

Modalidade: em grupo (no máximo 3 alunos por grupo)

Data de entrega: 30/novembro até 23:59 na tarefa do Microsoft Teams. Entregar UM ÚNICO ARQUIVO COMPACTADO .ZIP CONTENDO APENAS OS ARQUIVOS FONTES e o relatório em PDF. São descontados 25% da nota por dia de atraso.

Fazer um programa em linguagem C para gerenciar o sistema de pedidos de *fast food* do tipo *drive through*, chamado **BurgerTI**, conforme descrito a seguir:

- existem vários tipos de sanduíches cadastrados no **BurgerTI**, contendo as seguintes informações:
 - código: número inteiro que identifica univocamente cada tipo de sanduíche
 - nome: nome do sanduíche, tendo no máximo 50 caracteres
 - descrição: um texto descrevendo brevemente a composição do sanduíche, tendo no máximo 200 caracteres
 - disponível: indicação se há disponibilidade do sanduíche no momento atual
 - preços: um preço em reais para cada tamanho, existindo os tamanhos pequeno, médio e grande.
- existem vários tipos de bebidas cadastradas no **BurgerTI**, contendo a seguintes informações:
 - código: número inteiro que identifica univocamente cada tipo de bebida
 - nome: nome da bebida, tendo no máximo 50 caracteres
 - disponível: indicação se há disponibilidade da bebida no momento atual
 - preços: um preço em reais para cada tamanho, existindo os tamanhos pequena, média e grande.
- existem vários tipos de acompanhamentos extras cadastrados no **BurgerTI**, que podem ser pedidos juntos com o sanduíches:
 - código: número inteiro que identifica univocamente o extra
 - nome: nome do extra, tendo no máximo 50 caracteres
 - disponível: indicação se há disponibilidade do extra no momento atual
 - preço: preço em reais do extra
- existem vários tipos de sobremesas cadastradas no **BurgerTI**, contendo as seguinte informações:
 - código: número inteiro que identifica univocamente a sobremesa
 - nome: nome da sobremesa, tendo no máximo 50 caracteres
 - disponível: indicação se há disponibilidade da sobremesa no momento atual
 - preço: preço em reais da sobremesa
- o sistema **BurgerTI** registra cada pedido de cliente, contendo as seguintes informações:
 - código do pedido: número inteiro que identifica univocamente o pedido
 - CPF: 11 dígitos contendo o CPF do cliente
 - lista de itens do pedido: sanduíches, sobremesas e bebidas com as respectivas quantidades e tamanhos
 - total: valor total em reais

O sistema **BurgerTI** deve ter as seguintes funcionalidades implementadas, organizadas convenientemente em menus:

- cadastrar sanduíche

- cadastrar bebida
- cadastrar sobremesa
- cadastrar extra
- imprimir lista de sanduíches
- imprimir lista de bebidas
- imprimir lista de sobremesas
- imprimir lista de extras
- registrar novo pedido: insere um novo pedido na fila de pedidos
- imprimir pedido: dado o código do pedido, imprime as informações do pedido
- imprimir lista de pedidos atendidos: imprime a lista com todos os pedidos já atendidos

Toda a interface do programa deve ser apresentada em modo texto, sendo executado no *prompt* do *shell* do sistema operacional.

O registro de um novo pedido deve ser feito a partir do CPF do cliente, sendo que devem ser exibidos até os 3 últimos pedidos do cliente, permitindo-o escolher repetir um mesmo pedido entre estes 3, ou ainda realizar um pedido diferente dos 3. Em ambos os casos, o novo pedido deve ser incluído na fila de pedidos não atendidos (descrita mais adiante).

O sistema **BurgerTI** também registra a fila de pedidos ainda não atendidos, sendo possível realizar as seguintes operações, apresentadas convenientemente em menu:

- atender pedido: atende o primeiro pedido retirando-o da fila e incluindo-o na lista de pedidos atendidos.
- desistir de pedido: retira um pedido da fila, mediante a desistência do cliente, realizada a partir do CPF.
- imprimir fila: imprime a fila de pedidos ainda não atendidos no momento.

