



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Estrutura de Dados I – BCC 202
Aluno: Felipe Fontenele de Ávila Magalhães
Matrícula: 15.1.4331



Documentação Questão 2

Implementação

O código em questão, foi separado em arquivos **.h**, **.cpp** e o **main.cpp**. Inicialmente é feita uma matriz na qual o usuário define o tamanho da mesma, sendo necessário usar alocação dinâmica, já que não sabemos qual vai ser o tamanho que o usuário vai definir. Em um segundo momento, pergunta-se a forma que o usuário quer preencher a matriz, se vai ser aleatoriamente ou se ele quer preencher campo a campo, garantindo que, mesmo quando o usuário digitar uma opção diferente do que foi predefinido, pergunta-se novamente até ser uma opção satisfatória, sendo nesse caso, **A** para aleatoriamente e **M** para preencher manualmente. Após, imprime a matriz em questão e pergunta a cor que o usuário quer preencher e nas coordenadas que ele quer colocar. Também tenho a preocupação de garantir que a cor esteja entre as opções predefinidas e que as coordenadas não ultrapassem as medidas da matriz. Logo após, é chamada a função recursiva **Boundaryfill** na qual são parâmetros: a matriz, a cor que o usuário quer preencher, as coordenadas (x, y) que o usuário queira colocar na matriz, as dimensões da matriz (largura, altura) e a cor antiga naquela posição da matriz que o usuário quer alterar. Dentro dessa função, chama-se 4 vezes a si mesma para verificar a vizinhança se existe uma cor igual ao que havia anteriormente, fazendo que se for verdade, essa posição receber a cor nova. Para essa função não entrar em um loop infinito, foram criadas condições de parada como, por exemplo, as coordenadas (x, y) tem que serem maiores ou iguais a zero e menores que a dimensão da matriz e que a cor nas coordenadas desejada seja igual à cor anterior. Por fim, imprime a matriz com as devidas alterações e efetua a liberação da memória alocada dinamicamente.

Impressões gerais

Em minha opinião, essa questão foi relativamente fácil, contudo houve alguns obstáculos na construção do código, como por exemplo, se a matriz já iria preenchida aleatoriamente ou se o usuário iria preenchê-la manualmente. Obtive confusão também na hora de definir a função recursiva, analisando os casos em que seria necessária a interrupção da mesma. Percebi semelhanças entre esse código e um jogo chamado Campo Minado, na qual são liberadas casas até onde não há minas, isto é, possuindo a mesma cor, fazendo uma analogia ao código.



Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Estrutura de Dados I – BCC 202
Aluno: Felipe Fontenele de Ávila Magalhães
Matrícula: 15.1.4331



Conclusões

Esse código como foi dito anteriormente, foi de um nível fácil para intermediário onde fiz com muita atenção para não desobedecer a nenhuma norma imposta e concluo que consegui fazer de uma forma que garanto a qualidade do código, tentando reduzir o uso de variáveis ou comparações desnecessárias. Foram feitas funções para preencher a matriz, tanto aleatoriamente quanto manualmente, a função Boundaryfill, outra para imprimir a matriz e uma para liberar a memória.