



Os Jogos Olímpicos de 2016 realizar-se-ão em São Sebastião do Rio de Janeiro/RJ de 5 a 21/08/2016. A aquisição dos **ingressos** é feita em várias fases:

1. A pessoa candidata-se aos **eventos** a que deseja assistir, informando, também o número de ingressos que deseja para cada **evento**.<sup>1</sup>
2. A atribuição dos ingressos é feita por sorteio. Se a pessoa for sorteada, ser-lhe-á enviado um boleto para que pague pelos ingressos. Se a pessoa não for sorteada, receberá uma carta informando com pesar a notícia de sua pouca sorte.<sup>2</sup>
3. Os espectadores que receberam o boleto deverão quitá-lo até a data definida (1º prazo).<sup>3</sup>
4. Caso o sorteado não quite o boleto, terá uma segunda chance de fazê-lo, porém, a um preço maior. Para tal, o próprio boleto já contém a informação sobre o segundo prazo e a multa a ser aplicada.
5. Finalmente, os ingressos (para cada um dos eventos) referentes ao boleto quitado são enviados pelo correio.<sup>4</sup>

Os alunos de Estrutura de Dados da UEL venceram uma concorrência internacional para implementar um programa que gerencie o processo descrito acima, conforme detalhado a seguir.

## Eventos

A Figura 1 mostra um trecho do calendário de competições da Olimpíada com algumas anotações em vermelho. Este calendário indica, por exemplo, que haverá **um evento** de Ciclismo de Estrada, no Forte de Copacabana, no dia 6/8/16. O calendário também indica que haverá **4 eventos** de futebol no Estádio Olímpico, nos dias 3,4,6 e 7/8. Em resumo, **cada "bolinha"** na tabela mostrada na figura corresponde a exatamente a **um evento**; além disso, **cada ingresso** corresponde a **um evento**.

## Entrada de Dados

O Comitê Olímpico Local enviará vários arquivos ao Centro de Processamento de Dados DC/LAB2 para que sejam processados e sejam impressos os boletos, ingressos e cartas. Os arquivos enviados são descritos a seguir.

### Lista de eventos

A figura abaixo mostra um exemplo de um arquivo (eventos0001.txt) contendo uma lista com três eventos. Cada evento é caracterizado por 5 informações: o código do evento (por exemplo, ba001), o esporte relacionado (por exemplo, basquetebol), o local onde será realizado (ex,

1 veja arquivo <nome-arq>.si

2 veja arquivo <nome-arq>.ci

3 veja arquivo <nome-arq>.bol

4 arquivos <nome-arq>.ing e \*.bp

Arena da Juventude), a data que ocorrerá (ex, 06/08/2016) e o valor do ingresso (ex, R\$ 50,00) sem centavos. Cada uma destas informações está em uma linha do arquivo. Os eventos são separados por uma linha que contém um caractere =.

```
ba001
basquetebol
Arena da Juventude
06/08/2016
50
=
ba002
basquetebol
Arena da Juventude
07/08/2016
50
=
ce001
ciclismo de estrada (prova de estrada)
Forte de Copacabana
06/08/2016
20
#
```

**eventos0001.txt**

### Solicitação de Ingressos

A figura abaixo mostra um exemplo do arquivo de solicitação de ingressos (<nome-arq>.si). Este exemplo contém as solicitações de ingressos de três pessoas. A organização deste arquivo é semelhante ao anterior. Para cada pessoa são informados os seguintes dados: o nome da pessoa (ex, José da Silva), seu CPF (ex, 999.888.777-66), seu endereço para o envio de correspondências, e os eventos para os quais a pessoa deseja ingressos. Cada evento solicitado é anotado em uma linha, no seguinte formato:

código:qtdade-ingressos

Assim, por exemplo, ba002:4 significa que o solicitante deseja 4 ingressos para o evento cujo código é ba002. Note que existe um evento com este código na lista de eventos.

```
José da Silva
999.888.777-66
Rua dos Coqueiros, 150
ba002:4
ce001:3
=
```

```
Maria da Silva
111.222.333-44
Rua das Laranjeiras, 188 ap 15
ba001:1
ba002:5
ce001:4
=
João Oliveira
555.666.777.88
Rua das Oliveiras, 171
ce001:1
#
```

**<nome-arq>.si**

## **Arquivos Intermediários**

O arquivo descrito abaixo é produzido na fase inicial execução do programa e utilizado como entrada na fase final da execução do programa. Lembre-se que provavelmente haverá mais solicitações do que ingressos disponíveis. O programa deverá sortear as solicitações que serão contempladas e "enviar" os respectivos boletos. A seguir, programa deverá produzir aleatoriamente os arquivos dos boletos pagos no primeiro e no segundo prazo e processá-los, a fim de produzir os ingressos.

