

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
PROGRAMAÇÃO II – Prof<sup>a</sup>: Claudia Boeres  
TRABALHO COMPUTACIONAL – 2017/2  
Entrega: 15/12/2017

Leia atentamente **TODO** o enunciado do trabalho (a especificação do problema e os detalhes sobre a confecção, submissão e avaliação do trabalho).

### 1) Apresentação

Cada dupla de alunos deve desenvolver uma solução computacional usando a linguagem C para o problema enunciado na Seção 2. No processo da resolução desse trabalho, o aluno deve produzir um código resultante das seguintes etapas:

1. Compreensão do problema;
2. Planejamento e definição da estratégia de solução;
3. Elaboração de várias soluções, de onde possa ser escolhida a melhor;
4. Codificação da solução na linguagem C.

Os códigos devem apresentar as seguintes características, que serão consideradas na correção:

1. **Identação** correta;
2. **Modularização**: sempre que possível, crie funções para modularizar o seu programa;
3. **Estratégias de solução** visando a simplificação do código, evitando repetição de operações desnecessárias;
4. **Documentação**: o seu programa deve ser todo comentado, explicando o que está sendo codificado. O início do código deve conter um cabeçalho seguindo o seguinte formato:

```
/*  
Trabalho Computacional 2 – Programação II – 2017/2 – Ciência da Computação  
Grupo: <nome completo do Componente 1> e <nome completo do Componente 2>  
*/
```

### 2) O problema proposto

#### Agendamento de consultas em uma clínica médica

A demanda pela automatização de processos em estabelecimentos comerciais cresce a cada dia. Em uma clínica médica por exemplo, automatizar o processo de marcação de consultas para cada médico cadastrado na clínica pode facilitar, e muito, o cotidiano do pessoal administrativo que lida com essas informações.

Seguindo esse contexto, o objetivo do trabalho é implementar uma agenda eletrônica que leia as informações referentes aos médicos de uma clínica e de seus pacientes e efetue a marcação automática de consultas

solicitadas pelos pacientes. As informações de entrada devem ser fornecidas em arquivos texto, de acordo com formato preestabelecido. Além da marcação automática, o seu programa deve imprimir um relatório com informações relativas às consultas marcadas.

### 3) Os dados de entrada

As informações dos médicos que trabalham na clínica e dos pacientes que desejam marcar consultas com esses médicos são fornecidas como entrada na forma de arquivos texto.

O seu programa deverá manipular 5 arquivos texto, um com as informações dos médicos e os quatro restantes, com as informações dos pacientes que desejam marcar suas consultas em uma dada semana do mês. Consideraremos neste trabalho o período de 4 semanas. Os nomes dos arquivos de entrada são `dadosMedicos.txt`, `listaPacientes-Semana1.txt`, `listaPacientes-Semana2.txt`, `listaPacientes-Semana3.txt` e `listaPacientes-Semana4.txt`. Um exemplo de cada arquivo texto é dado a seguir. Neste exemplo, temos os dados de 3 médicos que fazem parte da clínica (arquivo `dadosMedicos.txt`); 3 pacientes que querem marcar consultas na 1ª semana (arquivo `listaPacientes-Semana1.txt`); 3 pacientes que querem marcar consultas na 2ª semana (arquivo `listaPacientes-Semana2.txt`); 2 pacientes que querem marcar consultas na 3ª semana (arquivo `listaPacientes-Semana3.txt`) e 4 pacientes que querem marcar consultas na 4ª semana do mês (arquivo `listaPacientes-Semana4.txt`).

#### `dadosMedicos.txt`

```
Joao Silva da Costa
32178
clinica geral
2a 08 09 10
5a 08 09 10
6a 16 17
```

```
Ana Carolina Maia
44516
gastro
3a 13
5a 08 09
```

```
Marcelo Andrade
84672
gastro
3a 13 14
5a 08 09 10
```

listaPacientes-Semana1.txt	listaPacientes-Semana2.txt
Maria Helena Gomes da Silva 112 27955553574 10 10 1994 Joao Silva da Costa	Amelia de Albuquerque 117 27955352299 15 06 2000 Marcelo Andrade
Ana Claudia Soares 113 27952255522 05 12 1967 Ana Carolina Maia	Joana Cruz de Almeida 119 27955545571 04 01 1990 Marcelo Andrade
Carlos Lins 114 27955553434 06 03 1960 Joao Silva da Costa	Carmen Lucia dos Santos Neves 120 27955512523 23 10 1949 Joao Silva da Costa

listaPacientes-Semana3.txt	ListaPacientes-Semana4.txt
Ricardo Silveira 118 27951151518 20 03 2001 Ana Carolina Maia	Francisco Buarque de Almeida 115 27956655576 20 10 1939 Joao Silva da Costa
Luisa da Silva Reis 140 27988888765 17 05 1941 Joao Silva da Costa	Jose Henrique Silveira 112 27955552558 18 08 1999 Ana Carolina Maia
	Luiz Alberto dos Santos 116 27955435521 21 08 1950 Ana Carolina Maia
	Alvaro Campos de Albuquerque 134 27954955332 07 07 1985 Ana Carolina Maia

