

```

/*=====*/
/*==          Trabalho 1          ==*/
/*==          Programação 1        ==*/
/*=====*/
/*== Autores: Nicolas A. Hilbert & Bruno Leal Maciel ==*/
/*== Data: 13/07/22                ==*/
/*== Turma: INFO 62B               ==*/
/*=====*/

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <Windows.h>

int main()
{
    // Troca a cor do CMD para fundo roxo e letras amarelas
    system("color 5e");

    // Troca a linguagem de execução do programa para PT-BR
    setlocale(LC_ALL, "Portuguese");

    int flagPedindo=1; // Flag para permitir a execução do programa
    int encerrarPrograma(float); // Função para encerrar o programa
    int encerrarPedido(float *, int); // Função para encerrar o pedido
    int quantPedidos=0; // Quantidade total de produtos no pedido
    float total=0.0; // Quantidade total do valor dos produtos no pedido
    float fazerPedido(int *, float *); // Função para fazer um novo pedido
    char codOpcao; // Variável que recebe a opção escolhida pelo usuário no menu

    // Loop principal do programa, que executará enquanto a flagPedindo for igual a 1
    while(flagPedindo){
        system("cls"); // Limpa a tela

        // Menu de opções do programa
        printf("=== MENU IGREBOYS ===\n");
        printf("\t[N] Novo Pedido\n");
        printf("\t[E] Encerrar Pedido\n");
        printf("\t[S] Sair do Programa\n");
        printf("\t-> Informe sua opção: ");

        // Recebe a opção do usuário e a atribui para a variável codOpcao
        fflush(stdin);
        scanf("%c", &codOpcao);

        // Switch que decide o que fazer com a opção do usuário no menu
        switch(codOpcao){
            case 'n':
            case 'N': fazerPedido(&quantPedidos, &total); break; // Fazer um novo pedido, atribuindo o endereço
das variáveis total e quantPedidos
            case 'e':
            case 'E': encerrarPedido(&total, quantPedidos); break; // Encerra o pedido atual, atribuindo à
função o endereço da variável total e o valor da variável quantPedidos
            case 's':
            case 'S': flagPedindo = encerrarPrograma(total); break; // Encerra o programa dependendo do retorno
da função encerrarPrograma, atribui à função a variável total
            default: printf("\nDigite uma opção válida!!!\n"); Sleep(1500); system("cls"); break; // Opção
inválida, pede para o usuário redigitar
        }
    }

    return 0;
}

float fazerPedido(int *quantPedido, float *total) {
    // Função para realizar um novo pedido
    // Recebe do função main os endereços das variáveis quantPedido (int)

```

```

// E da variável total (float)

int codProduto; // Variável onde é atribuído o código do produto
int quantProduto=0; // Variável que armazena a quantidade de produtos neste pedido
float totalPedido=0.0; // Variável que armazena o valor total deste pedido

system("cls"); // Limpa a tela

// Cardápio do programa
printf("*****\n");
printf("==          MENU IGREBOYS          ==*\n");
printf("==          CARDÁPIO          ==*\n");
printf("*****\n");
printf("== Cód.      Produto      Preço ==*\n");
printf("== 100      Pastel      R$3,50 ==*\n");
printf("== 101      Torrada      R$4,00 ==*\n");
printf("== 102      Enroladinho    R$1,50 ==*\n");
printf("== 103      Suco 500ml    R$3,50 ==*\n");
printf("== 104      Refrigerante Lata R$3,00 ==*\n");
printf("*****\n");

// Loop que pede o código do produto até que um válido (ou 0) seja digitado
do {
    // Recebe o código do produto e armazena na variável codProduto
    printf("\n-> Informe o código de seu produto (0 para cancelar): ");
    fflush(stdin);
    scanf("%d", &codProduto);

    // Se o código for inválido, escreve na tela "Digite um código válido!"
    if(codProduto!=0&&(codProduto<100||codProduto>104))
        printf("\nDigite um código válido!\n");
} while(codProduto!=0&&(codProduto<100||codProduto>104));

// Caso o código de produto selecionado não seja 0
// Este loop irá pedir a quantidade do produto desejado (até que seja válida / maior ou igual a 1)
if(codProduto!=0){
    do {
        printf("\n-> Informe a quantidade do produto: ");
        fflush(stdin);
        scanf("%d", &quantProduto);

        // Se a quantidade for inválida (menor que 1), escreve na tela "Digite uma quantidade válida!"
        if(quantProduto < 1)
            printf("\nDigite uma quantidade válida!\n");
    } while(quantProduto < 1);
}

// Switch que decide o que fazer com o código do produto selecionado
// Caso o código for 0, o pedido é cancelado
// Se não, a variável total pedido receberá o valor do produto vezes sua quantidade
// Caso algum código inválido chegue até aqui, o programa escreverá "UM ERRO OCORREU!" e cancelará o pedido
switch(codProduto){
    case 0: printf("\n\nPedido cancelado."); Sleep(1500); return 0;
    case 100: totalPedido = 3.5*quantProduto; break;
    case 101: totalPedido = 4.0*quantProduto; break;
    case 102: totalPedido = 1.5*quantProduto; break;
    case 103: totalPedido = 3.5*quantProduto; break;
    case 104: totalPedido = 3.0*quantProduto; break;
    default: system("cls"); printf("UM ERRO OCORREU!"); Sleep(1500); return 0;
}

system("cls"); // Limpa a tela

// Escreve na tela o código do produto, sua quantidade, e o valor total do pedido
printf("Código: %d\nQuantidade: %d\nTotal deste pedido: R$%.2f", codProduto, quantProduto, totalPedido);

