

Universidade Federal de Viçosa

CCF110- Curso de Programação

Projeto em dupla de programação

Participantes:

Nomes:

Jordane Andrade Soares n° 5106

Vitor Flávio Macedo Silva n° 4913

SUMÁRIO

Introdução.....	3
Explicação do código.....	4
Conclusão.....	14

INTRODUÇÃO

O projeto é basicamente um programa para treinar as operações matemáticas, ele é composto por três funcionalidades diferentes, a primeira é a tabuada, a segunda é calculadora composta por operações básicas e operações com matrizes, e por fim o mini game responsável por colocar em prática todo o conhecimento adquirido com as outras funcionalidades.

DESENVOLVIMENTO

Será feita uma explicação das etapas principais do código.

```
float soma(float valor1, float valor2);  
float subtracao(float valor1, float valor2);  
float multiplicacao(float valor1, float valor2);  
float divisao(float valor1, float valor2);  
int potencia(float valor1, float valor2);  
void operacoesMatrizes(char tipoMat, int linhaA, int colunaA, int linhaB, int colunaB, float matrizA[][colunaA], float matrizB[][colunaB]);
```

Declaração das funções utilizadas no código.

```
void main()  
{  
    // Variaveis  
    char nome[99];  
    int opcao = 1, i;  
  
    FILE *arquivo;  
    arquivo = fopen("pontuacao.txt", "a+");  
  
    FILE *historicoTab;  
    historicoTab = fopen("historico.txt", "a+");  
  
    // Interface  
    printf("=====  
    printf(".....PROGRAMA DE OPERAÇÕES....  
    printf(".....BEM VINDO.....  
    printf("=====  
    printf("\n\n\n");
```

Função principal onde fica a interface do usuário com a entrada dos dados. Na imagem apresenta a criação dos dois arquivos utilizados no programa, o FILE arquivo apresenta salva os dados e a pontuação do usuário no mini game e o FILE historicoTab salva o histórico de pesquisa da calculadora.

```
// Gerencia as opções
while (opcao != 0)
{
    int opcao2 = 1;

    if (opcao == 1)
    {
```

Esse while é responsável por gerenciar as 3 opções do programa mostrado abaixo:

```
1.Tabuada
2.Calculadora
3.Mini Game
0.Sair

=====
█
```

O primeiro if é responsável pelas operações da calculadora e as variáveis referente ao mesmo é apresentada nesse if, como mostrado abaixo:

```
if (opcao == 1)
{
    // TABUADA...

    while (opcao2 == 1)
    {

        int valorTabuada, tabuada, opcao2 = 1;
        float resultadoTab;
        float valor1 = 0, valor2 = 0;
        char tipoTab;

        // Interface da tabuada
        printf("#####\n");
        printf("Escolha o tipo de operação:\nOBS:\n+ --> Soma\n- --> ");
        scanf(" %[^\n]c", &tipoTab);
        printf("Informe ate qual valor vc quer a execução da tabuada ");
        scanf("%d", &valorTabuada);
        printf("Informe a tabuada:\n");
        printf("#####\n");
        scanf("%d", &tabuada);
        system("clear");
```

O while opcao2, aparece nas 3 etapas do programa, pois é responsável por da a opção de escolha do usuário de manter ou não nessa etapa como mostra abaixo:

```
.....
Vitor deseja informar uma nova tabuada?:
Pra sim digite 1.
Pra sair digite qualquer outro valor
.....
█
```

Esse for é responsável por analisar o tipo de operação que será realizada que sera avaliado pelo switch.

```
printf("_____TABUADA DA %c DO %d_____")

for (i = 0; i <= valorTabuada; i++)
{
    valor1 = i;
    valor2 = tabuada;

    switch (tipoTab)
    {

        case '+':

            resultadoTab = soma(valor1, valor2);

            break;
```

Quando o tipo de operação for escolhido o programa chama a respectiva função

```
float soma(float valor1, float valor2)
{
    return valor1 + valor2;
}

float subtracao(float valor1, float va
{
    return valor1 - valor2;
}

float multiplicacao(float valor1, floa
{
    return valor1 * valor2;
}
```

que ira realizar a operação e retornar o respectivo valor.

OBS: Essas função são reutilizadas nas 3 funcionalidades do programa.

Depois que a função retorna o valor o for que é responsável por imprimir a tabuada

```
printf("|\\t");  
printf(" %.0f %c %.0f = %.0f", valor1, tipoTab, valor2, resultadoTab);  
printf("\\t|\\n");
```

imprime os valores, até o valor em que o usuário pediu para a tabuada ser impressa.

O segundo if é da calculadora e nele apresenta as variáveis da calculadora é na calculadora é possível realizar dois tipos de operações.

```
if (opcao == 2)  
{  
    // CALCULADORA...  
  
    while (opcao2 == 1)  
    {  
  
        int tipoOperacao = 0, opcao2 = 1;  
        char tipoCal;  
  
        printf("#####  
        printf("Escolha a opção abaixo\\n");  
        printf("1.Operações básicas\\n2.Operações  
        printf("#####  
        scanf("%d", &tipoOperacao);  
        system("clear");  
  
        if (tipoOperacao == 1)  
        {
```

Operações básica:

```

printf("-->");
scanf("%f", &valor1);
printf("%c\n", tipoCal);
printf("-->");
scanf("%f", &valor2);
printf("\n");

switch (tipoCal)
{
case '+':

    resultadoCal = soma(valor1, valor2);

    break;
case '-':

    resultadoCal = subtracao(valor1, valor2);

```

Em que não é necessário explicar como funciona, pois é parecido com o caso da tabuada, pois acessam as mesma função.

