

# MAC0438 – Programação concorrente – 1s2013

## EP1

Data de Entrega: 01/04/2012

Prof. Daniel Macêdo Batista

### 1 Problema

O Campeonato Mundial de Iron Man <sup>1</sup> é uma das competições mais difíceis de todas as modalidades de esporte do mundo. Nele diversos triatletas de diversos lugares do mundo se reúnem no Havaí em busca da medalha de ouro. Vence aquele que conseguir cruzar a linha de chegada em primeiro após ter percorrido 3.8Km de natação, 180Km de ciclismo e 42Km de corrida.

A sua tarefa neste EP é simular uma edição do Iron Man, fornecendo ao término da simulação a lista com a classificação dos atletas da elite masculina, dos atletas amadores masculinos, das atletas da elite feminina e das atletas amadoras femininas. Essa classificação deve informar para cada colocação:

- Nome do atleta
- Tempo total de prova
- Tempo gasto na etapa de natação
- Tempo gasto na T1 (transição da natação para o ciclismo)
- Tempo gasto na etapa de ciclismo
- Tempo gasto na T2 (transição do ciclismo para a corrida)
- Tempo gasto na etapa de corrida

Esta página da wikipedia mostra um exemplo de como a saída deve ser exibida: [http://en.wikipedia.org/wiki/2012\\_Ironman\\_World\\_Championship](http://en.wikipedia.org/wiki/2012_Ironman_World_Championship)

Você também precisa fornecer essas listas parcialmente (apenas os 3 primeiros de cada grupo) durante a execução da simulação informando os tempos totais e separados para cada classe de atletas de 30 em 30 minutos.

A simulação deve considerar que  $m$  atletas estão competindo.

Cada modalidade do triathlon possui algumas particularidades e restrições:

**Na natação:**

- Os homens profissionais conseguem fazer cada 100m entre 1min15seg e 1min40seg

---

<sup>1</sup><http://www.ironman.com/triathlon/events/ironman/world-championship.aspx>

- As mulheres profissionais conseguem fazer cada 100m entre 1min30seg e 2min
- Os homens amadores conseguem fazer cada 100m entre 2min e 4min
- As mulheres amadoras conseguem fazer cada 100m entre 2min30seg e 5min

#### **No ciclismo:**

No máximo 3 atletas podem estar lado a lado e atletas que estejam atrás de outros precisam manter uma distância de no mínimo 100 metros, porque é proibido tirar proveito do vácuo na prova.

- Os homens profissionais conseguem pedalar nos trechos planos entre 50Km/h e 55Km/h, nas subidas entre 30Km/h e 40Km/h e nas descidas entre 80Km/h e 100Km/h
- As mulheres profissionais conseguem pedalar nos trechos planos entre 45Km/h e 50Km/h, nas subidas entre 20Km/h e 30Km/h e nas descidas entre 70Km/h e 90Km/h
- Os homens amadores conseguem pedalar nos trechos planos entre 30Km/h e 45Km/h, nas subidas entre 12Km/h e 20Km/h e nas descidas entre 50Km/h e 70Km/h
- As mulheres amadoras conseguem pedalar nos trechos planos entre 25Km/h e 40Km/h, nas subidas entre 10Km/h e 15Km/h e nas descidas entre 50Km/h e 65Km/h

#### **Na corrida:**

- Os homens profissionais conseguem fazer cada Km entre 4min e 4min20seg
- As mulheres profissionais conseguem fazer cada Km entre 4min20seg e 4min50seg
- Os homens amadores conseguem fazer cada Km entre 5min e 7min
- As mulheres amadoras conseguem fazer cada Km entre 5min30seg e 7min

#### **Nas transições:**

Apenas 1 atleta pode passar pelos portais de transição por vez, tanto na entrada quanto na saída. Ou seja, por exemplo, se 3 atletas conseguem terminar a natação ao mesmo tempo, eles precisam sair em fila indiana até chegar nos locais onde estão as suas bicicletas (mas o processo de transição, após passar pelo portal, pode ser feito em paralelo).

- Os homens profissionais conseguem fazer a T1 entre 1min30seg e 2min30seg e a T2 entre 2min e 3min
- As mulheres profissionais conseguem fazer a T1 entre 1min30seg e 2min30seg e a T2 entre 2min30seg e 6min
- Os homens amadores conseguem fazer a T1 entre 2min30seg e 5min30seg e a T2 entre 4min e 10min
- As mulheres amadoras conseguem fazer a T1 entre 2min30seg e 6min30seg e a T2 entre 4min30seg e 12min

## 2 Requisitos

### 2.1 Linguagem

O simulador deve ser escrito em C e toda a gerência de threads deve ser feita utilizando POSIX threads (pthreads). Programas escritos em outra linguagem ou utilizando alguma biblioteca extra para gerenciar as threads **terão nota zero**.

Informações sobre como programar utilizando pthreads podem ser encontradas na seção 4.6 do livro do Andrews (basta ler até a subseção 4.6.1 inteira), na página da wikipedia em [http://en.wikipedia.org/wiki/POSIX\\_Threads](http://en.wikipedia.org/wiki/POSIX_Threads) ou no tutorial da IBM disponível em <http://www.ibm.com/developerworks/library/l-posix1/index.html>.

