

INSTITUTO SUPERIOR TÉCNICO

Introdução aos Algoritmos e Estruturas de Dados

2011/2012

Enunciado do 2^o Projecto

Data de entrega: 11 de Novembro de 2011 às 23h59

1 Introdução

O EuroMilhões é uma lotaria jogada actualmente em nove países europeus: Portugal, Espanha, França, Irlanda, Reino Unido, Áustria, Bélgica, Suíça e Luxemburgo. Embora a probabilidade de acertar na chave do primeiro prémio seja de apenas 1 em 116 531 800, o EuroMilhões é popular e apostas não faltam, apesar da crise.

Para jogar aposta-se cinco números, que podem tomar valores de 1 a 50, e duas estrelas, que podem tomar valores de 1 a 11. Durante o sorteio, são extraídos automaticamente os cinco números e as duas estrelas, usando duas máquinas que contêm bolas numeradas de 1 a 50 e de 1 a 11, respectivamente.

O prémio atribuído a cada jogador é diferente consoante a combinação obtida de números e estrelas. O montante disponível para prémios num dado concurso é distribuído por 13 prémios de acordo com a Tabela 1. Para ganhar o 1^o prémio o jogador tem que acertar os cinco números e as duas estrelas mas para ter direito ao 13^o prémio “basta” acertar dois números.

Neste projecto pretende-se desenvolver um programa que manipule informação relativa a um concurso do EuroMilhões, nomeadamente o valor disponível para prémios, as apostas efectuadas e a chave premiada (5 números + 2 estrelas), e forneça informações sobre os resultados.

Por questões de simplicidade considera-se que não há possibilidade de jackpot e não há apostas múltiplas. Considera-se ainda que as apostas são restritas aos nove países europeus que participam actualmente no EuroMilhões.

Tabela 1: Prémios e montante distribuído

Prémio	Acertos (números + estrelas)	Distribuição
1º Prémio	5 + 2	32.00%
2º Prémio	5 + 1	4.80%
3º Prémio	5 + 0	1.60%
4º Prémio	4 + 2	0.80%
5º Prémio	4 + 1	0.70%
6º Prémio	4 + 0	0.70%
7º Prémio	3 + 2	0.50%
8º Prémio	3 + 1	2.30%
9º Prémio	2 + 2	2.20%
10º Prémio	3 + 0	3.70%
11º Prémio	1 + 2	6.50%
12º Prémio	2 + 1	17.60%
13º Prémio	2 + 0	18.00%

2 Especificação do Programa

O programa recebe como *input* informação acerca de um dado concurso do EuroMilhões: o montante disponível para distribuir em prémios de acordo com a Tabela 1; a chave premiada (5 números + 2 estrelas); o número de apostas; e detalhes de cada aposta (país onde foi registada, código e chave apostada).

Pode receber ainda um número predeterminado de pedidos de informação acerca dos resultados por país. Os nove países são identificados usando os códigos seguintes: POR, ESP, FRA, IRL, GBR, AUT, BEL, SUI e LUX.

O programa deverá apresentar no *output*, para cada um dos 13 prémios possíveis (do 1º prémio ao 13º prémio), a identificação do prémio, o montante disponível para distribuir pelos vencedores do prémio no concurso actual, o número de vencedores do prémio calculado com base nas apostas registadas no concurso e na chave vencedora, e o valor do prémio.

Deverá ainda calcular o número de países distintos com apostas premiadas.

Em resposta a um pedido de informação sobre um dado país, o programa deverá indicar o número de 1º prémios, 2º prémios, ..., 13º prémios distribuídos nesse país.

Finalmente, deverá exibir uma lista das apostas premiadas ordenadas por prémio, e dentro de cada prémio ordenadas pelo código da aposta, isto é, lista de apostas que ganharam o 1º prémio ordenadas por código de aposta, seguidas das apostas que ganharam o 2º prémio ordenadas por código de aposta, etc.

3 Dados de Entrada

O programa deverá ler os dados de entrada a partir do *standard input*. O formato será o seguinte:

- uma linha contendo um real M ($M > 0$) que indica o montante disponível para distribuir em prémios;
- uma linha contendo a chave premiada definida por 7 números inteiros separados por espaços: cinco números no intervalo $[1, 50]$ e duas estrelas no intervalo $[1, 11]$;
- uma linha contendo um inteiro A ($A \geq 0$) que indica o número de apostas registadas;
- uma sequência de A linhas, contendo informação relativa às A apostas registadas, em que cada uma tem o seguinte formato:
 - código do país onde a aposta foi registada (sequência única de três caracteres) seguido de um espaço em branco;
 - código da aposta (sequência única de seis dígitos) seguido de um espaço em branco;
 - 7 números inteiros correspondentes à chave apostada: 5 números no intervalo $[1, 50]$ e duas estrelas no intervalo $[1, 11]$ separados por espaços;
- uma linha com um número inteiro P ($P \geq 0$) que indica o número de pedidos de informação acerca do número de prémios (do 1º prémio ao 13º prémio) que serão distribuídos num dado país tendo em conta as apostas registadas e a chave premiada;
- uma sequência de P linhas com os pedidos de informação e o seguinte formato:
 - um carácter ‘p’ seguido de um espaço em branco;
 - o código do país (sequência única de três caracteres).

