

Aula 19: Trabalho 4

Prof.: Paulo Roberto Nunes de Souza

Trabalho 4

Um determinado hospital experimental frequentemente altera os seus processos para pesquisar qual seria o melhor. Atualmente um dos elementos a serem experimentados é a triagem do pronto socorro.

Cada paciente, ao dar entrada no pronto socorro, é abordado por um enfermeiro que preenche um prontuário com alguns dados. Estes dados serão utilizados para determinar quem terá prioridade de atendimento no pronto socorro.

Prontuário de triagem

O enfermeiro da triagem preenche um prontuário com os dados de cada paciente que dá entrada no pronto socorro. Os dados deste prontuário são:

Campo 1 Nome (até 50 letras)

Campo 2 Grau de gravidade (de 0 a 4)

Campo 3 Data de nascimento (AAAAMMDD)

Campo 4 Estado Civil (0 ou 1)

Campo 5 Quantidade de filhos

Campo 6 Data e Hora de chegada ao pronto socorro (MMDDAAAHHMM)

Campo 7 Com primeiros socorros (0 ou 1)

Campo 8 Renda anual (999999.99)

Os campos do prontuário seguem algumas codificações. O grau de gravidade pode receber um dentre 5 valores, sendo o 0 a gravidade mais baixa e 5 a gravidade mais alta. O campo estado civil utiliza 0 para representar solteiro e 1 para representar casado. O campo de data e hora de chegada dispõe a informação toda concatenada com a hora representada no formato 24h, ou seja, com as horas indo de 00 a 23. O campo que informa se o paciente deu entrada com primeiros socorros pode receber dois valores, 0 para paciente sem primeiros socorros e 1 para pacientes com primeiros socorros. Além da codificação própria, cada campo é representado por um número para facilitar a sua identificação nos procedimentos de triagem.

Os prontuários preenchidos com os dados dos pacientes atualmente sendo atendidos no pronto socorro são passados diariamente, em arquivo, ao setor de pesquisa, que irá projetar as novas rotinas experimentais de triagem. O arquivo de prontuários segue o seguinte formato:

```
nome_1;gravidade_1;data_1;civil_1;qtdfilhos_1;dthrchegada_1;socorro_1;renda_1
nome_2;gravidade_2;data_2;civil_2;qtdfilhos_2;dthrchegada_2;socorro_2;renda_2
nome_3;gravidade_3;data_3;civil_3;qtdfilhos_3;dthrchegada_3;socorro_3;renda_3
nome_4;gravidade_4;data_4;civil_4;qtdfilhos_4;dthrchegada_4;socorro_4;renda_4
...
nome_n;gravidade_n;data_n;civil_n;qtdfilhos_n;dthrchegada_n;socorro_n;renda_n
```

Veja a seguir o exemplo de um arquivo de prontuários.

Prontuário de entrada no pronto socorro

1	Maribel Parent;3;19660525;0;0;201103271832;0;269.47
	Jarrett Qualey;0;19320301;1;2;201201282228;0;0.03
	Lorina Henkle;3;19660802;0;3;201206071008;1;0.00
	Willette Womac;1;19470325;1;0;201210200233;0;0.06
	Karen Lafon;0;19550404;1;1;201209280309;1;10271.08
6	Lynell Larkin;2;19540520;0;0;201204120355;0;0.00
	Marilyn Parmelee;2;19910827;1;2;201107030518;1;0.01
	Marcelle Budd;1;19391219;1;1;201211111448;0;0.00
	Gena Coyle;3;19820826;1;1;201206090155;1;29.49
	Kourtney Hallberg;4;19670809;0;2;201110200456;1;0.00
11	Charlyn Harrelson;1;19120418;0;2;201110180555;1;7478.13
	Deirdre Cancel;4;19920723;1;1;201203060421;1;168222.20
	Lakeesha Ary;3;19930212;1;0;201106260443;1;0.59
	Leatrice Pendergraft;4;19980718;0;0;201111181326;0;0.64
	Corrina Dawdy;0;19850125;0;2;201106271728;0;112.58
16	Clyde Betters;0;19980708;1;0;201203091321;1;14192.13
	Wilfredo Babiarz;3;19690625;1;3;201110042002;0;29743.89
	France Dohm;4;19661024;1;1;201108080647;0;0.01
	Elizebeth Pound;2;19340813;0;7;201205081234;1;36.06
	Tyrell Port;4;19250303;0;2;201109121418;1;28.72
21	Genaro Pritt;3;19510411;0;1;201201211946;0;621670.53
	Charlotte Pech;2;19170225;1;0;201108271112;0;6120.49
	Ezequiel Runk;0;19890202;1;2;201109011034;1;17.66
	Octavio Vitolo;0;19380118;1;3;201107050746;0;0.08
	Melodi Gunnells;2;19931120;1;8;201212050526;1;56185.31

Programa de triagem

Você deverá desenvolver o programa que irá ordenar os pacientes dos pronto socorro segundo os critérios estabelecidos pela equipe de pesquisa. Para tal, o programa deve ser flexível para ordenar os dados do cliente segundo quaisquer dos campos do prontuário, em ordem crescente ou decrescente.