Observe que no arquivo dadosMedicos.txt, as informações de cada médico são dadas em blocos de linhas, cada bloco separado por uma linha em branco. Cada bloco possui o nome do médico (1ª linha), seu número de matrícula (2ª linha), sua especialidade (3ª linha) e os seus horários indisponíveis na semana. Os horários são representados usando, no máximo, as próximas 5 linhas, indicando em cada linha, o dia da semana, seguido da hora de início da consulta. Considere que todas as consultas tem duração de 1 hora. Assim, por exemplo, a linha 2a 08 09 10 indica

que o médico não pode atender na segunda feira, nos horários de 8 às 9, 9 às 10 e 10 às 11. Assim, o formato dos dados em cada bloco de linhas referente às informações de um médico é:

```
<nome do médico>
<matrícula do médico>
<especialidade>
<d>a <h1> ... <hn>, onde  $d = 1, \dots, 6$ ;  $n = 1, \dots, 10$  ( $n$  indica o número máximo de horários indisponíveis naquele dia e  $h_i = 8, \dots, 17$ , para  $i = 1, \dots, n$ , representa cada horário do dia, que começa às 8 da manhã e vai até 18 horas.
```

Nos arquivos listaPacientes-Semana1.txt a listaPacientes-Semana4.txt, as informações de cada paciente também são dadas em blocos de linhas, cada bloco separado por uma linha em branco. Cada bloco segue o mesmo formato, que é:

```
<nome do paciente>
<número de identificação>
<número do telefone>
<data de nascimento>
<nome do médico solicitado>
```

#### 4) O código

De acordo com as informações fornecidas nos arquivos de entrada, faça um programa em C que implemente as seguintes solicitações:

**1. Marcação semanal de consultas:** seu programa deve construir, para cada médico, a tabela semanal de consultas a partir dos arquivos de entrada relativos aos médicos e aos pacientes. A tabela deve ser representada por uma matriz de dimensões fixas  $h \times d$ , onde  $h$  é o número máximo de horários (10) e  $d$ , o número de dias úteis da semana (5).

Esta tabela deve informar os horários indisponíveis (preenchidos com o valor -1); os horários ocupados com consultas (preenchidos com os números de identificação de cada paciente e os horários vagos (preenchidos com 0).

Os horários indisponíveis são aqueles informados pelos médicos no arquivo de entrada e o horário de almoço (entre 12 e 13 horas) em todos os dias.

A alocação das consultas nas tabelas de cada semana deve ser feita aleatoriamente dentre as opções disponíveis (ou seja, os horários que ainda estão preenchidos com 0). Se não há opções disponíveis na tabela para marcar uma nova consulta na semana, seu programa deve imprimir uma mensagem informando ao paciente que não há opções de horários na semana, sugerindo que ele tente nas próximas.

Por exemplo, ao final da Semana 1, a tabela de horários do médico João Silva da Costa deveria ser preenchida da seguinte maneira:

	2a	3a	4a	5a	6a
8-9	-1	0	0	-1	0
9-10	-1	0	0	-1	0
10-11	-1	0	0	-1	0
11-12	0	0	0	0	0
12-13	-1	-1	-1	-1	-1
13-14	0	0	0	114	0
14-15	112	0	0	0	0
15-16	0	0	0	0	0
16-17	0	0	0	0	-1
17-18	0	0	0	0	-1

As tabelas a serem criadas conforme as especificações acima devem fazer parte de vetores de estruturas que armazenam todas as informações de todos os médicos da clínica semanalmente. Assim, para cada semana, um vetor de elementos do tipo *agMedico* deve ser definido. Além disso, um vetor de elementos do tipo *cliente* também deve ser criado. Os tipos *agMedico* e *cliente* são estruturas em C que armazenam respectivamente todas as informações relativas aos médicos e aos pacientes. Esses tipos devem ser definidos como a seguir. Nas definições dos tipos, os identificadores *dim*, *h* e *d* são definidos como constantes iguais a 30, 10 e 5 e representam respectivamente o tamanho máximo que um nome pode possuir e o número máximo de horários e de dias de uma tabela.

