

INSTITUTO SUPERIOR TÉCNICO

Introdução aos Algoritmos e Estruturas de Dados

2013/2014 - 2º Semestre

Enunciado do 1º Projecto

Data de entrega: 25 de Março de 2014 às 23h59

1 Introdução

A União Europeia (UE) é uma união económica e política de **28 Estados-membros** independentes situados principalmente na Europa: Alemanha (DE), Áustria (AT), Bélgica (BE), Bulgária (BG), Chipre (CY), Croácia (HR), Dinamarca (DK), Eslováquia (SK), Eslovénia (SI), Espanha (ES), Estónia (EE), Finlândia (FI), França (FR), Grécia (GR), Hungria (HU), Irlanda (IE), Itália (IT), Letónia (LV), Lituânia (LT), Luxemburgo (LU), Malta (MT), Países Baixos (NL), Polónia (PL), Portugal (PT), Reino Unido (GB), República Checa (CZ), Roménia (RO) e Suécia (SE). As eleições na UE ocorrem a cada cinco anos por sufrágio universal. Os eurodeputados são eleitos para o Parlamento Europeu, que é eleito directamente desde 1979. As próximas eleições serão em Maio e serão eleitos 571 eurodeputados.

A atribuição de lugares para cada Estado-Membro é baseada no princípio da proporcionalidade degressiva, a fim de que, embora o tamanho da população de cada país seja tida em conta, os mais pequenos elegem mais deputados do que seria estritamente justificado pelo tamanho das suas populações. Em 2014 foi acordada a seguinte **divisão proporcional dos 571 eurodeputados (nº de deputados por país)**: Alemanha (96), Áustria (18), Bélgica (21), Bulgária (17), Chipre (6), Croácia (11), Dinamarca (13), Eslováquia (13), Eslovénia, (8), Espanha (54), Estónia (6), Finlândia (13), França (74), Grécia (21), Hungria (21), Irlanda (11), Itália (73), Letónia (8), Lituânia (11), Luxemburgo (6), Malta (6), Países Baixos (26), Polónia (51), Portugal (21), Reino Unido (73), República Checa (21), Roménia (32) e Suécia (20).

Os eurodeputados são eleitos por Partidos Políticos Europeus (PPE). Um partido político a nível europeu é uma organização que segue um programa político, que é composto por indivíduos e partidos nacionais e que está representado em vários Estados-Membros. Sendo assim,

os partidos políticos de cada país pertencem a um PPE e os deputados eleitos a nível nacional por um dado partido serão eleitos pelo PPE a que esse partido pertence.

Nas próximas eleições irão a votos **12 PPEs**: *European People's Party* (EPP), *Party of European Socialists* (PES), *Alliance of Liberals and Democrats for Europe Party* (LDE), *European Green Party* (EGP), *Alliance of European Conservatives and Reformists* (ECR), *European Left Party* (ELP), *Movement for a Europe of Liberties and Democracy* (ELD), *European Democrat Party* (EDP), *European Alliance for Freedom* (EAF), *Alliance of European National Movements* (EMN), *European Christian Political Movement* (ECP), e *Alliance for a Europe of Democracies* (AED).

Relativamente ao sistema de votação, não existe um modelo uniforme de sistema de votação para a eleição dos deputados. Cada Estado-Membro é livre de escolher o seu próprio sistema desde que obedeça a regras pré-definidas. Em Portugal, por exemplo é usado o **método D'Hondt**. O método D'Hondt, também conhecido como método dos quocientes ou método da média mais alta D'Hondt, consiste num algoritmo destinado a calcular a distribuição dos mandatos pelos partidos concorrentes a um órgão de natureza colegial, como a Assembleia da República ou o Parlamento Europeu, em que cada mandato é sucessivamente alocado ao partido cujo número total de votos dividido pelos números inteiros sucessivos, começando na unidade seja maior. O processo de divisão prossegue até se esgotarem todos os mandatos e todas as possibilidades de aparecerem quocientes iguais aos quais ainda caiba um mandato. Em caso de igualdade em qualquer quociente, o mandato é atribuído ao partido menos votado. Isto é, matematicamente o método pode ser representado pela fórmula, $V/(s+1)$ onde V é o número total de votos apurado para o partido e s o número de mandatos já atribuídos em cada iteração do cálculo. O processo repete-se até todos os mandatos estarem atribuídos.

