

```
//PROJETO INTERDISCIPLINAR 1 - PROFESSORA: ANA MONTEIRO
// JOGO CAMPO MINADO - DESENVOLVIDO EM LINGUAGEM C
// ALUNOS: GIVANILSON MOURA - RA: 22345
//          NICHOLAS GUARATINI - RA: 21912
//          RAQUEL FIGUEIREDO - RA: 21910
// CIENCIAS DA COMPUTAÇÃO - 4º SEMESTRE
```

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

```
#define li 500
#define col 500
```

```
//recebe uma matriz, dimensões da matriz e a quantidade de bombas, as bombas então são
geradas em posições aleatórias. 1 = bomba
```

```
void gera_campoMinado(int a[li][col], int linha, int coluna, int bomba ) {
```

```
    int x = 0;
    int cont = 0;
    while( cont < bomba ) {
        int i = rand() % linha ;
        int j = rand() % coluna;
        if (a[i][j] == 0) {
            a[i][j] = 1;
            cont++;
        }
    }
}
```

```
// Imprime o campo minado, X = bomba
```

```
void imprimir_campoMinado(int a[li][col], int linha, int coluna) {
```

```
    int soma = 0;

    for(int i = 1; i < linha; i++) {
        for(int j = 1; j < coluna; j++) {
            if (a[i][j] == 1) {
                printf("[ X ]");
                soma = soma +1;
            }
            else printf("[ O ]");
        }
        printf("\n");
    }
    printf("\n\n total de bombas = %d\n", soma);
}
```

```
// Imprime a quantidade de bombas ao redor de cada posição.
```

```
void imprimir_campoVizinhas(int a[li][col], int linha, int coluna) {
```

```
    for (int i = 1; i < linha; i++) {
        for(int j = 1; j < coluna; j++) {
            printf ("[ %d ]", a[i][j]);
        }
        printf("\n");
    }
}
```

```
// Verifica quantas bombas existem ao redor de cada posição. Utilizando uma função recursiva
```

```
void Verifica_vizinhas(int a[li][col], int b[li][col], int linha, int coluna) {
```

```
    int soma = 0;

    for( int i = 1; i < linha; i++)
        for (int j = 1; j < coluna; j++) {
```

```
        if ( j > 0 && a[i][j-1] == 1) soma++;

        if ( j < coluna && a[i][j+1] == 1) soma++;

        if ( i < linha && a[i+1][j] == 1) soma++;

        if ( i > 0 && a[i-1][j] == 1) soma++;

        if ( i < linha && j>0 && a[i+1][j-1] == 1) soma++;

        if ( i < linha && j < coluna && a[i+1][j+1] == 1 ) soma++;

        if ( i > 0 && j > 0 && a[i-1][j-1] == 1) soma++;

        if ( i > 0 && j < linha && a[i-1][j+1] == 1) soma++;

        b[i][j] = soma;

        soma = 0;
    }
}

// Imprime a matriz do jogo como está atualmente

void imprimir_jogo(int CampoMinado[li][col],int Vizinhas[li][col], int linha, int coluna,
int posicao_A, int posicao_B ) {

    int i,j;

    for(i=1; i<linha;i++){
        for(j=1; j< coluna; j++) {

            if(CampoMinado[i][j] <= (linha*coluna) )
                printf("[ ? ]");

            else if (CampoMinado[i][j] == ((linha*coluna) +1)) {
                printf("[ X ]");
            }
            else if (CampoMinado[i][j] == ((linha*coluna)+2)) {
                printf("[ %d ]",Vizinhas[i][j]);
            }
        }
        printf("\n");
    }
}

// Inicia o jogo, recebendo as posições e imprimindo o estado do jogo em cada passo

void jogo(int CampoMinado[li][col], int Vizinhas[li][col], int linha, int coluna) {

    int posicao_A;
    int posicao_B;
    int fimdeJogo = 0;
    printf("inicio de jogo!");
    printf("\n\n");
    imprimir_jogo(CampoMinado, Vizinhas, linha, coluna, posicao_A, posicao_B);
    printf("\n");

    while(fimdeJogo != 1) {
        printf("Informe uma posicao:\n ");

        do {
            printf("linha(menor que %d): ", linha);
            scanf("%d", &posicao_A);
        }

        while(posicao_A < 0 || posicao_A >= linha);
    }
}
```

```
do {
    printf("coluna(menor que %d): ", coluna);
    scanf("%d", &posicao_B);
}

while(posicao_B < 0 || posicao_B >= coluna);

// Verifica se dentro dessas posições tem bomba se tiver ele dá um estouro e finaliza o
jogo

if(CampoMinado[posicao_A][posicao_B] == 1) {
    CampoMinado[posicao_A][posicao_B] = (linha * coluna) + 1;

    printf(" BOOOOM!!! Voce Perdeu!!! :( \n");

    imprimir_jogo(CampoMinado, Vizinhas, linha, coluna, posicao_A, posicao_B);
    fimdeJogo = 1;
}

// Caso contrário se não tiver bomba na posição que o usuário informar, o jogo pede para
informar nova posição.

else if (CampoMinado[posicao_A][posicao_B] == 0) {
    CampoMinado[posicao_A][posicao_B] = (linha * coluna) + 2;
    imprimir_jogo(CampoMinado, Vizinhas, linha, coluna, posicao_A, posicao_B);
}
}
}

int main() {

    int campoMinado[li][col], campoVizinhas[li][col], linha, coluna, bomba;

    printf("informe a quantidade de linhas: ");
    scanf("%d", &linha);

    printf("\n");

    printf("informe a quantidade de colunas: ");
    scanf("%d", &coluna);

    printf("\n");

    printf("informe a quantidade maxima de bombas: ");
    scanf("%d", &bomba);

    printf("\n");

    for(int i = 1; i < linha; i++) {
        for( int j = 1; j < coluna; j++) {
            campoMinado[i][j] = 0;
        }
    }

    for(int i = 1; i < linha; i++) {
        for( int j = 1; j < coluna; j++) {
            campoVizinhas[i][j] = 0;
        }
    }

    printf("\n\n Campo Minado completo:\n");

    printf("\n");

    srand( (unsigned)time(NULL) );

    gera_campoMinado(campoMinado, linha, coluna, bomba);
}
```

```
    imprimir_campoMinado(campoMinado, linha, coluna);

    Verifica_vizinhas(campoMinado, campoVizinhas, linha, coluna);

    printf("\n\n\n");

    imprimir_campoVizinhas(campoVizinhas, linha, coluna);

    printf("\n\n\n\n\n\n\n");

    jogo(campoMinado, campoVizinhas, linha, coluna);

    system("pause");
    return 0;
}
```