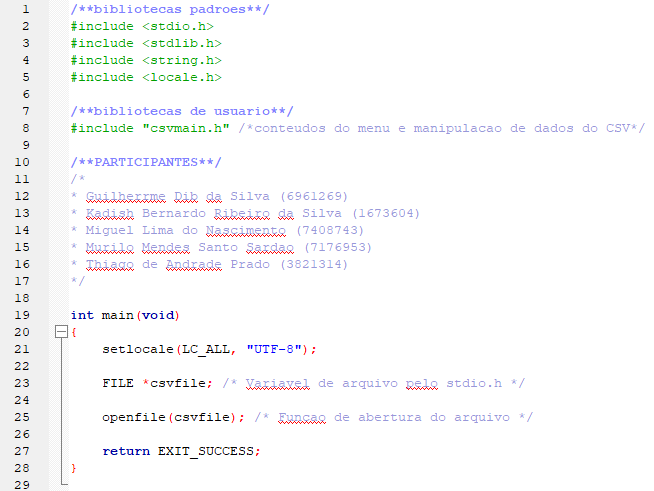
**Atividade Prática - Programação de computadores**

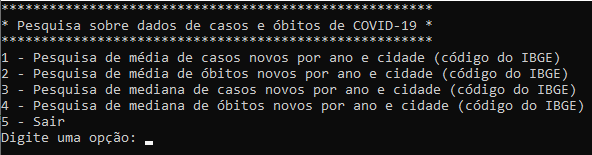
O Código contém 3 principais componentes modularizados, o arquivo **main.c** (código-fonte geral do projeto), **csvmain.c** (código do cabeçalho **csvmain.h**, que contém as funções do projeto), e **menu.c** (que contém exclusivamente a função do menu ligados novamente a header).

## **main.c**

Consta apenas com as informações e menções mais essenciais dos projeto, como a variável ponteiro de arquivo (**FILE**) **\*csvfile**, já sendo usado como parâmetro da função openfile (declarada na header csvmain.h) com o retorno EXIT\_SUCCESS retornando o valor inteiro 0.

## **menu.c**

Após o retorno positivo da abertura do arquivo (na função openfile), direciona ao menu que oferece as opções de filtragem e manipulação dos dados de acordo com a entrada do usuário, e assim executando a função correspondente até que o usuário digite 5 para finalizar a execução do programa, usa de *setlocale* da biblioteca *<locale.h>* para acentuação da interface e um código simples de impressão seguido de uma entrada em número inteiro e uma estrutura de escolha *switch-case* para cada opção, tudo em um loop *do-while* para reiniciar a função em caso de invalidez nas entradas do código mais para frente.

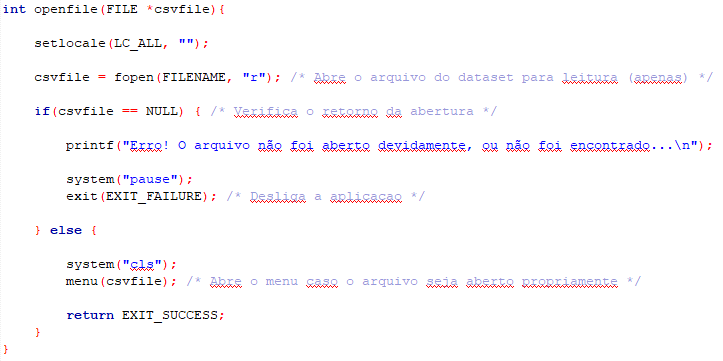


## 

## **csvmain.c**

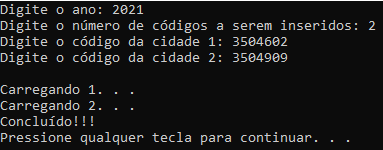
Executa todas as principais funções do código, ligada a main pelo cabeçalho **csvmain.h**, com as funções **openfile**, **case1**, **case2**, **case3**, **case4**, **resetfunction**, e **cmpfunc**, respectivamente.

A função openfile que tem como parâmetro o arquivo *\*csvfile*, usando setlocale para acentuação, *fopen* (da biblioteca *<stdio.h>*) para abertura inicial do arquivo em modo de leitura “r” (reading), e depois a verificação com *if-else* do retorno da função, no caso da leitura retorna um valor nulo (ou seja, erro na leitura), o usuário é notificado e a função retorna 1 como EXIT\_FAILURE, em caso de retorno não nulo, a tela do prompt e limpa com *system(“cls”)* e redireciona o usuário ao menu.