Consideremos o seguinte exemplo prático: O Estado-membro "X" tem direito a eleger 7 deputados e nas suas eleições nacionais concorrem 4 partidos: A, B, C e D (pertencentes a 4 PPEs diferentes, por exemplo, EEP, PES, ELP e EDP, respectivamente). Apurados os votos, a distribuição foi a seguinte: A - 12000 votos; B - 7500 votos; C - 4500 votos; e D - 3000 votos. Da aplicação do método d'Hondt resulta a seguinte série de quocientes:

Partido	Divisor			
	1	2	3	4
A	12000	6000	4000	3000
B	7500	3750	2500	1875
C	4500	2250	1500	1125
D	3000	1500	1000	750

No exemplo acima, os quocientes correspondentes a mandatos, assinalados a negrito, levam à seguinte distribuição de mandatos:

- Partido A (EEP) - 3 eurodeputados, correspondentes aos quocientes 12000 (1º eleito), 6000 (3º eleito) e 4000 (5º eleito). Note-se que apesar do quociente resultante da divisão

por 4 ser 3000, igual aos votos obtidos pelo partido D, o mandato é atribuído ao menos votado, isto é ao Partido D, que assim elege o seu deputado.

- Partido B (PES) - 2 deputados, correspondentes aos quocientes 7500 (2^a eleito) e 3750 (6^a eleito).
- Partido C (ELP) - 1 deputado, correspondente ao quociente 4500 (4^o eleito).
- Partido D (EDP) - 1 deputado, correspondente ao quociente 3000 (7^o e último eleito), beneficiando da regra que em igualdade atribui o lugar à lista menos votada, arrebatando, por um só voto, o lugar ao partido A.

2 Especificação do Programa

Neste projecto pretende-se desenvolver um programa que simule a noite das eleições europeias. O objectivo final do programa é determinar o número de mandatos a atribuir a cada um dos PPEs dadas as votações de cada Estado-membro e o número de mandatos a que cada Estado-membro tem direito. **De forma a simplificar o problema assume-se que todos os Estados-membros da UE utilizam o método D'Hondt. Assume-se também que nos boletins de voto a nível nacional são usados os 12 PPEs, isto é, os cidadãos de cada país votam directamente nos PPEs.**

À medida que o programa recebe resultados parciais das votações em cada Estado-membro deve actualizar, aplicando o método D'Hondt, a lista de mandatos atribuídos a cada PPEs (a nível nacional e europeu). A qualquer momento o programa deve permitir receber contagens de votos e listar os resultados por Estado-membro (nº de mandatos por PPE já atribuídos) e os resultados a nível europeu (nº de mandatos por PPE já atribuídos na UE).

Neste contexto, o programa a desenvolver deverá ter opções (ver Tabela 1) que permitam, num dado instante, e a pedido do utilizador, ler como *input* o número de votos apurado para um dado PPE num dado Estado-membro, e actualizar o número de eurodeputados apurados por PPE usando o método D'Hondt. Deverá também ter opções que permitam num dado instante, e a pedido do utilizador, obter como *output*: informação acerca do número de mandatos apurados por PPE para um dado Estado-membro; e informação sobre o número de mandatos apurados por PPE na UE (soma dos mandatos apurados por Estado-membro).

+	adicionar votos a um PPE num dado país
m	pedir informação sobre deputados já eleitos por PPE num dado Estado-membro
e	pedir informação sobre deputados já eleitos por PPE no total da UE
x	terminar programa

Tabela 1: Opções.

