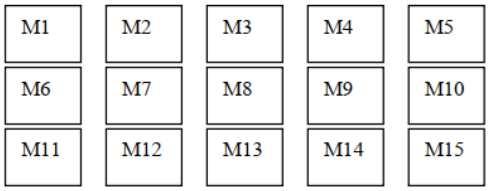
Trabalho 4: Sistema de Restaurante (Filas e Pilhas)

Crie um sistema para gerenciamento de um restaurante que satisfaça aos seguintes requisitos:

1. O restaurante possui *n* mesas de 4 lugares. Estas mesas são organizadas seguindo uma forma de matriz (linhas e colunas), e o usuário informa como é esta distribuição (por exemplo, observe a figura abaixo: há 15 mesas, e elas estão distribuídas em uma matriz de tamanho 3x5). Implemente as mesas como uma matriz de *structs*. Usuário informa número de linhas e de colunas, e consequentemente já se sabe quantas mesas o restaurante possui.
Exemplo de distribuição de mesas:



2. Quando chega um grupo de pessoas ao restaurante, é informado o número de pessoas e é atribuída uma mesa para tal grupo. Caso o grupo seja formado por mais de 4 pessoas, então eles são divididos em diversos subgrupos, mas não há garantia de que sentarão em mesas adjacentes.

3. Sobre cada mesa, guardar o número da mesa e se está livre ou ocupada; se estiver ocupada, guardar a quantidade de pessoas sentadas e o número da comanda. Cada mesa possui uma comanda.

4. Caso não existam mesas suficientes, deve ser formada uma fila de espera. O grupo que aguarda na fila recebe uma senha única. Pode haver grupos grandes, e neste caso, assim que liberar uma mesa, alguns membros do grupo podem conseguir a mesa e outros ainda ficarem na fila (em razão do tamanho da mesa). Grupos diferentes não compartilham mesas, mesmo que haja lugar vago em certa mesa (ex: se houver 2 grupos de 1 pessoa cada, estes 2 grupos ocuparão 2 mesas, e não compartilharão uma única mesa de 4 lugares). Grupos podem desistir de esperar por uma mesa e com isso liberar suas senhas de espera (sair da fila de espera).

5. Clientes podem finalizar a refeição e sair do restaurante, liberando as mesas; com isso, os clientes que por ventura aguardam na fila podem sentar-se (saindo da fila de espera).

6. Há uma pilha de pratos única que armazena os pratos limpos do restaurante. Sempre que uma mesa estiver vazia (ainda não ocupada ou recentemente liberada), um funcionário arruma a mesa (coloca pratos que são retirados da pilha). Como as mesas possuem 4 lugares, sempre são colocados 4 pratos. Caso um grupo com menos de 4 pessoas ocupe uma certa mesa, os pratos excedentes são retirados da mesa e recolocados na pilha de pratos. Um funcionário repõe pratos na pilha de tempos em tempos (não sendo necessário que a pilha esteja vazia para que novos pratos sejam adicionados).

Faça um menu que permita no mínimo as seguintes operações:

1) Chegar (grupo de) clientes ao restaurante (implica em ocupar mesa se há disponibilidade ou ir pra fila de espera)
2) Finalizar refeição/liberar mesa (liberar a mesa, chamar clientes da fila de espera (se houver), e arrumar mesa)
3) Desistir de esperar (sair da fila de espera)
4) Arrumar mesa (retirar pratos da pilha)
5) Repor pratos (adicionar pratos na pilha)
6) Imprimir pilha de pratos, fila de espera e ocupação das mesas, conforme descrito a seguir:

- Ocupação das mesas (número da mesa e quantidade de pessoas que ocupam a mesa) - o usuário pode pesquisar por número de mesa ou então consultar todas as mesas;
- Fila de espera (quantos grupos estão na fila de espera, e quantas pessoas aguardam na fila de espera. Por ex: grupo 1 aguarda por 3 lugares, grupo 2 aguarda por 5 lugares, logo há 2 grupos aguardando, e um total de 8 pessoas esperam na fila);
- Pilha de pratos (quantos pratos existem na pilha de pratos).

Entrega: por e-mail para deise@inf.ufsm.br os arquivos pertencentes ao projeto em questão, incluindo os *arquivos.c* e os *arquivos.h*. Não inclua os arquivos *.exe*.

Data da entrega: 30 de outubro

Observações para este trabalho:
• as *structs* devem sempre ser alocadas dinamicamente.
• o código deve ser organizado em diferentes arquivos e funções.
• deve ter um menu de opções que permita ao usuário escolher, a qualquer momento, qual ação deseja executar;
• trabalhos atrasados não serão corrigidos - se você não terminou tudo a tempo, entregue o que conseguiu fazer;
• formato da entrega: enviar os arquivos fonte .c e .h (não anexar arquivos executáveis!) para deise@inf.ufsm.br;
• **trabalho em grupos de 2 alunos**. Enviar um e-mail por dupla, identificando a disciplina, nome dos alunos e número do trabalho (trabalho 4, por exemplo);
• caso não receba confirmação do recebimento do e-mail em 24h, contate novamente o professor (reclamações posteriores sobre possíveis problemas no envio do e-mail não serão aceitas);
• cópias da internet e/ou colegas de outras duplas anulam a nota do trabalho de ambos os grupos;
• trabalhos com erros de compilação não serão corrigidos;
• professor usará o Dev-C ou Code Blocks para corrigir os trabalhos. Podem usar qualquer IDE para implementar o trabalho, mas certifiquem-se de que os arquivos estão compilando corretamente em alguma destas IDEs;
• obrigatoriamente a solução de vocês deve fazer uso de funções e passagem de parâmetros (eventuais respostas que pareçam corretas na execução, mas não usem funções e passagem de parâmetros, serão consideradas erradas);
• não devem ser usadas variáveis globais;
• a qualquer momento, vocês podem ser chamados para responder questionamentos sobre o trabalho entregue. Quando chamados, os alunos devem apresentar/responder aos questionamentos do professor, os quais farão parte da nota do trabalho;
• juntamente com o trabalho deve ser entregue um arquivo readme.txt que descreve sucintamente o que foi e o que não foi feito do enunciado do trabalho;
• peso do trabalho: verificar no site da disciplina em “Avaliação”.