A chamada do programa deve seguir o seguinte formado:

```
<programa> <arquivo_de_prontuario> <arquivo_de_saida> <campos> <ordem>
```

Supondo que o nome do arquivo executável criado seja "triagem", o nome do arquivo de prontuarios seja "prontuario.dat", o nome do arquivo de saída seja "atendimento.dat", os campos a serem utilizados para ordenação sejam "gravidade", "primeiros socorros", "data de nascimento" e "quantidade de filhos", nesta ordem, o parâmetro campos será "2735" e considerando que se deseje que se ordene os campos 2, 3 e 5 em ordem crescente e o campo 7 em ordem decrescente, o parâmetro ordem será "+-++", a linha de comando ficaria da seguinte forma:

Execução do programa

```
./triagem prontuario.dat atendimento.dat 2735 +-++
```

Como serão utilizados vários campos como critérios de ordenação, seu programa deverá utilizar o método de ordenação por radicais (Radixsort) sendo que em cada passada deve ser utilizado um método de ordenação estável que seja mais adequado ao campo utilizado como chave de ordenação.

Os métodos de ordenação estável que podem ser utilizados são:

- Bubble Sort
- Insertion Sort
- Merge Sort
- Counting Sort

Arquivo de Saída

O programa deverá gerar como saída um arquivo, que tenha o nome que foi informado na linha de comando, contendo todos os clientes do prontuário ordenados segundo os campos e a ordem que foram informados na linha de comando. O arquivo utilizado como exemplo se fosse submetido à linha de comando informada geraria o seguinte arquivo de saída:

atendimento.dat

	Karen Lafon;0;19550404;1;1;201209280309;1;10271.08
	Ezequiel Runk;0;19890202;1;2;201109011034;1;17.66
	Clyde Betters;0;19980708;1;0;201203091321;1;14192.13
4	Jarrett Qualey;0;19320301;1;2;201201282228;0;0.03
	Octavio Vitolo;0;19380118;1;3;201107050746;0;0.08
	Corrina Dawdy;0;19850125;0;2;201106271728;0;112.58
	Charlyn Harrelson;1;19120418;0;2;201110180555;1;7478.13
	Marcelle Budd;1;19391219;1;1;201211111448;0;0.00
9	Willette Womac;1;19470325;1;0;201210200233;0;0.06
	Elizebeth Pound;2;19340813;0;7;201205081234;1;36.06
	Marilyn Parmelee;2;19910827;1;2;201107030518;1;0.01
	Melodi Gunneils;2;19931120;1;8;201212050526;1;56185.31
	Charlotte Pech;2;19170225;1;0;201108271112;0;6120.49
14	Lynell Larkin;2;19540520;0;0;201204120355;0;0.00
	Lorina Henkle;3;19660802;0;3;201206071008;1;0.00
	Gena Coyle;3;19820826;1;1;201206090155;1;29.49
	Lakeesha Ary;3;19930212;1;0;201106260443;1;0.59
	Genaro Pritt;3;19510411;0;1;201201211946;0;621670.53
19	Maribel Parent;3;19660525;0;0;201103271832;0;269.47
	Wilfredo Babiarez;3;19690625;1;3;201110042002;0;29743.89
	Tyrell Port;4;19250303;0;2;201109121418;1;28.72
	Kourtney Hallberg;4;19670809;0;2;201110200456;1;0.00
	Deirdre Cancel;4;19920723;1;1;201203060421;1;168222.20
24	France Dohm;4;19661024;1;1;201108080647;0;0.01
	Leatrice Pendergraft;4;19980718;0;0;201111181326;0;0.64

Critérios de correção

O programa deve ser escrito na linguagem de programação Ansi-C. Os programas submetidos serão testados usando-se o compilador GCC no sistema operacional Linux.

O trabalho será corrigido segundo 5 critérios:

P Ausência de plágio

A Adequação ao solicitado

C Correção dos resultados

E Entrega no prazo

D Desempenho computacional

Para a obtenção da nota do trabalho, cada critério receberá um valor entre 0 e 1 seguindo os testes efetuados pelo professor. A nota será o resultado da seguinte ponderação:

$$Nota = P \times A \times (7 \times C + E \times (2 + D))$$

Grupo

Os trabalhos devem ser feitos em grupos de até 2 pessoas.

Entrega

Os resultados devem ser entregue em meio digital até o dia **09/08/2013** pelo e-mail **paulonunes@cca.ufes.br**.