O programa deverá ler os dados de entrada a partir do *standard input*. As opções da Tabela 1 têm a seguinte sintaxe:

- + Estado-membro PPE #votos, onde Estado-membro é um inteiro de 1 a 28 (em que 1 corresponde à Alemanha (DE) e 28 corresponde à Suécia(SE)); PPE é um inteiro de 1 a 12 (em que 1 corresponde ao *European People's Party* (EPP) e 12 corresponde à *Alliance for a Europe of Democracies* (AED)); e #votos é um número inteiro positivo correspondente a uma contagem de votos parcial ou total do PPE no Estado-membro. O número de votos no PPE PPE e o número de mandatos atribuídos a cada PPE pelo método D'Hondt devem ser actualizados.

Por exemplo, + 1 1 1000 reporta 1000 votos no EPP na Alemanha. Da mesma forma que, + 28 13 2000 reporta 2000 votos na AED na Suécia.

O carácter + e os inteiros Estado-membro, PPE e #votos são separados por um espaço.

- m Estado-membro, onde Estado-membro é um inteiro de 1 a 28 (em que 1 corresponde à Alemanha (DE) e 28 corresponde à Suécia(SE)). Ver Secção 3 para detalhes sobre o *output*.

O carácter m e o inteiro Estado-membro são separados por um espaço.

- e (Ver Secção 3 para detalhes sobre o *output*.)
- x (Termina o programa)

Assuma que os dados de entrada para o programa não contêm erros sintácticos, isto é, obedecem sempre ao formato descrito nesta secção. Neste contexto, não precisa, por exemplo, de validar os valores dos códigos inseridos.

3 Dados de Saída

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

- quando a opção for m Estado-membro, escrever o seguinte:
 - uma linha com 3 espaços seguidos dos nomes dos 12 PPEs separados por espaços;
 - uma linha com o código do Estado-membro (2 caracteres) seguido de um espaço e do número de mandatos já apurados para cada um dos 12 PPE no Estado-membro (após a aplicação do método de D'Hondt tendo em conta que o PPE PPE tem mais #votos votos). O número de mandatos são separados por espaços. Quando um PPE não tem mandatos aparece 0. Usem o formato de escrita %3d para escrever os inteiros.

- quando a opção for e, escrever o seguinte:
 - uma linha com 3 espaços seguidos dos nomes dos 12 PPEs separados por espaços;
 - uma sequência de 28 linhas, uma por Estado-membro, onde a i -ésima coluna tem o número de mandatos já apurados para cada um dos 12 PPEs no Estado-membro i , separados por espaços. Quando um PPE não tem votos aparece 0. Usem o formato de escrita `%3d` para escrever os inteiros.
 - uma linha com a string “TE” seguida do número total de mandatos já apurados para cada um dos 12 PPE em todos os estados membros, separados por espaços. Quando um PPE não tem mandatos aparece 0. “TE” indica “Total Europa”.

A opções `+` e `x` actualizam a informação no programa e saem do programa, respectivamente, pelo que não produzem qualquer output.

4 Exemplo 1

4.1 Dados de Entrada

Considere que o ficheiro de entrada, que designaremos por `in1.txt`, tem o seguinte conteúdo:

```
+ 5 1 12000
+ 5 2 7500
+ 5 3 4500
+ 5 4 3000
m 5
e
x
```

4.2 Dados de Saída

Para o ficheiro de entrada `in1.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
CY	3	2	1	0	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
DE	0	0	0	0	0	0	0	0	0	0	0	0
AT	0	0	0	0	0	0	0	0	0	0	0	0
BE	0	0	0	0	0	0	0	0	0	0	0	0
BG	0	0	0	0	0	0	0	0	0	0	0	0
CY	3	2	1	0	0	0	0	0	0	0	0	0
HR	0	0	0	0	0	0	0	0	0	0	0	0