### 2.2 Detalhes do código

Seu simulador deve criar  $m$  threads “atleta” e 1 thread “classificacao”. Seu código deverá usar algum gerador de números aleatórios para definir as velocidades dos atletas durante os diversos trechos da prova. As velocidades devem ser sorteadas a cada 100m no caso da natação e a cada quilômetro nos casos da etapa de ciclismo e da etapa de corrida. Você pode sortear os valores seguindo a distribuição de probabilidade de sua preferência. O objetivo da thread “classificacao” é imprimir na saída padrão as listas parciais (apenas os 3 primeiros de cada grupo) durante a execução da simulação informando os tempos totais e separados para cada classe de atletas de 30 em 30 minutos.

O simulador terá ainda que receber na linha de comando a opção `-debug`. Quando essa opção for passada, as listas completas de cada grupo (não apenas os 3 primeiros) com as classificações de todos os atletas devem ser impressas na saída padrão de 1 em 1 minuto.

Seu código deve possuir 5 variáveis compartilhadas (não modifique os nomes no seu código): “PortalT1Ent”, “PortalT1Sai”, “estrada”, “PortalT2Ent”, “PortalT2Sai” que representarão respectivamente os portais de entrada e saída da transição T1, a estrada de 180Km utilizada na etapa de ciclismo e os portais de entrada e saída da transição T2. A variável `estrada` deve ser declarada como um vetor com 180 posições. Em um dado instante de tempo, a posição  $i$  de `estrada` deve ter armazenado os identificadores de todos os atletas que estão dentro do intervalo  $[i, i + 1)$ Km. Cada thread `atleta` tem a obrigação de escrever seu identificador na posição correta do vetor `estrada` a cada momento em que ele entra em um novo quilômetro, e de remover seu identificador da posição referente ao quilômetro que ele acabou de sair.

A saída do seu programa deve ser as impressões das classificações parciais durante a simulação da prova e, ao término da prova, a classificação final. Não há um formato padrão para a saída do seu programa. Basta que ela informe tudo que foi solicitado.

Com relação à entrada, seu simulador deve receber apenas um argumento que é o nome do arquivo com as informações sobre os atletas e sobre o circuito de ciclismo. O formato do arquivo segue abaixo:

```
hp
mp
ha
ma
P | S | D
k
P | S | D
k
```

P | S | D  
k  
...

As letras minúsculas representam valores inteiros:

- *hp*: a quantidade de atletas masculinos profissionais
- *mp*: a quantidade de atletas femininas profissionais
- *ha*: a quantidade de atletas masculinos amadores
- *ma*: a quantidade de atletas femininas amadoras
- *k*: a distância de cada trecho da prova de ciclismo (o tipo de etapa é definido na linha imediatamente acima)

As letras maiúsculas de fato aparecerão como letras no arquivo:

- *P*: a quilometragem da linha imediatamente abaixo é referente a um trecho plano
- *S*: a quilometragem da linha imediatamente abaixo é referente a um trecho de subida
- *D*: a quilometragem da linha imediatamente abaixo é referente a um trecho de descida

Segue abaixo um exemplo de arquivo definindo 500 atletas: 50 homens profissionais, 50 mulheres profissionais, 200 homens amadores e 200 mulheres amadoras; e uma etapa de ciclismo com as seguintes características: trecho plano de 50Km, subida de 10Km, descida de 20Km e um outro trecho plano de 100Km.

50  
50  
200  
200  
P  
50  
S  
10  
D  
20  
P  
100

Não há necessidade de validar a entrada.

Lembre que seu programa é um simulador. Ou seja, a simulação não precisa levar as 8 horas que uma prova do IronMan leva. Basta que os cálculos referentes aos tempos estejam de acordo com a velocidade definida para cada atleta.

### 3 Relatório

Você deve entregar um relatório apresentando uma saída para a execução do código com cada um dos seguintes arquivos de entrada:

`arquivo1.txt:`

5  
5  
20  
20  
P  
50  
S  
10  
D  
20  
P  
100

`arquivo2.txt:`

20  
15  
100  
65  
P  
10  
S  
10  
D  
10  
P  
20  
S  
6  
D  
10  
P  
20  
S  
10  
D  
10  
P  
30  
S  
10

D  
10  
P  
24

Apesar do relatório ter que apresentar os resultados para os dois arquivos acima, não significa que o EP será avaliado apenas com essas duas entradas. Serão feitas mais avaliações com quantidades de atletas de no máximo 500.

## 4 Sobre a entrega

Você deve entregar um arquivo .tar.gz contendo os seguintes itens:

- fonte, Makefile (ou similar), arquivo LEIAME;
- relatório em .pdf.

O desempacotamento do arquivo .tar.gz deve produzir um diretório contendo os itens. O nome do diretório deve ser ep1-membros\_da\_equipe. Por exemplo: ep1-joao-maria.

A entrega do .tar.gz deve ser feita através do PACA.

O EP pode ser feito individualmente ou em dupla.