E o segundo tipo de operações que a calculadora realiza é com a matrizes:

```

if (tipoOperacao == 2)
{
    char tipoMat;

    printf("#####");
    printf("Escolha o tipo de operação:\nOBS:\n+ --");
    printf("#####");
    scanf("%[^\\n]c", &tipoMat);
    system("clear");

    int linhaA, colunaA, linhaB, colunaB, i, j, k;

```

Nesse if a pessoa informa o tipo de operação que será realizado, e informa o tamanho da operação, e preenche a matriz correspondente.

Em seguida o programa chama a matriz que irá realizar a operação correspondente, diferente das outras funções essa é a única em que não apresenta retorno, pois achei mais fácil fazer todas as operações de matriz na mesma função, já que nesse caso não tem como utilizar as outras funções para a realização do cálculo.

```
void operacoesMatrizes(char tipoMat, int linhaA, int colunaA, int lin
{
    int i, j, k;
    float matrizR[linhaA][colunaB], aux = 0;

    switch (tipoMat)
    {

    case '+':
        for (i = 0; i < linhaA; i++)
        {
            for (j = 0; j < colunaB; j++)
            {
                matrizR[i][j] = matrizA[i][j] + matrizB[i][j];
            }
        }

        break;

    case '-':

        for (i = 0; i < linhaA; i++)
```

O terceiro if é referente ao mini game aqui apresenta as variáveis que foram utilizadas nessa etapa

```
if (opcao == 3)
{
    // MINI GAME...

    while (opcao2 == 1)
    {
        // variaveis do mini game
        int tentativas, valorAleatorio, valorUsu
        char tipoGame;
        int salvar, historico, j, y, maiorPontua
        int quantJogadores, indice = 0;
```

```
printf("=====
printf(".....BEM VINDO.....
printf(".....AO.....
printf(".....MINI GAME.....
printf("=====
printf("\n\n\n");

printf(">>> Informe a quantidade de jogadores que irão participar: ");
scanf("%d", &quantJogadores);
system("clear");

cadastro vetCadastro[quantJogadores];
```

Aqui a apresenta uma parte da interface do mini game, e também como pode ser visto temos um tipo struct:

```
typedef struct
{
    char nomeJogador[99];
    int cpf;
    int idade;
} cadastro;
```

É responsável pelo cadastro dos usuários que vão participar do mini game, esse for é responsável pelo cadastro dos jogadores

```
for (int k = 0; k < quantJogadores; k++)
{
    printf("#####
    printf("Cadastro do %d° jogador\n", (k + 1));
    printf("Nome:\n");
    scanf("%[^\n]s", vetCadastro[k].nomeJogador);
    fflush(stdin);
    printf("Cpf:\n");
    scanf("%d", &vetCadastro[k].cpf);
    printf("Idade:\n");
    scanf("%d", &vetCadastro[k].idade);
    fflush(stdin);
    printf("#####
    getchar();
    system("clear");
}
```

Esse primeiro for da imagem abaixo é responsável por gerenciar as jogadas de cada jogador.

```
for (i = 0; i < quantJogadores; i++)
{
    for (j = 0; j < tentativas; j++)
    {
        valor1mini[j] = rand() % valorAleatorio;
        valor2mini[j] = rand() % valorAleatorio;
    }
}
```

O for logo abaixo é responsável por sortear os valores aleatórios.

Em seguida esse for é responsável por receber a resposta de cada jogador.

```
for (j = 0; j < tentativas; j++)
{
    // respostas do usuario
    printf(">> %s quanto é %.0f %c %.0f ?\n", vet
    printf("Sua resposta: ");
    scanf("%d", &valorUsu);

    switch (tipoGame)
    {
        case '+':
            resultado[j] = soma(valor1mini[j], valor2
            break;

        case '-':
            resultado[j] = subtracao(valor1mini[j], v
```

Essa parte do switch onde é chamada às funções foi aplicada da mesma forma que as outras opções anteriores.

Nessa segunda função que analisa se o jogador acertou ou não a resposta como mostra no terminal abaixo.

```
.....
>> vitor quanto é 0 x 0 ?
Sua resposta: 0

PARABÉNS VOCÊ ACERTOU!!! :)

.....

>> vitor quanto é 1 x 0 ?
Sua resposta: █
```

Na imagem abaixo é realizado a contagem dos pontos de cada jogador.

```
pontuacao[i] = (acertos * valorAleatorio) / (tentativas / quantJogadores);
```

Logo abaixo tem o if que verifica se o jogador que salvar os seus dados com seu placar.

```
if (salvar == 1)
{
    fprintf(arquivo, "Dados do jogador : %s\nCpf : %d\nIdade : %d\nPontuação total
```

Os próximos print apresenta a verificação que é feita para salvar quem foi o ganhador, ou verificar quando ocorrer empate.

```
for (int x = 0; x < quantJogadores; x++)
{
    if (pontuacao[x] > maiorPontuacao)
    {
        maiorPontuacao = pontuacao[x];
        empate = 0;
    }
    for (y = quantJogadores; y > x; y--)
        if (pontuacao[x] == pontuacao[y])
        {
            empate = 1;
        }
}

if (empate == 1)
{
    printf("Ocorreu empate.\n");
}
else
{
    for (int x = 0; x < quantJogadores; x++)
    {
```

CONCLUSÃO

Portanto, este projeto tem como objetivo ajudar seus usuários a realizar com mais facilidades operações, além de ser um programa de entretenimento com amigos e outros com a opção do mini game, onde você pode competir com outras pessoas enquanto aprende.