RELATÓRIO DO TRABALHO PRÁTICO

ALUNO : VITOR DE CARVALHO SILVA (12121BSI263)

1 - INTERPRETAÇÃO

Ao falar sobre o trabalho, julgo importante primeiro falar sobre a minha interpretação e ponto de vista em relação ao texto base que me foi enviado.

O texto base da a introdução do problema onde cita um sistema de validação de comandas para um restaurante, assim como a distribuição de um brinde para o cliente. Nas instruções citam que são necessários uma fila de clientes e uma pilha contendo os brindes. A fila deve ser invariavelmente grande já a pilha, interpretei como se seu tamanho ficasse ao meu critério.

2 - IMPLEMENTAÇÕES

2.1 FILA

A fila foi implementada de maneira estática, sem muitas diferenças em relação ao que foi visto em sala de aula. A única alteração feita foi em relação a alocação, onde, no lugar de alocar um tamanho MAX, como era de costume, usei uma função para me gerar um número aleatório e alocar esse número gerado ao tamanho da fila. Na função main, a fila é cheia de 1 até ‘n’ termos, e o número do cliente é exibido na tela. No programa final deixei esse valor aleatório sendo entre 0 e 25, mas pode ser facilmente alterado na biblioteca “Fila.c”.

2.2 PILHA

A pilha foi implementada também de maneira estática, utilizando apenas as funções necessárias para o funcionamento do programa. Não houve alterações em relação ao que foi visto em sala. Como foi dito antes, julguei que o tamanho da pilha ficaria por conta do programador, então ela está inicialmente alocada com 10 itens, também aleatórios. O sistema de aleatoriedade de itens será mais bem explicado no tópico 2.3. A pilha de chocolates possui três opções de doces: Sonho de Valsa, Diamante Negro e Bis. É distribuído um doce para cada cliente, e, quando a pilha está vazia, ela se enche novamente de maneira aleatória.

2.3 COMANDA

O sistema de comanda foi a primeira parte a ser implementada no programa. Nela, foi trabalhado uma maneira mais ‘randômica’ para se aproximar mais do real. Tendo do ponto de vista que quem está gerenciando o programa é um atendente do restaurante, para ele aparecerá uma comanda aleatória, com itens e valores totalmente diversos.

Assim, adotei um sistema que gera um tamanho aleatório para a comanda, inicialmente de 0 a 15, e alocar esse tamanho de maneira dinâmica a um ponteiro dentro da estrutura da comanda, criando assim uma comanda que pode ter um tamanho invariavelmente grande, assim como a fila.

Para distribuir os produtos usei o mesmo sistema, um número aleatório é gerado dentro de um laço, e, cada número é relacionado com um elemento do menu, até que a comanda esteja cheia. Ao mesmo tempo, a soma dos valores já é feita. A pilha funciona do mesmo modo, uma estrutura que contêm todos os itens se relaciona com o número desempilhado.

Logo, a exibição é a seguinte:

2.4 MENU

O menu nada mais é do que uma estrutura, simples.

```
struct menu
{
    int id;
    char nome[30];
    float preco;
};
```

Porém é de extrema utilidade, pois manipular uma estrutura dessa maneira é muito prático. No programa foi utilizado um vetor de estruturas, e cada posição desse vetor equivale a um item, com nome, número de identificação e o preço.

```
struct menu item_menu[5] =
{
    {0,"Pizza",10.50},
    {1,"Hamburguer",12.30},
    {2,"Refrigerante",3.99},
    {3,"Sorvete",5.80},
    {4,"Cerveja",8.40}
};
```

Utilizei apenas 5 itens, mas é possível alterar para utilizar quantos itens forem necessários. A estrutura dos chocolates da pilha trabalha da mesma maneira um vetor de estruturas contendo o nome e a identificação do chocolate.

Em termos gerais, um laço percorre todas suas posições e uma variável recebe o valor de cada item, esse item é relacionado com a posição no vetor de estruturas do menu, se relacionando com o id, nome e preço. A pilha funciona da mesma forma, mas faz isso ao ser desempilhada.

3 – VISÃO GERAL

Em termos gerais, a experiencia de produção foi ótima. Houve algumas frustrações quanto a compilação em determinados momentos, mas bastante coisa foi aproveitada do trabalho. O uso de um vetor de estruturas e a criação de bibliotecas foram coisas que nunca havia testado.

Em relação aos resultados da produção, tudo ocorreu da maneira desejada. Poucas coisas correram fora do plano.

O programa foi implementado usando Code::Blocks e, como foi solicitado, tem um repositório no github com todas as bibliotecas e códigos utilizados.