O sistema **BurgerTI** ainda deve permitir a realização de operações de inclusões em lote, lidas de um arquivo texto, conforme sintaxe abaixo:

< tipo – item >; < codigo >; < campo1 >; < campo2 >; ... < campoN >
onde:

- *< tipo – item >*: pode ser SD (sanduíche), BB (bebida), EX (extra), SM (sobremesa) ou PD (pedido)
- *< campo1 >; < campo2 >; ... < campoN >* correspondem aos campos com informações referentes ao respectivo item.

Exemplo de conteúdo de arquivo .txt contendo lote de operações a serem executadas:

```
SD;1;X-tudo; hambúrguer, ovo, pão, bacon, alface, tomate;D;12,90;17,90;21,90
SD;2;X-salada; ovo, pão, queijo, alface, tomate;D;9,90;12,90;17,90
SD;3;X-duplo;2 hambúrgueres, ovo, pão, bacon, queijo, alface, tomate;D;15,90;20,90;25,90
SD;4;X-light; pão, queijo, alface, tomate;D;15,90;20,90;25,90
SD;5;X-vegano; pão, proteína de soja, queijo vegano, alface, tomate;I;20,90;23,90;27,90
BB;1;coca-cola;D;3,00;5,00;7,00
BB;2;suco de laranja;D;5,00;7,00;9,00
BB;3;guaraná;D;3,00;5,00;7,00
EX;1;maionese;D;2,00
EX;2;cebola;D;3,00
EX;3;molho barbecue;D;3,00
EX;4;batata frita;D;5,00
SM;1;sorvete;D;10,00
SM;2;doce de leite;D;5,00
```

```
SM;3;musse de chocolate;D;7,00
PD;1;11111111111;(SD,1,2,M);(SD,2,1,P);(BB,2,3,M);(EX,3,4);(SM,2,2)
PD;2;11111111111;(SD,3,1,G);(BB,1,1,G);(EX,1,1);(SM,1,1)
PD;3;11111111111;(SD,2,3,P);(SD,4,1,M);(BB,1,1,M);(BB,2,1,P)
PD;4;22222222222;(SD,4,3,M);(BB,1,2,P);(BB,2,1,G);(SM,3,1)
PD;5;33333333333;(BB,1,2,M);(EX,4,2)
```

Assim, a linha

SD;1;X-tudo; hambúrguer, ovo, pão, bacon, alface, tomate;D;12,90;17,90;21,90
é interpretada como:

- o código do sanduíche é 1
- o nome é X-tudo
- a descrição é hambúrguer, ovo, pão, bacon, alface, tomate
- está disponível (D)
- os preços dos tamanhos pequeno, médio e grande são respectivamente 12,90 ,17,90 e 21,90 reais.

No caso de pedidos, a interpretação de uma linha, por exemplo, o seguinte conteúdo:

PD;1;11111111111;(SD,1,2,M);(SD,2,1,P);(BB,2,3,M);(EX,3,4);(SM,2,2)
significa:

- o código do pedido é 1
- o CPF do cliente é 11111111111
- compõem o pedido: 2 X-tudo de tamanho médio (M), 1 X-salada de tamanho pequeno (P), 3 sucos de laranja de tamanho médio (M), 4 molhos barbacue, e 2 doces de leite de sobremesa

O sistema **BurgerTI** deve armazenar todas as informações em arquivo(s) binário(s).

Entregar um relatório em PDF com as seguintes informações:

- nomes dos componentes do grupo
- opções implementadas, seguidas de uma breve explicação para o usuário do programa, ao modo de um arquivo *readme*

Critérios de avaliação:

- o trabalho deverá ser apresentado oralmente em grupo, seguido de arguição sobre o código/teoria relacionada.
 - a nota referente à apresentação oral é individual conforme desempenho do aluno na arguição.
 - o aluno do grupo que não esteja presente durante a apresentação oral, não terá a nota correspondente.
- documentação/organização do código-fonte: peso 1
- relatório PDF: peso 1
- apresentação oral: peso 1
- implementação das funcionalidades: peso 7

Em todas as funcionalidades, a avaliação será feita considerando corretude e qualidade do código. Trabalhos copiados ou plagiados receberão nota ZERO!