### Boletos Pagos

O exemplo abaixo mostra um arquivo (<nome>.bp) contendo o CPF dos solicitantes que efetivamente pagaram os boletos.

```
999.888.777-66
111.222.333-44
```

**<nome>.bp**

5-21 Agosto

# Calendário de competição

			3	4	5	6	7
			Quarta	Quinta	Sexta	Sábado	Domingo
<b>AT</b>	Atletismo	Estádio Olímpico					
<b>AM</b>	Atletismo - Maratona	Sambódromo					
<b>AA</b>	Atletismo - Marcha atlética	A confirmar					
<b>BD</b>	Badminton	Riocentro - Pavilhão 4					
<b>BA</b>	Basquetebol	Arena da Juventude / Arena Carioca 1				• ba001	• ba002
<b>...</b>	Boxe	Riocentro - Pavilhão 6				•	•
	Canoagem slalom	Estádio de Canoagem Slalom					•
	Canoagem velocidade	Estádio da Lagoa					
	Ciclismo - BMX	Centro Olímpico de BMX					
	Ciclismo de estrada (contrarrelógio)	Pontal					
	Ciclismo de estrada (prova de estrada)	Forte de Copacabana				• ce001	• ce002
	Ciclismo de pista	Velódromo Olímpico do Rio					
	Ciclismo - Mountain bike	Centro de Mountain Bike					
	Esgrima	Arena Carioca 3				•	•
	Futebol	Estádio Mineirão	•			•	
		Estádio Mané Garrincha		•			•
		Arena Amazônia		•			•
		Maracanã					
		Estádio Olímpico	•	•		•	•
		Arena Fonte Nova		•			•
	A confirmar	A confirmar	•			•	
	Ginástica artística	Arena Olímpica do Rio				•	•
	Ginástica de trampolim	Arena Olímpica do Rio					
	Ginástica rítmica	Arena Olímpica do Rio					

Figura 1: Calendário das competições

## Saida de Dados

A execução do programa produzirá como resultado alguns arquivos-texto descritos a seguir.

## Boletos

A impressão dos boletos será simulada em um arquivo, como o exemplo abaixo.

Data do vencimento: 21/08/15

Eventos solicitados:

4 (ba002)basquetebol/Arena da Juventude 07/08/2016

3 (ce001)ciclismo de estrada (prova de estrada)/Forte de Copacabana 06/08/2016

Sacado:

José da Silva

999.888.777-66

Rua dos Coqueiros, 150

VALOR TOTAL: R\$ 260,00

ATENÇÃO: caso este boleto não seja pago até a data de vencimento, poderá ser pago até 05/09/15 com multa de 30%

.....

Data do vencimento: 21/08/15

Eventos solicitados:

1 (ba001)basquetebol/Arena da Juventude 06/08/2016

5 (ba002)basquetebol/Arena da Juventude 07/08/2016

4 (ce001)ciclismo de estrada (prova de estrada)/Forte de Copacabana 06/08/2016

Sacado:

Maria da Silva

111.222.333-44

Rua das Laranjeiras, 188 ap 15

VALOR TOTAL: R\$ 380,00

ATENÇÃO: caso este boleto não seja pago até a data de vencimento, poderá ser pago até 05/09/15 com multa de 30%

<nome-arq>.bol

### Ingressos

A impressão dos ingressos também será simulada em um arquivo-texto, como no exemplo abaixo:

RIO 2016 – BASQUETEBOL

LOCAL: Arena da Juventude

DATA: 07/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0001

Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento

no site <https://sou.azarado.cob.br/perdi/meu/ingresso>

VALOR DO INGRESSO: R\$ 50,00

.....

