se pode ver na figura a seguir:



Se fosse possível colocar 3 torres, então todos os 6 pontos críticos podiam ser controlados. Entre várias hipóteses de colocação das torres, esta assegurava esse mesmo controlo:



O Problema

Escreve um programa que dadas as localizações dos pontos críticos na fronteira, o número de torres de vigia disponíveis e o respectivo alcance, calcula o número máximo de pontos críticos que é possível vigiar, colocando as torres em posições de coordenadas inteiras, diferentes entre si (não é possível colocar duas torres no mesmo sítio), ao longo da fronteira.

Input

Na primeira linha de *input* vêm dois números inteiros positivos, T e A , separados por um único espaço. T indica o número de torres de vigia e A o alcance em quilómetros ($1 \leq T \leq 100$, $0 \leq A \leq 50$).

Na segunda linha vem um único número inteiro positivo P indicando o número de pontos críticos ($1 \leq P \leq 1\ 000$ e $T \leq P$). Seguem-se exactamente P linhas, cada uma contendo um número inteiro positivo indicando a localização P_i de um ponto crítico em quilómetros ($0 \leq P_i \leq 5\ 000$). Os pontos críticos podem vir em qualquer ordem no *input*. É garantido que todos os pontos críticos estão em locais diferentes uns dos outros.

Output

O *output*

é constituído por uma única linha, a qual contém o número máximo de pontos críticos que é possível vigiar colocando todas as torres de vigia dadas em locais diferentes. As posições onde as torres podem ser colocadas têm de ser números inteiros, todas as torres têm de ser colocadas e em cada posição não pode haver mais do que uma torre.

Exemplo de Input

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

Exemplo de Output

```
5
```

Anexo V - Classificações do concurso de Qualificação

Classificação Oficial da Qualificação'2008

Classificação obtida pelos concorrentes válidos que submeteram pelo menos uma vez
(entre parenteses encontra-se o número de submissões efectuadas no respectivo problema)

#	Nome	Problemas			Pontos
		A	B	C	
1	ES_JMoniz Pedro_Abreu	100 (4)	100 (3)	100 (7)	300
2	ES_SJTalha Joao_Silvestre	100 (2)	85 (3)	100 (4)	285
2	ES_SJTalha Andre_Ferreira	100 (3)	85 (7)	100 (4)	285
4	CI_Carvalhos Tiago_Andrade	100 (1)	100 (1)	70 (1)	270
5	ES_PBSalgado Antonio_Silva	100 (1)	100 (3)	66 (2)	266
6	ES_DManuell Ricardo_C_Martins	100 (1)	85 (1)	67 (1)	252
7	CI_Carvalhos Jose_Marques	100 (2)	100 (2)	42 (1)	242
8	ES_DAHenriques Ricardo_Goncalves	100 (1)	100 (1)	30 (1)	230
9	EP_Leiria Diogo_Sousa	100 (2)	80 (1)	36 (3)	216
10	EP_Leiria Samuel_Pedro	100 (4)	85 (2)	30 (1)	215
11	ES_MLamas David_Ferreira	76 (3)	100 (7)	36 (1)	212
12	CI_Carvalhos Nuno_Sousa	100 (2)	100 (3)		200
13	ES_SaBandeira Francisco_Huhn	60 (3)	100 (3)	30 (6)	190
14	CI_Carvalhos Diogo_Basto	100 (1)	85 (3)		185
15	ES_DJCorreia Eduardo_Ferreira	84 (3)	60 (2)	30 (2)	174
16	CI_Carvalhos Filipe_Mota	52 (5)	80 (2)	36 (1)	168
17	ES_ALusitano Cristiano_Santos	50 (2)	80 (1)	36 (2)	166
18	ES_ALusitano Ivan_Antunes	50 (13)	80 (2)	30 (8)	160
19	EP_CJCampanha Andre_Oliveira	48 (2)	35 (2)	36 (4)	119
20	ES_DSampaio Ruben_Ramalho		60 (1)	48 (2)	108
20	ES_LCoelho Mario_Rodrigues	24 (1)	60 (13)	24 (1)	108
22	CI_Carvalhos Diogo_Pinho		80 (1)	26 (13)	106
23	ES_ALusitano Daniel_Trigueiros		100 (1)	0 (2)	100
23	ES_DAHenriques Fabio_Ferreira		100 (1)	0 (3)	100
25	ES_DJCCGomes Pedro_Neves	90 (5)			90
25	ES_DAHenriques Vasco_Ribeiro	0 (1)	90 (6)		90
27	ES_APortela Luis_Martins	48 (2)	40 (7)		88
28	CI_Carvalhos Gabriel_Damaso		80 (6)		80
29	AVE_CPaiva Nelson_Luis		60 (1)	18 (1)	78
29	ES_DJCorreia Nuno_Santos		60 (2)	18 (5)	78
31	CI_Carvalhos Ruben_Aguiar		60 (2)		60

