



CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
ESCOLA SUPERIOR POLITÉCNICA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO

ATIVIDADE PRÁTICA

OZEIAS FERREIRA MARQUES– RU: 2265904
Prof. Winston SenLun Fung

TRES LAGOAS– ms
2020

Exercício 1:

Vamos supor que várias pedras do jogo de xadrez estão no tabuleiro. Para facilitar a indicação das peças, vamos convencionar:

1 – peões 3 – torres 5 – reis 0 – ausência de peças
2 – cavalos 4 – bispos 6 – rainhas

O tabuleiro é o seguinte:

Construa um programa, em linguagem C, que:

a) Solicite ao usuário que informe qual peça deseja colocar em cada posição do tabuleiro.

b) Mostre o tabuleiro informado pelo usuário na tela.

c) Determine a qual a quantidade de cada tipo de peça no tabuleiro.

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

//ATIVIDADE PRATICA 2020 OZEIAS FERREIRA MARQUES, ANALIZE E
DESENVOLVIMENTO DE SISTEMAS, PROFESSOR: WINSTON SENLUN FUNG

```
enum pecas { espacovazio, peao, cavalo, torre, bispo, rei, rainha }; // VARIABEL  
CONTADORA  
int main()
```

```
{  
    setlocale(LC_ALL, "Portuguese");  
    float c; //VARIABEL DO TIPO FLUTUANTE
```

// CRIA TABULEIRO.

```
    int tabuleiro[8][8] = {  
        {1, 3, 0, 5, 4, 0, 2, 1},  
        {1, 0, 1, 0, 0, 1, 0, 0},  
        {0, 0, 0, 0, 1, 0, 6, 0},  
        {1, 0, 0, 1, 1, 0, 0, 1},  
        {0, 1, 0, 4, 0, 0, 1, 0},  
        {0, 0, 3, 1, 0, 0, 1, 1},  
        {1, 0, 6, 6, 0, 0, 1, 0},  
        {1, 0, 5, 0, 1, 1, 0, 6}  
    };  
    // MOSTRA TABULEIRO  
    printf("\nObserve o tabuleiro abaixo:\n\n");
```

```
    int linha, coluna;  
    for (linha = 0; linha < 8; linha++)  
    {  
        for (coluna = 0; coluna < 8; coluna++)  
        {  
            printf("%d ", tabuleiro[linha][coluna]);  
        }
```

```

        }
        printf("\n");
    }

int contador[7] = { 0 }; // VETOR CONDICIONAL BASEADO NO EXERCICIO
    int i, x;
    for (i = 0; i < 8; i++) {
        for (x = 0; x < 8; x++) {
            switch (tabuleiro[i][x])
            {
                case espacovazio:
                    contador[0]++;
                    break;
                case peao:
                    contador[1]++;
                    break;
                case cavalo:
                    contador[2]++;
                    break;
                case torre:
                    contador[3]++;
                    break;
                case bispo:
                    contador[4]++;
                    break;
                case rei:
                    contador[5]++;
                    break;
                case rainha:
                    contador[6]++;
                    break;
                default:
                    break;
            }
        }
    }
}

printf("\n");
printf("SIGNIFICADO DE CADA PEÇA EM NUMEROS\n");
printf("0 - Espaço vazio | 1 - Peão | 2 - Cavalo | 3 - Torre | 4 - Bispo | 5 - Rei | 6 - Rainha\n\n");

printf("\nQuantidade de casas vazias e peças:\n\n");
printf("Casas vazias\t= %d\n", contador[0]);
printf("Peões\t\t= %d\n", contador[1]);
printf("Cavalos\t\t= %d\n", contador[2]);
printf("Torres\t\t= %d\n", contador[3]);
printf("Bispos\t\t= %d\n", contador[4]);
printf("Reis\t\t= %d\n", contador[5]);
printf("Rainhas\t\t= %d\n", contador[6]);
printf("\n");

