

**Universidade Federal de Pernambuco**  
**Centro de Informática**  
**Introdução à Programação – IF669ec**

**Lista 5 – Structs**

**Prof: Adriano Sarmento**

**Data: 07/05/2012**

**Data de Entrega: 14/05/2012**

- É proibido usar a biblioteca conio.h
- Em todas as questões da lista, é obrigatório o uso de *structs*.
- Qualquer cópia entre alunos ou da internet terá como consequência a anulação da lista de todos os envolvidos
- Dúvidas: [listaip@googlegrousp.com](mailto:listaip@googlegrousp.com)
- Toda Memória alocada deve ser devidamente liberada no final do programa

1) **Dominó** é o jogo formado por peças retangulares, dotadas normalmente de uma espessura que lhes dá a forma de paralelepípedo, em que uma das faces está marcada por pontos indicando valores numéricos. O jogo aparentemente surgiu na China e sua criação é atribuída a um santo soldado chinês chamado Hung Ming, que viveu de 243 a.C a 182 a.C. O conjunto tradicional de dominós, conhecido como sino-europeu, é formado por 28 peças, ou pedras. Cada face retangular de dominó é dividida em duas partes quadradas, ou "pontas", que são marcadas por um número de pontos de 0 a 6, ou deixadas em branco. Um jogo de dominós é equivalente a um baralho de cartas ou jogo de dados, que podem ser jogados em uma diversidade indeterminada de maneiras.

[fonte: Wikipedia]

Faça um programa em C que simule uma partida de dominó. O jogo deve ser jogado entre dois usuários. Cada usuário possui um vetor de ponteiros com 3 posições para um tipo estruturado dominó, que possui dois inteiros que representam os seus números. O jogador que obtiver maior número de pontos somando cada número das peças que ele possui, deverá iniciar o jogo. Caso o jogador da vez não tenha uma peça apta para jogar, ele simplesmente passa a vez. A distribuição das peças entre os dois jogadores deve ser ALEATÓRIA. Mostre sempre na tela o número da rodada, as peças do jogador da vez e que valores ele pode jogar. Cuidado com o “gato-por-lebre” (quando um jogador joga uma peça indevida).

2) A Organização de Empresas Turísticas de PERNAMBUCO contratou os alunos de Introdução à Programação para desenvolver um programa que gere a média de lucros e prejuízos de uma companhia aérea.

Você receberá como entrada:

1. Os seguintes dados de 10 vôos:
  - número de vôo;
  - tipo de avião utilizado (707, 727, 737);
  - preço da passagem.
2. N pedidos de reserva de passagens, onde cada pedido contém:
  - número da identidade do passageiro; e
  - número do voo desejado;

Seu trabalho será verificar, para cada passageiro, se há disponibilidade no voo. Em caso afirmativo,

atualizar o número de lugares disponíveis e imprimir o número da identidade do passageiro, o número do voo desejado, o preço da passagem e a mensagem “RESERVA CONFIRMADA”.

Em caso negativo, imprimir a mensagem “VOO LOTADO”.

Ao final, imprimir uma estatística de lucros e prejuízos por voo e no total da companhia aérea. Considerar que a lotação de 60% de capacidade de cada avião não produz lucros nem prejuízos. Sendo assim, acima deste valor é lucro e abaixo é prejuízo.

Observação: Capacidade de cada avião: 707-200 lugares; 727- 170 lugares; 737- 120 lugares.

3) Uma rede de Cinemas do Recife esta com problema de superlotação nas suas salas, por conta da venda excessiva de ingressos. Para resolver esse problema, eles contrataram os geniais estudantes de IP.

Inicialmente, seu programa pedirá ao usuário a quantidade de salas que existem no Cinema, isso só deverá ser feito uma vez. Cada Sala é uma estrutura que possui um número de identificação, um ponteiro para a estrutura Filme e um Vetor de Ponteiros para uma estrutura do tipo Cadeira. Você devera pedir também quantas cadeiras tem em cada sala.

Feito isso, o programa deve exibir o seguinte menu:

1. **Cadastrar Filme:** Nessa função, você deve pedir ao usuário os dados do filme que passará em uma sala. Cada Filme possui um nome (máximo de 50 caracteres), tempo de duração, classificação etária e o valor do ingresso. Caso um filme seja cadastrado sobre outro que já existe, todas as cadeiras da sala devem estar vazias.
2. **Comprar Ingresso:** Deverá ser mostrado os filmes que passam em cada sala e uma vez escolhido o filme, deve-se mostrar todas as cadeiras disponíveis na sala. No lugar das cadeiras ocupadas deve-se mostrar um espaço em branco. Cada Cadeira contém o nome do usuário que a escolheu (máximo de 50 caracteres) e a idade do mesmo. Caso a idade do usuário não seja compatível com a faixa etária do filme, uma mensagem de erro deve ser impressa.
3. **Mostrar Rendimentos:** Mostra o Rendimento que o Cinema esta tendo com cada filme e no total.
4. **Sair:** Ao Sair, o seu programa deverá dizer se o cinema teve lucro ou prejuízo nesse dia. Se a soma de todos os ingressos comprados no dia for maior ou igual a 70% do total que o cinema poderia ter, o cinema tem lucro.

Exemplo(da esquerda pra direita):

```
Quantidade de Salas do Cinema: 2
Quantidade de Cadeiras da Sala 1: 25
Quantidade de Cadeiras da Sala 2: 36

1) Cadastrar Filme
2) Comprar Ingresso
3) Mostrar Rendimentos
4) Sair
Opcao: 1

->Nome do Filme: Pokemon 2000
->Tempo de Duracao (em minutos): 120
->Classificacao Etaria: 8
->Valor do Ingresso: R$50

1) Cadastrar Filme
2) Comprar Ingresso
3) Mostrar Rendimentos
4) Sair
Opcao: 2

->Filmes:
Sala 1) Pokemon 2000
Sala 2) Livre
->Opcao: 1
```

```
-->Cadeiras Livres:
 1 2 3 4 5
 6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
-->Opcao: 8
--->Nome: Thiago Soares
--->Idade: 21 anos

1) Cadastrar Filme
2) Comprar Ingresso
3) Mostrar Rendimentos
4) Sair
Opcao: 3

Rendimentos da Sala 1: R$50
Rendimentos da Sala 2: R$0
Total: R$50

1) Cadastrar Filme
2) Comprar Ingresso
3) Mostrar Rendimentos
4) Sair
Opcao: 4

Hoje o Cinema teve Prejuízo!
```

4) Este ano tem início o FAC(Falcão Arm's Championship), o Campeonato de quebra de braço oficial do Sylvester Stallone. Para esquentar o campeonato, as casas de apostas resolveram criar um simulador do campeonato que indique ao usuário em quem ele deve apostar. Para isso, eles precisam da ajuda dos mestres da Programação de IP.

Primeiro, deve-se pedir a quantidade N de lutadores do FAC, onde N tem que ser uma potência de 2. Depois, o usuário deve preencher os seguintes dados de cada Lutador:

1. Nome (30 caracteres)
2. Litros de Leite que ele toma por dia
3. Peso
4. Idade
5. Se ele é destro ou canhoto
6. Tamanho do Braço

Em seguida, os jogos devem acontecer na seguinte ordem:

Lutador 1 x Lutador N

Lutador 2 x Lutador (N-1)

...

Lutador (N/2) x Lutador (N/2 + 1)

O vencedor é o lutador que possui o maior Ki, onde o Ki é calculado da seguinte maneira:

$((\text{Peso}/\text{Idade}) * \text{Tamanho do Braço}) * \text{Leite}$

Em cada luta deve-se decidir se os lutadores lutarão com a mão esquerda ou direita e essa escolha é feita no Cara ou Coroa (faça uma função randômica). Caso a mão escolhida seja à direita, adiciona-se 20% do Ki ao(s) lutador(s) destro(s). A mesma coisa vale para a esquerda.

Você deve imprimir todas as lutas e o resultado de cada uma. Ao final de cada fase, os lutadores vencedores passam para a próxima fase e se enfrentam até restar apenas 1 lutador, sendo esse o Campeão do FAC e, conseqüentemente, quem o simulador indicará ao usuário para apostar.