32	Inete David_Silva	54 (4)	0 (34)	0 (2)	54
33	ES_Penafiel Fabio_Vieira		45 (1)		45
33	CI_Carvalhos Pedro_Simoes		45 (2)		45
35	AVE_CPaiva Jorge_Fernandes	44 (1)			44
36	ES_DJCCGomes Pedro_Abade	32 (7)			32
37	EP_CJCampanha Arlindo_Silva			30 (1)	30
37	EP_Felgueiras Pedro_Pereira		0 (10)	30 (1)	30
37	ES_Lourinha Filipe_Silva		0 (3)	30 (44)	30
40	ES_DGMachado Joao_Costa	28 (2)			28
41	EB_NMaia Daniel_Silva		25 (3)		25
42	ES_CMaia Diogo_Ferreira			24 (1)	24
42	ES_PRosa Rodrigo_Alho			24 (1)	24
42	ES_ABrotero Frederico_Cerveira			24 (1)	24
42	ES_PAVieira Joao_Patricio			24 (8)	24
46	EP_Felgueiras Joaquim_Carvalho		0 (2)	18 (2)	18
46	EP_Chaves Luis_Dias			18 (8)	18
46	Inete Mario_Jeronimo		0 (8)	18 (2)	18
49	ES_Barcelos Bruno_Carvalho		15 (5)		15
50	ES_SPedroSul Diogo_Santos	0 (1)	0 (2)	12 (1)	12
50	Inete Fabio_Matoso	0 (1)	0 (4)	12 (2)	12
52	ES_PRosa Bruno_Espanha			0 (1)	0
52	ES_MCargaleiro Erickson_Dias	0 (1)			0
52	ES_ALusitano Hugo_Fernandes		0 (1)		0
52	CI_Carvalhos Jorge_Silva		0 (1)		0
52	EP_CJCampanha Abel_Oliveira			0 (1)	0
52	ES_CTejo Eliseu_Cuginguilua		0 (1)		0
52	EB_DMariaII Tiago_AS_Gomes	0 (2)			0
52	EP_CJCampanha Fabio_Rego			0 (2)	0
52	EP_MSeixas Carlos_Cardoso			0 (3)	0
52	ES_MCargaleiro Tiago_MR_Gomes	0 (1)	0 (1)	0 (1)	0
52	ES_SLourenco Joao_Narciso	0 (1)		0 (3)	0
52	ES_Barcelos Tiago_Araujo	0 (2)	0 (2)		0
52	ES_MCargaleiro Rodrigo_Cunha	0 (2)	0 (1)	0 (1)	0
52	ES_SeGuarda Ricardo_Antunes	0 (3)		0 (1)	0
52	ES_MCargaleiro Miguel_Lorete	0 (2)	0 (1)	0 (2)	0
52	ES_MCargaleiro Ricardo_Letras	0 (2)	0 (2)	0 (2)	0
52	ES_MCargaleiro Claudio_Baia	0 (1)	0 (1)	0 (5)	0
52	ES_MCargaleiro Fabio_Santos	0 (5)	0 (1)	0 (2)	0
52	ES_MCargaleiro David_Candido	0 (4)	0 (4)	0 (5)	0
52	Inete Claudio_Faustino	0 (8)	0 (5)	0 (6)	0
52	ES_CMaia Antonio_Teixeira		0 (2)	0 (31)	0