```

```

        system("pause");
//CRIA TABULEIRO DO ZERO!
    contador[0] = 0;
    contador[1] = 0;
    contador[2] = 0;
    contador[3] = 0;
    contador[4] = 0;
    contador[5] = 0;
    contador[6] = 0;
//REPETE TABULEIRO JÁ ALTERADO
    int tabuleirousuario[8][8] = {
        {1, 3, 0, 5, 4, 0, 2, 1},
        {1, 0, 1, 0, 0, 1, 0, 0},
        {0, 0, 0, 0, 1, 0, 6, 0},
        {1, 0, 0, 1, 1, 0, 0, 1},
        {0, 1, 0, 4, 0, 0, 1, 0},
        {0, 0, 3, 1, 0, 0, 1, 1},
        {1, 0, 6, 6, 0, 0, 1, 0},
        {1, 0, 5, 0, 1, 1, 0, 6}
    };
int linhausuario, colunausuario; // VARIAVEIS DE PREENCHIMENTO
int escolhausuario;
int SimOuNao;
system("cls");
printf("Agora vamos começar a montar o seu tabuleiro!\n\n");
// PRINTA O TABULEIRO

for (linha = 0; linha < 8; linha++)
{
    for (coluna = 0; coluna < 8; coluna++)
    {
        printf("%d ", tabuleirousuario[linha][coluna]);
    }
    printf("\n");
}

// ADICIONA AS PEÇAS ESCOLHIDAS
printf("\nDigite a posição da linha (1 a 8) que deseja alterar:\n");
scanf_s("%d", &linhausuario);
while ((linhausuario < 1) || (linhausuario > 8))
{
    printf("O valor inserido é inválido! Digite um número de 1 a 8 para a posição na
linha!\n");
    scanf_s("%d", &linhausuario);
}

printf("Agora, digite a posição da coluna (1 a 8) que deseja alterar:\n");
scanf_s("%d", &colunausuario);
while ((colunausuario < 1) || (colunausuario > 8))
{
}

```

```

printf("O valor inserido é inválido! Digite um número de 1 a 8 para a posição na
coluna!\n");
scanf_s("%d", &colunausuario);
}

printf("Digite a peça que deseja inserir na posição %d,%d:\n", linhausuario,
colunausuario);
printf("0 = espaço em branco\n");
printf("1 = Peão\n");
printf("2 = Cavalo\n");
printf("3 = Torre\n");
printf("4 = Bispo\n");
printf("5 = Rei\n");
printf("6 = Rainha\n");
scanf_s("%d", &escolhausuario);
while ((escolhausuario < 0) || (escolhausuario > 6))
{
printf("A peça desejada não existe! Insira um valor válido conforme o menu!\n");
scanf_s("%d", &escolhausuario);
}

```

```

tabuleirousuario[linhausuario - 1][colunausuario - 1] = escolhausuario;
printf("QUER ALTERAR OUTRA POSIÇÃO?\n");
printf("(1) para SIM e (2) para NÃO\n");
scanf_s("%d", &SimOuNao);
while (SimOuNao == 1) // LAÇO DE ESPERA
{
printf("\nDigite a posição da linha (1 a 8) que deseja alterar:\n");
scanf_s("%d", &linhausuario);
while ((linhausuario < 1) || (linhausuario > 8)) // //AGUARDA PEÇA VALIDA
{
printf("O valor inserido é inválido! Digite um número de 1 a 8 para a posição
na linha!\n");
scanf_s("%d", &linhausuario);
}
printf("Agora, digite a posição da coluna (1 a 8) que deseja alterar:\n");
scanf_s("%d", &colunausuario);
while ((colunausuario < 1) || (colunausuario > 8)) // //AGUARDA PEÇA VALIDA
{
printf("O valor inserido é inválido! Digite um número de 1 a 8 para a posição
na coluna!\n");
scanf_s("%d", &colunausuario);
}
printf("Digite a peça que deseja inserir na posição %d,%d:\n", linhausuario,
colunausuario);
printf("0 = espaço em branco\n");
printf("1 = Peão\n");
printf("2 = Cavalo\n");

```

```

printf("3 = Torre\n");
printf("4 = Bispo\n");
printf("5 = Rei\n");
printf("6 = Rainha\n");
scanf_s("%d", &escolhausuario);
while ((escolhausuario < 0) || (escolhausuario > 6)) //AGUARDA PEÇA VALIDA
{
    printf("A peça desejada não existe! Insira um valor válido conforme o
menu!\n");
    scanf_s("%d", &escolhausuario);
}
tabuleirousuario[linhausuario - 1][colunausuario - 1] = escolhausuario;
printf("Deseja alterar outra posição?\n");
printf("(1) para SIM e (2) para NÃO\n");
scanf_s("%d", &SimOuNao);

}

//MOSTRA TABULEIRO
system("cls");
printf("Tabuleiro de Xadrez!\n\n");
printf("Este é o seu tabuleiro!\n\n");
int linha1, coluna1;
int contadorusuario[7];
for (linha1 = 0; linha1 < 8; linha1++)
{
    for (coluna1 = 0; coluna1 < 8; coluna1++)
    {
        printf("%d ", tabuleirousuario[linha1][coluna1]);
    }
    printf("\n");
}
int j, l;
for (j = 0; j < 8; j++) {
    for (l = 0; l < 8; l++)
    {
        switch (tabuleirousuario[j][l])
        {
            case espacovazio:
                contador[0]++;
                break;
            case peao:
                contador[1]++;
                break;
            case cavalo:
                contador[2]++;
                break;
            case torre:
                contador[3]++;
                break;
            case bispo:
                contador[4]++;
                break;
            case rei:
                contador[5]++;
                break;
            case rainha:
                contador[6]++;
                break;
        }
    }
}