Exemplo:

(entrada)

Digite a quantidade de Lutadores

2

Nome do Lutador 1:

Bruce Willis

Quantidade de Leite que o Lutador 1 toma (em litros):

5

Peso (em kg):

100

Idade:

53

Destro ou Canhoto?

Canhoto

Tamanho do braço (em centímetros) :

45

Nome do Lutador 2:

Van Damme

Quantidade de Leite que o Lutador 2 toma (em litros):

8

Peso (em kg)

100

Idade:

45

Destro ou Canhoto?

Destro

Tamanho do braço (em centímetros):

59

A Luta sera com a mão direita ou esquerda?

Direita

(Sáida)

Bruce Willis x Van Damme

Vencedor: Van Damme

Fim do Campeonato

Van Damme foi o Campeão!

5) Há alguns anos atrás, as Universidades públicas do Brasil instalaram um sistema maravilhoso e sem nenhum bug chamada SIG@. Neste sistema, os alunos podem escolher as cadeiras nas quais desejam se matricular no período, com a única restrição de não poderem cursar uma dada disciplina X sem antes terem cursado todas as disciplinas que tiverem sido estabelecidas como pré-requisitos de X.

Após alguns anos, o reitor da UFPE notou que muitos estudantes estavam sendo reprovados em muitas disciplinas, simplesmente porque estavam superlotando suas grades de horário (alguns alunos se matriculavam em até 17 cadeiras por período). Com toda a sua prudência, este ano o reitor irá adicionar uma regra limitando o número de disciplinas que cada estudante pode cursar por semestre a certo valor, mas os alunos estão bastante confusos agora na hora da matrícula. É aí que você entra na estória. O reitor resolveu disponibilizar um programa de computador para ajudar os alunos a fazerem suas escolhas de disciplinas, e solicitou sua ajuda. Mais precisamente, o reitor quer que o programa sugira as disciplinas a serem cursadas durante o curso da seguinte forma:

- A cada disciplina é atribuída uma prioridade.
- Se mais do que N disciplinas podem ser cursadas em um determinado semestre (obedecendo ao sistema de pré-requisitos), o programa deve sugerir que o aluno matricule-se nas N disciplinas de maior prioridade.
- Se N ou menos disciplinas podem ser cursadas em um determinado semestre, o programa deve sugerir que o aluno matricule-se em todas as disciplinas disponíveis.

Portanto, dadas a descrição de pré-requisitos para cada disciplina, a prioridade de cada disciplina, e o número máximo de disciplinas por semestre, seu programa deve calcular o número necessário de semestres para concluir o curso, segundo a sugestão do reitor, e a lista de disciplinas que o aluno deve matricular-se em cada semestre. Considere que nesse sistema, matriculam-se alunos de um único curso e que um aluno deve pagar todas as cadeiras cadastradas no sistema. Não há circularidade nos pré-requisitos (ou seja, se a disciplina B tem como pré-requisito a disciplina A então A não tem B como pré-requisito, direta ou indiretamente). O número total de disciplinas é no máximo igual a 50.

ENTRADA:

- Um inteiro H, representando a quantidade de disciplinas que NÃO POSSUEM pré-requisitos e o código, o nome e a prioridade ( $1 \leq P \leq 5$ , onde maior numero representa maior prioridade) das H disciplinas.
- Um inteiro K, representando a quantidade de disciplinas que POSSUEM pré-requisitos, o código, o nome, a prioridade ( $1 \leq P \leq 5$ , onde maior numero representa maior prioridade), a quantidade de disciplinas que são pré-requisito ( $1 \leq QTD \leq 4$ ) e os códigos das disciplinas que são pré-requisitos de cada uma das K disciplinas.  
Dica: Crie na estrutura um vetor de ponteiros para guardar o endereço das disciplinas que são pré-requisitos.
- Um N que representa a quantidade de disciplinas máximas por semestre.

SAIDA:

Um inteiro S, representando a quantidade de semestres necessários para que o estudante se forme. Depois mostre para cada semestre, o nome e o código das disciplinas que devem ser pagas naquele semestre pela sugestão do reitor.