RIO 2016 – BASQUETEBOL  
LOCAL: Arena da Juventude  
DATA: 07/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0002  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>

.....  
RIO 2016 – BASQUETEBOL  
LOCAL: Arena da Juventude  
DATA: 07/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0003  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>  
VALOR DO INGRESSO: R\$ 50,00

.....  
RIO 2016 – BASQUETEBOL  
LOCAL: Arena da Juventude  
DATA: 07/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0004  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>  
VALOR DO INGRESSO: R\$ 50,00

.....  
RIO 2016 – CICLISMO DE ESTRADA (PROVA DE ESTRADA)  
LOCAL: Forte de Copacabana  
DATA: 06/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0005  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>  
VALOR DO INGRESSO: R\$ 20,00  
.....

RIO 2016 – CICLISMO DE ESTRADA (PROVA DE ESTRADA)  
LOCAL: Forte de Copacabana  
DATA: 06/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0006  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>  
VALOR DO INGRESSO: R\$ 20,00

.....  
RIO 2016 – CICLISMO DE ESTRADA (PROVA DE ESTRADA)  
LOCAL: Forte de Copacabana  
DATA: 06/08/2016

COMPRADOR: José da Silva 999.888.777-66

Número de série: 0000.0000.0000.0007  
Em caso de extravio, solicitar segunda via do ingresso até o dia anterior ao do evento  
no site <https://sou.azarado.cob.br/perdi/meu/ingresso>  
VALOR DO INGRESSO: R\$ 20,00

<nome-arq>.ing

### Carta de Indeferimento

O arquivo abaixo exemplifica o arquivo contendo as cartas enviadas para os solicitantes de ingressos que não foram sorteados.

João Oliveira  
Rua das Oliveiras, 171

Com pesar informamos que não será possível fornecer-lhe o(s) ingresso(s) solicitado(s). Desejamos-lhe melhor sorte na próxima Olimpíada e bom divertimento ao assistir os jogos pela televisão.

Atenciosamente,

Comitê Olímpico Local

<nome-arq>.ci

### **A Implementação**

Cada aluno deverá implementar o TAD (Lista) como está especificado nas transparências da aula. O aluno fará as quatro implementações (módulos) deste TAD (alocação {estática, dinâmica} X encadeamento {simples, duplo}).

Os módulos deverão estar em arquivos diferentes do programa principal. Os arquivos executáveis deverão ter os nomes **riose**, **riosd**, **riode**, **riodd**. O primeiro usará a implementação da lista com encadeamento simples e alocação estática; o segundo, a implementação com

encadeamento simples e alocação dinâmica; o terceiro, encadeamento duplo+alocação estática; por fim, encadeamento duplo+alocação dinâmica. Eles serão invocados por um script e deverão aceitar os parâmetros:

```
-e <arq-eventos> -si <nome1>.si -b1 <nome2>.bp -b2 <nome3>.bp  
-o <dir-saida> -rsi <i> <f> -rig <i> <f>
```

A opção `-e` informa o arquivo de eventos; a opção `-si`, o arquivo com as solicitações de ingresso; `-b1`, os boletos pagos no primeiro prazo; `-b2`, os boletos pagos no segundo prazo; por fim, os arquivos de saída serão gravados no diretório informado na opção `-o`. Note que todos os arquivos podem incluir path absoluto ou relativo. Importante, os paths dos arquivos informados nas opções `-b1` e `-b2` são sempre relativos ao diretório de saída informado na opção `-o`.

As opções `-rsi` e `-rig` são opções para teste do programa. Elas possuem dois parâmetros: um número de início e um número de fim. Elas removem elementos da lista de solicitação de ingressos e da lista de ingressos, respectivamente, a partir do *i-ésimo* elemento (inclusive) até o *f-ésimo* elemento (inclusive). As remoções indicadas pela opção `-rsi` devem ser feitas após o processamento do arquivo indicado na opção `-si`, mas antes da emissão dos boletos. As remoções indicadas na opção `-rig` devem ser feitas após o processamento dos arquivos indicados nas opções `-b1` e `-b2`, após o "cálculo" dos ingressos, mas antes que sejam emitidos.

Os arquivos de saídas são nomeados a partir do nome do arquivo da opção `-si`, trocando as extensões consistentemente.