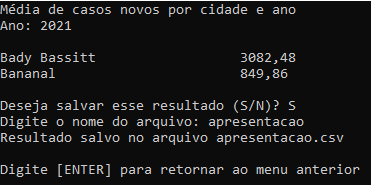
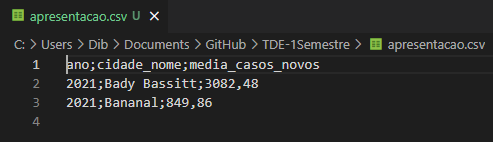
As funções de caso, case1, case2, case3, case4, seguem uma estrutura similar na base, alterando as funções apenas após a captura e filtragem dos dados, onde as funções **1** e **2**, fazem a média das linhas extraídas, enquanto as funções **3** e **4** armazenam os dados das linhas em uma matriz, para ordenar e de acordo com o número de números no vetor, dar como resultado a mediana.

As estruturas nessas funções se iniciam com primeiramente, a função setlocale para acentuação das saídas do código, após isso, a declaração das variáveis mais essenciais, e a entrada das opções de filtragem, sendo essas, o ano, o número de códigos (ou casos) a serem executados, e os códigos de cidade, todos com suas respectivas verificações em caso de entradas inválidas usando de estruturas *if-else* para retornar notificações de falhas e reiniciar a função para que haja a entrada novamente, a única peculiaridade seria na verificação final, dos código, onde foi usado primeiramente, um loop para cada código inserido com uma verificação individual (possibilitando o usuário uma maior precisão de identificar possíveis falhas nas entradas), onde foi usado um arquivo terceiro (*codigoibgever.txt*) aberto em modo de leitura com *openfile()* para verificar a validade dos código usando a função *strstr* (da biblioteca *<string.h>*) verificando linha por linha se o conteúdo da entrada do usuário existe em algum lugar no arquivo, se sim, um valor **verdadeiro** (**1**) e armazenado na variável de verificação do caso, caso não, o valor retornado é **falso** (**0**) e mais tarde na verificação das variáveis de validade dos códigos retorna um erro, reiniciando a função. No caso verdadeiro, o código continua normalmente e o nome da cidade (que também consta no arquivo) é armazenado na matriz *citycodename[i]* com a função *strcpy* e *strtok* para identificar até onde a leitura vai (e assim não copiar o código junto ao nome), e fechado o arquivo de verificação ao final.



Após a validação de todas as entradas, o programa procede em um loop (para cada caso inserida na segunda entrada na variável inteira **n**), primeiramente extraindo o cabeçalho do csv (por fim de evitar erros) e depois escaneando linha por linha até o fim do arquivo **EOF** (verificado por um *if* ao fim do *while* que verifica o retorno **NULL** da função *getline*), usando a função *strstr* para verificar se o código anteriormente digitado existe nas linhas, e assim separando as linhas que constam o código, dividindo as em string com *strtok* (usando um loop *for*) até a coluna desejada e realizando a ação condizente a escolha do usuário no menu (no caso da média acumulando o valor, e no caso da mediana armazenando os valores individualmente em uma matriz), e fechando o arquivo a fim de cada ciclo para cada código inserido.

Após a captura de todos os dados (usando da função *atof(val)* para transformar as strings em valores reais float), imprime a mensagem **“Concluído!!!"**, limpando a tela e pausando a execução, e preparando as strings e variáveis para impressão final, onde a formatação usada antes na verificação e retirada para imprimir os dados, mostrando-os e oferecendo a opção de salvar, ou não, os dados separados, caso a entrada seja ‘S’, abre um novo arquivo com modo de escritura **“w”** (write), imprimindo todas as informações (em formato CSV com separador **“;”**) com *fprintf()*, e depois fechando o arquivo para renomear de acordo com a entrada do usuário e depois reiniciando as funções voltando para o menu.



No caso das funções 1 e 2, antes da impressão, tanto para o usuário quanto para o arquivo, é feita a média dos valores acumulados e contador de linhas com o código da cidade, para que a média fosse armazenada no vetor *result[i]*. No caso das funções 3 e 4, antes da impressão, é feita a ordenação da matriz dos valores e a mediana do vetor do caso atual é armazenada na variável *result[i]*, para que a função de ordenação qsort seja executada, é necessário também a função *cmpfunc*, que captura a posição atual do vetor com a próxima para retornar um valor <0, caso **a** seja menor que **b**, 0 se forem iguais, e >0 caso **a** seja maior.

Após toda a execução, o retorno vai para o início da função *menu()*, assim sendo executada indefinitivamente até que o usuário digite a opção de saída (5) ou até que haja um erro na leitura dos arquivos csv. 