Assuma que os dados de entrada para o programa não contêm erros sintácticos, isto é, obedecem sempre ao formato descrito nesta secção.

4 Dados de Saída

O programa deverá escrever no *standard output* a seguinte informação:

- para cada um dos 13 prémios possíveis (do 1º prémio ao 13º prémio), uma linha com a informação seguinte separada por espaços:
 - identificação do prémio (inteiro p no intervalo $[1, 13]$);
 - montante disponível para o prémio p (real com duas casas decimais);
 - número de vencedores do prémio p (inteiro);

- valor do prémio p (real com duas casas decimais);
- número de países (distintos) com apostas premiadas;
- uma sequência de P linhas, onde cada linha corresponde à resposta a um dos P pedidos de informação efectuados no *input* e contém o número de 1º prémios, 2º prémios, ..., 13º prémios distribuídos separados por um espaço. A ordem das linhas do tipo p no *input* é respeitada no *output*.
- lista de apostas premiadas ordenadas por prémio, e dentro de cada prémio ordenadas pelo código da aposta, isto é, lista de apostas que ganharam o 1º prémio ordenadas por código de aposta, seguidas das apostas que ganharam o 2º prémio ordenadas por código de aposta, etc.

5 Exemplo

5.1 Dados de Entrada

Suponha que o ficheiro de entrada, que designaremos por `in.txt`, tem o seguinte conteúdo:

```
100000.00
20 10 5 1 49 1 11
10
ESP 000001 20 10 5 1 49 1 10
IRL 000002 20 10 5 1 49 2 10
POR 000003 20 10 5 1 49 2 11
ESP 000004 20 10 5 1 2 1 11
POR 000005 20 10 5 1 49 1 11
FRA 000006 20 10 5 1 49 2 10
POR 000007 21 11 6 2 48 2 10
POR 000008 21 11 6 2 48 2 10
POR 000009 21 11 6 2 48 2 10
POR 000010 21 11 6 2 48 2 10
5
p POR
p ESP
p IRL
p GBR
p FRA
```

5.2 Dados de Saída

Para o ficheiro de entrada `in.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

```
1 32000.00 1 32000.00
2 4800.00 2 2400.00
3 1600.00 2 800.00
4 800.00 1 800.00
5 700.00 0 0.00
6 700.00 0 0.00
7 500.00 0 0.00
8 2300.00 0 0.00
9 2200.00 0 0.00
10 3700.00 0 0.00
11 6500.00 0 0.00
12 17600.00 0 0.00
13 18000.00 0 0.00
4
1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0
000005
000001
000003
000002
000006
000004
```

6 Compilação do Programa

O compilador a utilizar é o `gcc` com as seguintes opções de compilação: `-ansi -Wall -pedantic`. Para compilar o programa deve executar o seguinte comando:

```
$ gcc -ansi -Wall -pedantic -o proj2 *.c
```

o qual deve ter como resultado a geração do ficheiro executável `proj2`, caso não haja erros de compilação. A execução deste comando não deverá escrever qualquer resultado no terminal. Caso a execução deste comando escreva algum resultado no terminal, considera-se que o programa não compilou com sucesso. Por exemplo, durante a compilação do programa, o compilador não deve escrever mensagens de aviso (*warnings*).

7 Execução do Programa

O programa deve ser executado da forma seguinte:

```
$ ./proj2 < in.txt > out.txt
```

8 Entrega do Projecto

A entrega do projecto deverá respeitar o procedimento seguinte:

- Na página da disciplina aceda ao sistema para entrega de projectos. O sistema será activado uma semana antes da data limite de entrega. Instruções acerca da forma de acesso ao sistema serão oportunamente fornecidas.
- Efectue o upload de um ficheiro arquivo com extensão `.zip` que inclua os ficheiros fonte (`.c`) e cabeçalho (`.h`) que constituem o programa.

Para criar um ficheiro arquivo com a extensão `.zip` deve executar o seguinte comando na directoria onde se encontram os ficheiros com extensão `.c` e `.h`, criados durante o desenvolvimento do projecto:

```
$ zip proj2.zip *.c *.h
```

- Como resultado do processo de upload será informado se a resolução entregue apresenta a resposta esperada num conjunto de casos de teste.
- **O sistema não permite submissões com menos de 10 minutos de intervalo para o mesmo grupo.** Exemplos de casos de teste serão oportunamente fornecidos.
- Data limite de entrega do projecto: **23h59 do dia 11 de Novembro de 2011.** Até à data limite poderá efectuar o número de entregas que desejar, sendo utilizada para efeitos de avaliação a **última** entrega efectuada. Deverá portanto verificar cuidadosamente que a última entrega realizada corresponde à versão do projecto que pretende que seja avaliada. Não serão abertas excepções.