<pre>typedef struct {     char nome[dim];     int id;     char especialidade[dim];     int agenda[h][d]; } agMedico;</pre>	<pre>typedef struct {     char nome[dim];     int id;     long int fone;     int idade;     char medico[dim]; } cliente;</pre>
--	--

**2. Impressão das tabelas de consultas de cada médico da clínica:** seu programa deve criar um arquivo de saída para cada médico, chamado **<nome do medico sem espaços em branco>.txt**, com as tabelas de horários relativas às 4 semanas. Por exemplo, o arquivo de nome **JoaoSilvadaCosta.txt** deveria conter as seguintes informações:

Medico: Joao Silva da Costa  
Id: 32178  
Especialidade: clinica geral

Quadro de consultas semanais

Semana 1:

	2a	3a	4a	5a	6a
8-9	-1	0	0	-1	0
9-10	-1	0	0	-1	0
10-11	-1	0	0	-1	0
11-12	0	0	0	0	0
12-13	-1	-1	-1	-1	-1
13-14	0	0	0	114	0

14-15	112	0	0	0	0
15-16	0	0	0	0	0
16-17	0	0	0	0	-1
17-18	0	0	0	0	-1

Semana 2:

	2a	3a	4a	5a	6a
8-9	-1	0	0	-1	0
9-10	-1	0	0	-1	120
10-11	-1	0	0	-1	0
11-12	0	0	0	0	0
12-13	-1	-1	-1	-1	-1
13-14	0	0	0	0	0
14-15	0	0	0	0	0
15-16	0	0	0	0	0
16-17	0	0	0	0	-1
17-18	0	0	0	0	-1

Semana 3:

	2a	3a	4a	5a	6a
8-9	-1	0	0	-1	0
9-10	-1	0	0	-1	0
10-11	-1	0	0	-1	0
11-12	0	0	0	0	0
12-13	-1	-1	-1	-1	-1
13-14	0	0	0	0	0
14-15	0	0	0	0	0
15-16	0	0	0	0	0
16-17	0	0	0	0	-1
17-18	0	140	0	0	-1

Semana 4:

	2a	3a	4a	5a	6a
8-9	-1	0	0	-1	0
9-10	-1	0	0	-1	0
10-11	-1	0	0	-1	0
11-12	0	0	0	0	0
12-13	-1	-1	-1	-1	-1
13-14	0	0	0	0	0
14-15	115	0	0	0	0
15-16	0	0	0	0	0
16-17	0	0	0	0	-1
17-18	0	0	0	0	-1

**3. Médico mais popular:** seu programa deve indicar o médico mais popular, ou seja, aquele que, nas 4 semanas, concedeu o maior número de consultas. Se houver empate, informe o nome de todos os campeões.

**4. Ranking da procura por especialidades médicas:** uma especialidade popular é aquela que mais consultas foi requisitada para a especialidade, independente do médico, em todo o período analisado (as quatro semanas). Informe em ordem decrescente de popularidade, as

especialidades e para cada uma delas, o número de pacientes atendidos por semana.

**5. Especialidade mais procurada por faixa etária:** considerando quatro faixas etárias divididas de 0 a 25 anos (faixa 1); 26 a 50 anos (faixa 2), 51 a 75 anos (faixa 3) e 76 a 100 anos (faixa 4), indique a especialidade campeã por faixa etária, em termos de procura por consultas.

**6. Impressão de relatório de informações da clínica:** seu programa deve criar um arquivo de saída chamado **dadosClinica.txt**, com todas as informações solicitadas nos itens 3 a 5 acima.

**7. Teste do programa:** para testar o programa, crie pelo menos 5 conjuntos de arquivos de entrada (dados dos médicos + dados dos pacientes), seguindo as especificações descritas na Seção 2.1.

### **3) Data de Entrega**

O trabalho deverá ser entregue até 15:00 horas do dia **15/12/2017**.

### **4) Forma de Entrega**

Os trabalhos devem ser enviados por e-mail para [boeres@inf.ufes.br](mailto:boeres@inf.ufes.br), seguindo ESTRITAMENTE o seguinte formato:

- **Assunto do email:** [PROGII-TC] <Nome Completo do Aluno 1> + <Nome Completo do Aluno 2>
- **Arquivos para enviar:** um arquivo "TC.zip", com o arquivo fonte trabalho.c, contendo o código de resolução desse trabalho e os arquivos de testes criados pela dupla.

### **5) Observações importantes**

- O trabalho deve ser feito em dupla.
- Devem ser usados no seu código vetores de estruturas e manipulação de arquivos.
- o código deverá fornecer os arquivos de saída solicitados, contendo todas as respostas necessárias.
- O trabalho que não estiver compilando por erros graves de compilação não será corrigido. Aqueles que tiverem erros simples, terão um valor relativo descontado da nota final.
- Trabalhos evidentemente iguais receberão nota ZERO.
- Um teste sobre o trabalho, valendo 1 ponto, será aplicado em horário a ser combinado com a turma.
- Não usar caracteres especiais tais como ç, ã, õ, é, etc etc no nome do arquivo.

- O recebimento dos trabalhos é automatizado. Siga as instruções à risca pois algum erro na submissão pode inviabilizar a entrega do seu trabalho. Não deixe para enviar seu trabalho nos momentos finais de seu prazo. É comum a ocorrência de problemas em virtude de erros na submissão. Logo, enviem com algumas horas de antecedência para que haja tempo hábil para eventuais correções.

BOM TRABALHO!!!