```

```

// Soma à variável "total" da função main o valor total deste pedido
*total += totalPedido;

// Soma à variável "quantPedido" da função main a quantidade de produtos deste pedido
*quantPedido += quantProduto;

// Dorme por 1 segundo e encerra a função
Sleep(1000);
return 0;
}

int encerrarPrograma(float total){
    // Função para encerrar o programa
    // Recebe o valor da variável total da função main (float)

    // Caso o total for maior que 0
    if(total>0.0){
        char confirmarSaida; // Variável para armazenar a opção de saída do usuário

        system("cls"); // Limpa a tela

        // Loop infinito até que haja um return
        while(1) {
            // Pergunta ao usuário se quer finalizar ou não o pedido em andamento
            printf("== Você não finalizou o pedido ==*\n\n");
            printf("Quer continuar?\n");
            printf("\t[S] Sim\n");
            printf("\t[N] Não\n");

            // Armazena a opção na variável confirmarSaida
            printf("\t-> Informe sua opção: ");
            fflush(stdin);
            scanf("%c",&confirmarSaida);

            // Switch que escolhe o que fazer com a opção selecionada
            switch(confirmarSaida){
                case 'S':
                case 's': return 1; // Continua o pedido de onde parou, retornando 1 como seu resultado
                case 'n':
                case 'N': system("cls"); printf("Volte sempre!\n"); return 0; // Encerra o programa, retornando
0
                default: printf("\nDigite uma opção válida!!!\n"); Sleep(1500); system("cls"); break; // pede
para que uma opção válida seja digitada
            }
        }
    }

    // Caso o total não for maior que 0
    else{
        system("cls"); // Limpa a tela
        printf("Volte sempre!\n"); // Escreve "Volte Sempre"

        return 0; // Encerra o programa, retornando 0
    }
}

int encerrarPedido(float *total, int quantPedidos){
    // Variável para encerrar o pedido atual
    // Recebe o endereço da variável total da função main (float)
    // E o valor da variável quantPedidos da função main (int)

    system("cls"); // Limpa a tela

    // Caso o valor total for maior que 0
    if (*total>0.0){
        // Escreve na tela o valor total do pedido, a quantidade de produtos e a confirmação do pagamento
        printf("Total do pedido: R$%.2f\nQuantidade de produtos: %d\n\n", *total, quantPedidos);
    }
}

```

```
printf("Você pagou com sucesso!");

// Zera o valor total após o pagamento
*total=0.0;
}
// Caso o valor não for maior que 0
else
    // Informa o usuário de que não há nenhum pedido em andamento
    printf("Não há um pedido em andamento!");

// Dorme por um segundo e retorna 0 para encerrar a função
Sleep(1000);
return 0;
}
```