



## Trabalho Prático 4

### Filas – Simulação de Atendimento do Restaurante Universitário da UFOP

**Valor: 0,5 pts (5% da nota total)**

**Documentação não-Latex: -0,1 pts**

**Interface gráfica com simulador: +0,1 pts**

**Impressão não frente-verso: -0,05 pts (ecologicamente correto)**

**Data de entrega:**

**03/11/2009**

O objetivo deste trabalho é aplicar o conceito de filas computacionais para simular o atendimento das filas aos usuários do Restaurante Universitário (RU) da UFOP.

O restaurante universitário é composto basicamente por cinco entidades: usuário (que faz fila para ser atendido/servido); bancadas (o local físico); servente (a pessoa que serve o usuário); ingrediente (que constituem o cardápio); vasilha (que armazena ingrediente do cardápio a ser servido).

Para elaborar o simulador, considere as seguintes regras.

- O restaurante universitário abre às 11 horas e fecha às 13h15min horas para o almoço, e abre às 18h e fecha às 19h30min para o jantar. Portanto, o RU fica aberto por 03h45min durante o dia.  
Defina macros (`#define`) para estes valores, e.g., `TEMPSIMAM`; `TEMPSIMPM`;
- O tempo de atendimento de cada usuário por cada servente deve variar de 3 segundos até 20 segundos.  
Defina macros (`#define`) para estes valores, e.g., `TEMPUSUMIN` e `TEMPUSUMAX`;
- Devem existir de uma a `NFILAS` filas para os usuários, quando eles entram no RU;  
Defina uma macro (`#define`) para este valores, e.g., `NFILAS`;
- Os usuários escolhem aleatoriamente qual fila entrar, e uma vez que a fila foi escolhida eles não trocam de fila;
- Existem usuários especiais; os vegetarianos. Estes usuários seguem apenas por uma das filas e no momento de serem servidos entram em uma fila especial (que existe na bancada desta fila), pois um de seus ingrediente é preparado especialmente. Crie um tempo de espera adicional para o cliente vegetariano.
- Inicialmente, quando o RU abre, somente uma bancada e 3 serventes estarão trabalhando;
- Para finalizar, quando o RU fechar, nenhum usuário entra mais nas filas. Porém os usuários que ainda estão dentro dele devem ser atendidos.
- Bancadas podem ser ativadas e desativadas caso o “gerente” do RU considere necessário;
- Obrigatoriamente no jantar uma fila deve estar sem funcionar, no mínimo;
- Considere que as quantidades mínima e máxima de bancadas sejam definidas por macros, .e.g. `QTDBANMIN` e `QTDBANMAX`;
- Uma bancada pode ser composta de 3 a 6 serventes;  
Defina macros (`#define`) para estes valores, e.g., `BANSERMIN` e `BANSERMAX`;
- Um servente não pode ficar trabalhando direto (sem intervalos) mais de 1h.  
Defina uma macro (`#define`) para este valor, e.g., `TEMPSERMAX`;
- Um servente deve ter um intervalo de descanso/almoço de pelo menos 30 minutos.



- Defina uma macro (`#define`) para este valor, e.g., `TEMPSERINT`;
- Considere que as quantidades mínima e máxima de serventes sejam definidas por macros, .e.g. `QTDSERMIN` e `QTSERMAX`;
- Considere que o cardápio do dia é composto por 6 ingredientes  
Defina uma macro (`#define`) para este valor, e.g., `NING`;
- Considere que cada ingrediente é armazenado em vasilhas, e cada vasilha tem sua capacidade (em gramas);  
Defina macros (`#define`) para estes valores, e.g., `CAPING1, ...`;
- Cada usuário pode ou não desejar ser servido de um determinado ingrediente (existe uma probabilidade) – cada ingrediente tem um “grau de aceitação” pelos usuários;  
Defina macros (`#define`) para estas probabilidades, e.g., `NPROBING1, ...`;
- A quantidade (em gramas) que cada usuário é servido pode variar.  
Defina macros (`#define`) para cada ingrediente, e.g., `QTDMINING1, QTDMAXING1`.
- Quando uma vasilha de um determinado ingrediente acabar, deve ser considerado um tempo para a substituição da vasilha;  
Defina macros (`#define`) para cada vasilha, e.g., `TEMPING1`.
- O simulador deve ter resolução de 1 segundo, ou seja, os controles devem ser atualizados segundo a segundo: um usuário pode chegar, pode sair da fila; uma servente pode abrir, pode fechar, etc.

Ao final de cada expediente (quando o RU fechar e não existir mais usuários para serem atendidos), devem ser impressos: a quantidade de usuários atendidos por cada servente e cada bancada, o tempo médio de atendimento de cada servente, de cada bancada e dos usuários; o tempo de espera médio dos usuários nas filas; a quantidade total consumida de cada ingrediente.

Procure simular o cenário do RU com variações que contemplem o amplo espectro do problema – o dia a dia.

### Dicas:

Defina macros para determinar os eventos de forma aleatória, como por exemplo: a quantidade máxima de usuários que chega por segundo ao RU, e então utilize números aleatórios (`rand()`) para determinar a quantidade de usuários que chega a um dado minuto. Um *plus* ao seu trabalho prático: tente fazer a simulação o mais próximo da realidade, neste caso, procure por funções de distribuições de probabilidade que se assemelham a chegada de alunos ao RU.



## O que deve ser entregue

- Código fonte do programa em C ou C++ (bem identada e comentada).
- Documentação do trabalho. Entre outras coisas, a documentação deve conter:
  1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados na implementação devem ser inseridos na documentação.
  3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementados e do programa como um todo (notação O).
  4. Listagem de testes executados: os testes executados devem ser simplesmente apresentados.
  5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização
  7. Em Latex: Caso o trabalho não seja elaborado/escrito em latex, perde-se 0,1 pontos.
  8. Impressão: Caso o trabalho não seja impresso usando modo frente-verso, perde-se 0,05 pontos.
  9. Formato: mandatoriamente em PDF (<http://www.pdf995.com/>).

Obs.: Veja modelo de como fazer o trabalho em latex:

<http://www.decom.ufop.br/prof/menotti/aedI092/tps/modelo.zip>

## Como deve ser feita a entrega:

A entrega DEVE ser feita via Moodle ([www.decom.ufop.br/moodle](http://www.decom.ufop.br/moodle)) na forma de um **único** arquivo zipado, contendo o código, os arquivos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho.

## Comentários Gerais:

- Clareza, identificação e comentários no programa também vão valer pontos;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero;
- Trabalhos entregues em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.