DK	0	0	0	0	0	0	0	0	0	0	0	0
SK	0	0	0	0	0	0	0	0	0	0	0	0
SI	0	0	0	0	0	0	0	0	0	0	0	0
ES	0	0	0	0	0	0	0	0	0	0	0	0
EE	0	0	0	0	0	0	0	0	0	0	0	0
FI	0	0	0	0	0	0	0	0	0	0	0	0
FR	0	0	0	0	0	0	0	0	0	0	0	0
GR	0	0	0	0	0	0	0	0	0	0	0	0
HU	0	0	0	0	0	0	0	0	0	0	0	0
IE	0	0	0	0	0	0	0	0	0	0	0	0
IT	0	0	0	0	0	0	0	0	0	0	0	0
LV	0	0	0	0	0	0	0	0	0	0	0	0
LT	0	0	0	0	0	0	0	0	0	0	0	0
LU	0	0	0	0	0	0	0	0	0	0	0	0
MT	0	0	0	0	0	0	0	0	0	0	0	0
NL	0	0	0	0	0	0	0	0	0	0	0	0
PL	0	0	0	0	0	0	0	0	0	0	0	0
PT	0	0	0	0	0	0	0	0	0	0	0	0
GB	0	0	0	0	0	0	0	0	0	0	0	0
CZ	0	0	0	0	0	0	0	0	0	0	0	0
RO	0	0	0	0	0	0	0	0	0	0	0	0
SE	0	0	0	0	0	0	0	0	0	0	0	0
TE	3	2	1	0	0	0	0	0	0	0	0	0

5 Exemplo 2

5.1 Dados de Entrada

Considere agora o ficheiro de entrada `in2.txt` com o seguinte conteúdo:

```
+ 2 1 12000
+ 2 2 7500
+ 2 3 4500
+ 2 4 3000
m 2
e
x
```

5.2 Dados de Saída

Para o ficheiro de entrada `in2.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	8	5	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
DE	0	0	0	0	0	0	0	0	0	0	0	0
AT	8	5	3	2	0	0	0	0	0	0	0	0
BE	0	0	0	0	0	0	0	0	0	0	0	0
BG	0	0	0	0	0	0	0	0	0	0	0	0
CY	0	0	0	0	0	0	0	0	0	0	0	0
HR	0	0	0	0	0	0	0	0	0	0	0	0
DK	0	0	0	0	0	0	0	0	0	0	0	0
SK	0	0	0	0	0	0	0	0	0	0	0	0
SI	0	0	0	0	0	0	0	0	0	0	0	0
ES	0	0	0	0	0	0	0	0	0	0	0	0
EE	0	0	0	0	0	0	0	0	0	0	0	0
FI	0	0	0	0	0	0	0	0	0	0	0	0
FR	0	0	0	0	0	0	0	0	0	0	0	0
GR	0	0	0	0	0	0	0	0	0	0	0	0
HU	0	0	0	0	0	0	0	0	0	0	0	0
IE	0	0	0	0	0	0	0	0	0	0	0	0
IT	0	0	0	0	0	0	0	0	0	0	0	0
LV	0	0	0	0	0	0	0	0	0	0	0	0
LT	0	0	0	0	0	0	0	0	0	0	0	0
LU	0	0	0	0	0	0	0	0	0	0	0	0
MT	0	0	0	0	0	0	0	0	0	0	0	0
NL	0	0	0	0	0	0	0	0	0	0	0	0
PL	0	0	0	0	0	0	0	0	0	0	0	0
PT	0	0	0	0	0	0	0	0	0	0	0	0
GB	0	0	0	0	0	0	0	0	0	0	0	0
CZ	0	0	0	0	0	0	0	0	0	0	0	0
RO	0	0	0	0	0	0	0	0	0	0	0	0
SE	0	0	0	0	0	0	0	0	0	0	0	0
TE	8	5	3	2	0	0	0	0	0	0	0	0