## A entrega

O aluno deverá entregar:

- Um CD com os arquivos-fonte organizado conforme descrito à frente. Anote o seu nome no CD. O CD não será devolvido.
- Um relatório impresso com os TADs usados (exceto o TAD das transparências)



## PROCESSO DE COMPILAÇÃO E TESTES DO TRABALHO

### Organização do CD a ser entregue

A organização do CD a ser entregue pelo aluno deve ser a seguinte:

#### [RAIZ DO CD]

LEIA-ME.txt

*colocar matrícula e o nome do aluno. Atenção: O número da matrícula de estar no início da primeira linha do arquivo. Só colocar os números; não colocar qualquer pontuação*

/src

*(arquivos-fonte)*

Makefile

*deve ter target para a geração do arquivo objeto de cada módulo e os targets **riose**, **riosd**, **riode**, **riodd** que produzirão, respectivamente, os executáveis de mesmo nome, conforme descrito anteriormente*

\*.h e \*.c

### Organização do diretório para a compilação e correção dos trabalhos (no computador do professor):

#### [HOME DIR]

\*.py

*scripts para compilar e executar*

\testes

*diretório contendo os arquivos de testes*

\e

*arquivos de eventos*

\i

*arquivos de solicitações de ingresso (\*.si)*

\alunos

*(contém um diretório para cada aluno)*

\m0001

*(diretório para o aluno cuja matrícula é 0001. O conteúdo do CD é copiado para dentro deste diretório)*

*outros subdiretórios para os arquivos de saída informados na opção -o*

Os passos para correção serão os seguintes:

1. O conteúdo do CD será copiado para um subdiretório específico do aluno. O nome deste subdiretório é o número da matrícula do aluno (lido do arquivo LEIA-ME.txt)
2. O makefile provido pelo aluno será usado para compilar os módulos e produzir os executáveis riose, riosd, riode e riodd. Os executáveis devem ser produzidos no mesmo diretório dos arquivos fontes. O professor usará o GNU Make 3.81. Serão executadas (a partir dos scripts) os seguintes comandos:
  - make riose
  - make riosd
  - make riode
  - make riodd
3. Os programas serão executados automaticamente (por meio de um script) e o resultado produzido será inspecionado visualmente pelo professor. Exemplo de uma possível invocação do programa riose:

- `riose -e .\testes\eventos01.txt`  
`-si .\testes\i\ig001.si`  
`-b1 .\bol001-1.bp`  
`-b2 .\bol001-2.bp`  
`-o alunos\m0001`
- `riose -e .\testes\eventos01.txt`  
`-si .\testes\i\ig002.si`  
`-b1 .\bol002-1.bp`  
`-b2 .\bol002-2.bp`  
`-o alunos\m0001 -rsi 20 30 -rig 60 76`

O compilador usado pelo professor será o gcc 4.7.2

## EXEMPLO DE UM PROGRAMA QUE TRATA OS PARÂMETROS RECEBIDOS

Arquivo: args.c

```
/*
  invocacao:
      args -d <path> -f <nomearq>

  cria arquivo <path>/<nomearq>
  grava no arquivo criado a string "Hello world!"
*/
int main(int argc, char *argv[])
{
    int i=1;
    char *nomeArq=NULL;
    char *path=NULL;
    char *arq=NULL;
    FILE *f;

    while(i<argc) {
        if (strcmp("-f",argv[i])==0) {
            /* proximo argumento deve ser nome do arquivos */
            /* se não existir próximo parametro: ERRO! */
            i++;
            nomeArq = (char *)malloc((strlen(argv[i])+1)*sizeof(char));
            strcpy(nomeArq,argv[i]);
        }
        else if (strcmp("-d",argv[i])==0) {
            /* proximo argumento deve ser nome do arquivos */
            /* se não existir próximo parametro: ERRO! */
            i++;
            path = (char *)malloc((strlen(argv[i])+1)*sizeof(char));
            strcpy(path,argv[i]);
        }
        i++;
    }

    arq = (char *)malloc((strlen(nomeArq)+strlen(path)+2)*sizeof(char));
    sprintf(arq,"%s/%s",path,nomeArq);

    f = fopen(arq,"w");
    fprintf(f,"Hello world!\n");
    fclose(f);
    return 0;
}
```