```

```

                break;
            case bispo:
                contador[4]++;
                break;
            case rei:
                contador[5]++;
                break;
            case rainha:
                contador[6]++;
                break;
            default:
                break;
        }
    }
}

// MOSTRA CONTADOR
printf("\n");
printf("LEGENDA\n");
printf("0 - Espaço vazio | 1 - Peão | 2 - Cavalo | 3 - Torre | 4 - Bispo | 5 - Rei | 6 -
Rainha\n\n");
printf("\nQuantidade de casas vazias e peças:\n");
printf("Casas vazias\t= %d\n", contador[0]);
printf("Peoes\t\t= %d\n", contador[1]);
printf("Cavalos\t\t= %d\n", contador[2]);
printf("Torres\t\t= %d\n", contador[3]);
printf("Bispos\t\t= %d\n", contador[4]);
printf("Reis\t\t= %d\n", contador[5]);
printf("Rainhas\t\t= %d\n", contador[6]);
printf("\n");

int escolhafimprograma;
printf("Digite (1) para encerrar o programa ou (2) para exibir um tabuleiro inicial do
jogo!\n"); //OPÇÃO DE ESCOLHA PARA UDUARIO
scanf_s("%d", &escolhafimprograma);
while ((escolhafimprograma < 1) || (escolhafimprograma > 2)) //AGUARDA PEÇA VALIDA
{
    printf("\nOpção inválida! Digite (1) para encerrar ou (2) para exibir um tabuleiro
inicial do jogo!\n");
    scanf_s("%d", &escolhafimprograma);
}
if (escolhafimprograma == 1) // CONDIÇÃO PARA CONTINUAR OU PARAR
{
    system("cls");
    printf("XEQUE-MATE!");
    return 0;
}
if (escolhafimprograma == 2) //EXIBE TABULEIRO
{
    contador[0] = 0;
}

```

```

    contador[1] = 0;
    contador[2] = 0;
    contador[3] = 0;
    contador[4] = 0;
    contador[5] = 0;
    contador[6] = 0;
    system("cls");
printf("POSIÇÃO INICIAL DAS PEÇAS DE UM TABULEIRO DE XADREZ\n\n");
int tabuleirodefault[8][8] = { //INICIA UM TABULEIRO NOVO
{3, 2, 4, 6, 5, 4, 2, 3},
{1, 1, 1, 1, 1, 1, 1, 1},
{0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0},
{1, 1, 1, 1, 1, 1, 1, 1},
{3, 2, 4, 6, 5, 4, 2, 3}
};
int linhadefault, colunadefault;
for (linhadefault = 0; linhadefault < 8; linhadefault++)
{
    for (colunadefault = 0; colunadefault < 8; colunadefault++)
    {
        printf("%d ", tabuleirodefault[linhadefault][colunadefault]);
    }
    printf("\n");
}
int m, n;
for (m = 0; m < 8; m++) {
    for (n = 0; n < 8; n++) {
        switch (tabuleirodefault[m][n])
        {
            case espacovazio:
                contador[0]++;
                break;
            case peao:
                contador[1]++;
                break;
            case cavalo:
                contador[2]++;
                break;
            case torre:
                contador[3]++;
                break;
            case bispo:
                contador[4]++;
                break;
        }
    }
}

```

```

        case rei:
            contador[5]++;
            break;
        case rainha:
            contador[6]++;
            break;
        default:
            break;
        }
    }
}

// EXIBE O CONTADOR
printf("\n");
printf("LEGENDA\n");
printf("0 - Espaço vazio | 1 - Peão | 2 - Cavalo | 3 - Torre | 4 - Bispo | 5 - Rei | 6 -
Rainha\n\n");
printf("\nQuantidade de casas vazias e peças:\n");
printf("Casas vazias\t= %d\n", contador[0]);
printf("Peões\t\t= %d\n", contador[1]);
printf("Cavalos\t\t= %d\n", contador[2]);
printf("Torres\t\t= %d\n", contador[3]);
printf("Bispos\t\t= %d\n", contador[4]);
printf("Reis\t\t= %d\n", contador[5]);
printf("Rainhas\t\t= %d\n", contador[6]);
printf("\n");
system("pause");
}
system("cls");
printf("XQUE-MATE!\n");
return 0;
}

```

Imagen do código funcionando no seu computador: