T3 - Trabalho de C – CI1001 e CI067
Introdução aos Tipos Abstratos de Dados
Departamento de Informática/UFPR
Luiz Albini, Luis Bona, Marcos Castilho e André Grégio

A volta às aulas presenciais e a incerteza perante uma pandemia interminável e com o aparecimento de variantes consecutivas fez com que a administração do restaurante universitário (RU) tomasse algumas medidas visando a segurança das pessoas que ali se alimentam. Para tanto, estipulou-se que os frequentadores do RU estejam vacinados e façam uso de máscara. Entretanto, caso uma pessoa esteja vacinada e não tenha uma máscara, esta tem a possibilidade de comprar uma e voltar à fila para entrar no RU e comer um delicioso bandejão. Neste trabalho, você deve implementar o controle do ingresso no RU e sua contabilidade ao fim de um dia de serviço.

A seguir, especifica-se os requisitos de implementação dos tipos abstratos de dados a serem utilizados neste trabalho, a saber, duas pilhas, uma fila e uma lista.

Instruções:

1. Gere uma pilha de máscaras cuja quantidade deve ser definida por aleat(1,100) [1];
2. Gere uma pilha de refeições cuja quantidade deve ser definida por aleat(500,1000);
3. Defina e gere uma estrutura de dados para representar um conjunto de 1000 (mil) pessoas, sendo que cada pessoa deve ser gerada aleatoriamente e ter os seguintes atributos:
   - ticket que identifica a pessoa na fila (inteiro positivo não-repetido);
   - vacinado (0 para não, 1 para sim);
   - de máscara (0 para não, 1 para sim); e
   - dinheiro (sortear algumas pessoas para ter 1.30, outras para ter 3.80, com proporção de 60/100 para o primeiro caso).
4. Crie uma fila vazia de atendimentos (armazena os tickets das pessoas);
5. Crie uma lista ligada simples vazia de tickets não-atendidos.

[1] a função aleat(n,m) deve ser criada por você e gera números aleatórios entre n e m (ver `man 3 rand`)

Insira o ticket de todas as 1000 pessoas na fila. Após popular a fila, o "fiscal" da fila deve decidir o que fazer com cada pessoa de acordo com as condições abaixo:

- Se não estiver vacinada, ela é retirada da fila sem ser atendida e seu ticket é colocado em uma lista ligada simples de tickets não-utilizados com política de inserção ordenada ascendente.
- Se estiver vacinada, o fiscal verifica pela presença de máscara:
  - Se a pessoa estiver de máscara, ela "paga" 1.30 de seu dinheiro (se tiver saldo disponível) e "ganha" uma refeição da pilha.
  - Se a pessoa não estiver de máscara, ela é removida da fila, "paga" 2.50 da máscara (se tiver dinheiro suficiente e máscara disponível) e é reinserida no final da fila, **mantendo seu ticket**.
  - Caso não haja mais máscaras disponíveis, a pessoa deve colocar seu ticket na lista de tickets não-utilizados (a mesma dos não vacinados) e ir embora sem comer.

O programa finaliza quando se acabarem as refeições. Após a última refeição ser dada, todas as pessoas da fila são dispensadas (destruir fila e liberar memória) e é mostrada a contabilidade do dia:

- Quanto de dinheiro foi arrecadado no total;
- Quanto de dinheiro foi para refeições;
- Quanto de dinheiro rendeu a venda de máscaras;
- Quantos e quais (imprima a lista de tickets) foram os tickets não utilizados e por que razão.

Use uma struct para armazenar estas informações ao longo do seu programa.

Neste trabalho você deve usar as bibliotecas desenvolvidas no T2, isto é, os TADs lista, pilha e fila.

Você deve ter implementado os respectivos arquivos `liblista.c`, `libpilha.c` e `libfila.c` quando fez seu T2, e caso queira ou precise, pode modificar apenas os arquivos `.c`.

Nos arquivos `.c` você pode implementar funções que não estão nos arquivos `.h` caso queira ou caso precise. Elas serão consideradas como funções internas.

Você deve também produzir um programa em C chamado `ru_na_pandemia.c` que implemente corretamente o enunciado deste trabalho (acima).

Observe que você **não pode** alterar os arquivos `.h` fornecidos!

Importante também observar a proibição de violação dos Tipos Abstratos de Dados! Assim, no seu arquivo `ru_na_pandemia.c` você **não pode** acessar diretamente as estruturas da pilha, fila ou lista a não ser pelo uso das funções que estão disponíveis nos seus `.h`.

Observações:

- Todos os tipos abstratos de dados devem ser alocados dinamicamente e obedecer às mesmas especificações do T2
- As boas práticas de programação são fundamentais neste trabalho;
- As bibliotecas `.h` fornecidas via Moodle não podem ser alteradas.

Como entregar? Crie um diretório de nome "t3" e coloque todos os seus arquivos do trabalho 3 neste diretório.

Quando for entregar o trabalho, vá para o diretório 'pai' (isto é, o anterior, resultado de fazer `cd ..`) do diretório t3 e rode nele o comando `tar`, assim:

```
tar zcvf t3.tar.gz t3
```

Deste modo, quando os professores abrirem o `tar`, o próprio `tar` vai criar o diretório t3 e seus arquivos dentro dele.

Juntamente com seus arquivos de código, entregue um arquivo de nome `Makefile` que contenha todos os comandos necessários para compilar e executar seus programas de teste, que terão necessáriamente os nomes acima definidos.

Seu arquivo `.tar.gz` deve portanto conter, pelo menos:

- `liblista.{c,h}`
- `libpilha.{c,h}`
- `libfila.{c,h}`
- `Makefile`
- `ru_na_pandemia.c`
- Algum outro arquivo que você ache necessário implementar adicionalmente.

Seu programa deve ser executado assim:

```
./ru_na_pandemia
```

Note que os professores, ao corrigirem o trabalho executarão:

```
make
```

sem mais nenhuma outra palavra adicional e portanto seu executável deverá estar disponível desta maneira após o comando `make` ter sido executado.

Seu programa deve compilar sem erros com as flags `-std=C90 -Wall` do compilador `gcc`.