6 Exemplo 3

6.1 Dados de Entrada

Considere agora o ficheiro de entrada `in3.txt` com o seguinte conteúdo:

```
+ 2 1 12000
+ 2 2 7500
```

```

+ 2 3 4500
+ 2 4 3000
m 2
+ 2 1 1500
m 2
+ 2 1 1500
m 2
x

```

6.2 Dados de Saída

Para o ficheiro de entrada `in3.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	8	5	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	8	5	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	9	4	3	2	0	0	0	0	0	0	0	0

7 Exemplo 4

7.1 Dados de Entrada

Considere agora o ficheiro de entrada `in4.txt` com o seguinte conteúdo:

```

+ 2 1 12000
+ 2 2 7500
+ 2 3 4500
+ 2 4 3000
m 2
+ 2 1 1500
m 2
+ 2 1 1500
m 2
+ 5 1 12000
+ 5 2 7500
+ 5 3 4500
+ 5 4 3000
m 5
e
x

```


7.2 Dados de Saída

Para o ficheiro de entrada `in4.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	8	5	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	8	5	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
AT	9	4	3	2	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
CY	3	2	1	0	0	0	0	0	0	0	0	0
	EPP	PES	LDE	EGP	ECR	ELP	ELD	EDP	EAF	EMN	ECP	AED
DE	0	0	0	0	0	0	0	0	0	0	0	0
AT	9	4	3	2	0	0	0	0	0	0	0	0
BE	0	0	0	0	0	0	0	0	0	0	0	0
BG	0	0	0	0	0	0	0	0	0	0	0	0
CY	3	2	1	0	0	0	0	0	0	0	0	0
HR	0	0	0	0	0	0	0	0	0	0	0	0
DK	0	0	0	0	0	0	0	0	0	0	0	0
SK	0	0	0	0	0	0	0	0	0	0	0	0
SI	0	0	0	0	0	0	0	0	0	0	0	0
ES	0	0	0	0	0	0	0	0	0	0	0	0
EE	0	0	0	0	0	0	0	0	0	0	0	0
FI	0	0	0	0	0	0	0	0	0	0	0	0
FR	0	0	0	0	0	0	0	0	0	0	0	0
GR	0	0	0	0	0	0	0	0	0	0	0	0
HU	0	0	0	0	0	0	0	0	0	0	0	0
IE	0	0	0	0	0	0	0	0	0	0	0	0
IT	0	0	0	0	0	0	0	0	0	0	0	0
LV	0	0	0	0	0	0	0	0	0	0	0	0
LT	0	0	0	0	0	0	0	0	0	0	0	0
LU	0	0	0	0	0	0	0	0	0	0	0	0
MT	0	0	0	0	0	0	0	0	0	0	0	0
NL	0	0	0	0	0	0	0	0	0	0	0	0
PL	0	0	0	0	0	0	0	0	0	0	0	0
PT	0	0	0	0	0	0	0	0	0	0	0	0
GB	0	0	0	0	0	0	0	0	0	0	0	0
CZ	0	0	0	0	0	0	0	0	0	0	0	0
RO	0	0	0	0	0	0	0	0	0	0	0	0
SE	0	0	0	0	0	0	0	0	0	0	0	0
TE	12	6	4	2	0	0	0	0	0	0	0	0

8 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 proj1 *.c
```

o qual deve ter como resultado a geração do ficheiro executável `proj1`, 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*).

9 Execução do Programa

O programa deve ser executado da forma seguinte:

```
$ ./proj1 < in.txt > out.txt
```

10 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` (se for o caso), criados durante o desenvolvimento do projecto:

```
$ zip proj1.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 30 minutos de intervalo para o mesmo grupo.** Exemplos de casos de teste serão oportunamente fornecidos.

- Data limite de entrega do projecto: **23h59 do dia 25 de Março de 2014**. 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.

11 Avaliação do Projecto

11.1 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.**

11.2 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 32 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.

11.3 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.

11.4 Considerações adicionais

Todos os programas são avaliados do modo seguinte:

```
$ ./proj1 < 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.