9 Avaliação do Projecto

9.1 Componentes da Avaliação

Na avaliação do projecto serão consideradas as seguintes componentes:

1. A primeira componente avalia o desempenho da funcionalidade do programa realizado. Esta componente é avaliada entre 0 e 16 valores.

2. A segunda componente avalia a qualidade do código entregue, nomeadamente os seguintes aspectos: comentários, indentação, estruturação, modularidade, abstracção, entre outros. Esta componente poderá variar entre -4 valores e +4 valores relativamente à classificação calculada no item anterior e será atribuída na discussão final do projecto.

Durante a discussão final do projecto será averiguada a participação de cada elemento do grupo na realização do projecto, bem como a sua compreensão do trabalho realizado, sendo a respectiva classificação ponderada em conformidade, isto é, elementos do mesmo grupo podem ter classificações diferentes. **Elementos do grupo que se verifique não terem participado na realização do respectivo projecto terão a classificação de 0 (zero) valores.**

9.2 Atribuição Automática da Classificação

A classificação da primeira componente da avaliação do projecto é obtida automaticamente através da execução de um conjunto de testes executados num computador com o sistema operativo **GNU/Linux**. Torna-se portanto essencial que o código compile correctamente e que respeite o formato de entrada e saída dos dados descrito anteriormente. Projectos que não obedeçam ao formato indicado no enunciado serão penalizados na avaliação automática, podendo, no limite, ter 0 (zero) valores se falharem todos os testes. Os testes considerados para efeitos de avaliação podem incluir (ou não) os disponibilizados na página da disciplina, além de um conjunto de testes adicionais. A execução de cada programa em cada teste é limitada na quantidade de memória que pode utilizar, até um máximo de 64 MBytes, e no tempo total disponível para execução, sendo o tempo limite distinto para cada teste.

Note-se que o facto de um projecto passar com sucesso o conjunto de testes disponibilizado na página da disciplina não implica que esse projecto esteja totalmente correcto. Apenas indica que passou alguns testes com sucesso, mas este conjunto de testes não é exaustivo. É da responsabilidade dos alunos garantir que o código produzido está correcto.

Em caso algum será disponibilizada qualquer tipo de informação sobre os casos de teste utilizados pelo sistema de avaliação automática. A totalidade de ficheiros de teste usados na avaliação do projecto serão disponibilizados na página da disciplina após a data de entrega.

9.3 Detecção de Cópias

A avaliação dos projectos inclui a utilização de um sistema para detecção de situações de cópia entre projectos. A submissão de um projecto pressupõe o compromisso de honra que o trabalho incluso foi realizado única e exclusivamente pelos alunos referenciados nos ficheiros submetidos para avaliação. A quebra deste compromisso, ou seja a tentativa de um grupo se apropriar de trabalho de outrem (sejam colegas ou outra pessoa), tem como consequência a reprovação de todos os alunos envolvidos (incluindo quem possibilitar a ocorrência de cópias) à disciplina de IAED.

Toda e qualquer situação de fraude ou facilitação de fraude terá então como consequência a reprovação imediata à disciplina de IAED neste semestre, assim como a comunicação da

ocorrência ao respectivo Coordenador de curso e ao Conselho Pedagógico do IST para sanções adicionais de acordo com as regras aprovadas pela UTL e publicadas em Diário da República.

9.4 Considerações adicionais

Todos os programas são avaliados do modo seguinte:

```
$ ./proj2 < in.txt > out.txt; diff out.txt exp.txt
```

em que o ficheiro `exp.txt` representa o resultado esperado da execução do programa para os dados de entrada definidos pelo ficheiro `in.txt`. A impossibilidade de verificar automaticamente o resultado da execução de um dado programa implica uma penalização de **100%** na componente de avaliação automática. Considera-se que um programa passou um teste com sucesso se o resultado produzido por esse programa for exactamente igual ao resultado esperado, o que significa que o comando `diff` não deverá encontrar diferenças entre o resultado produzido pelo programa submetido e o esperado. Para poder ser avaliado, um projecto deverá compilar correctamente num computador com o sistema operativo **GNU/Linux**, sendo o utilizado o compilador **gcc** da **GNU**. A entrega de código não compilável, ou a não inclusão de qualquer dos ficheiros requeridos, ou a utilização de nomes diferentes para o ficheiro executável conduz a uma classificação de 0 (zero) valores. Não serão aceites quaisquer justificações.