Legenda:

- Problema A: **Matrículas**
- Problema B: **Indiana Jones e o Triângulo Perdido**
- Problema C: **O Carteiro Paulo**

- **Nota 1:** Cada problema tinha uma pontuação máxima de 100 pontos.
- **Nota 2:** Os 30 primeiros classificados qualificam-se para a final nacional.
- **Nota 3:** Algumas submissões de concorrentes foram desqualificadas tendo em conta o ponto 9 do **regulamento**.

Anexo VI - Classificações da Final Nacional

Classificação Oficial da Final Nacional das ONI'2008

Classificação obtida pelos 28 concorrentes que participaram presencialmente na final
(entre parenteses encontra-se o número de submissões efectuadas no problema)

#	Escola	Aluno	Problemas			Pontos
			A	B	C	
1	CI Carvalhos	Tiago Andrade	100 (2)	100 (1)	25 (2)	225
2	ES D. Manuel I	Ricardo Martins	34 (1)	65 (2)	25 (2)	124
3	ES P. B. Salgado	António Silva	40 (2)	51 (3)	30 (1)	121
4	ES J. Moniz	Pedro Abreu	40 (1)	51 (2)	25 (2)	116
5	CI Carvalhos	Nuno Sousa	46 (1)	37 (1)	25 (1)	108
6	EP Leiria	Diogo Sousa	40 (1)	9 (3)	55 (3)	104
7	CI Carvalhos	Filipe Mota	46 (5)	0 (2)	50 (10)	96
8	EP C.J. Campanhã	André Oliveira	46 (2)	37 (1)	10 (3)	93
9	ES Sá da Bandeira	Francisco Huhn	46 (3)	9 (2)	25 (1)	80
10	ES A. Lusitano	Cristiano Santos	40 (4)	9 (5)	30 (2)	79
11	EP Leiria	Samuel Pedro	52 (3)			52
12	ES D. J. Correia	Nuno Santos	40 (8)		10 (5)	50
13	ES DSampaio	Rúben Ramalho	40 (1)			40
13	ES D. J. Correia	Eduardo Ferreira	40 (9)			40
15	CI Carvalhos	José Marques	16 (1)	9 (1)	10 (1)	35
16	ES S. J. Talha	André Ferreira	34 (3)			34
16	ES L. Coelho	Mário Rodrigues	10 (3)	9 (1)	15 (2)	34
18	ES M. Lamas	David Ferreira	28 (2)		0 (3)	28
19	ES D. A. Henriques	Ricardo Gonçalves	10 (1)	0 (2)		10
20	ES D. A. Henriques	Fábio Ferreira	0 (1)			0
20	ES A. Lusitano	Daniel Trigueiros	0 (3)		0 (2)	0
20	ES A. Portels	Luís Martins	0 (4)	0 (1)		0
20	AVE C. Paiva	Nelson Luís			0 (5)	0
20	ES D. J. C. C. Gomes	Pedro Neves	0 (3)	0 (1)	0 (2)	0
20	ES D. A. Henriques	Vasco Ribeiro	0 (6)			0
20	ES A. Lusitano	Ivan Antunes		0 (6)	0 (3)	0
20	CI Carvalhos	Diogo Basto	0 (7)		0 (3)	0
20	ES S.J. Talha	João Silvestre	0 (4)		0 (17)	0

Legenda:

- Problema A: Controle de Tráfego Aéreo
- Problema B: Números Fáceis de Escrever
- Problema C: Vigiando a Fronteira

- **Nota 1:** Cada problema tinha uma pontuação máxima de 100 pontos.
- **Nota 2:** Os 8 primeiros classificados irão agora ter um estágio de preparação específico para concursos de programação, e desse estágio sairão os 4 alunos que irão representar Portugal nas Olimpíadas Internacionais de Informática no Egipto.