

Questão 01.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <limits.h>
5
6  /* Definições de tamanho de matriz e faixa de números aleatórios */
7  #define N 4
8  #define M 5
9  #define MIN 1
10 #define MAX 20
11
12 /* Função para gerar uma matriz de tamanho NxM preenchida com números aleatórios de 1 a 20 */
13 void generateMatrix(int matrix[N][M]){
14     /* Semeia a função rand com o tempo atual para garantir que os números gerados sejam diferentes a
15     cada execução do programa */
16     srand(time(NULL));
17     for(int i = 0; i < N; i++){
18         for(int j = 0; j < M; j++){
19             /* Preenche a posição atual da matriz com um número aleatório de 1 a 20 */
20             matrix[i][j] = (rand() % (MAX)) + MIN;
21             printf("%d ", matrix[i][j]);
22         }
23         printf("\n");
24     }
25 }
26
27 /* Função para verificar se a matriz satisfaz a condição especificada */
28 int verifyCondition(int matrix[N][M]){
29     int minSum = INT_MAX, maxMult = INT_MIN;
30
31     for(int i = 0; i < N; i++){
32         int sum = 0;
33         for(int j = 0; j < M; j++){
34             sum += abs(matrix[i][j]);
35         }
36         /* Se a soma da linha atual é menor que o mínimo encontrado até agora, atualiza o mínimo */
37         if(sum < minSum){
38             minSum = sum;
39         }
40     }
41
42     printf("\nSomatorio Minimo dentre todas as linhas:%d\n", minSum);
43
44     for(int i = 0; i < M; i++){
45         int mult = 1;
46         for(int j = 0; j < N; j++){
47             mult *= matrix[j][i];
48         }
49         /* Se a multiplicação da coluna atual é maior que o máximo encontrado até agora, atualiza o
50 máximo */
51         if(mult > maxMult){
52             maxMult = mult;
53         }
54     }
55
56     printf("\nProdutorio Maximo dentre todas as colunas:%d\n", maxMult);
```

```
57      /* Retorna 1 (verdadeiro) se o maior produto das colunas é maior ou igual à menor soma das  
linhas.  
58          Caso contrário, retorna 0 (falso). */  
59      return maxMult >= minSum;  
60 }  
  
61  
62 int main(){  
63  
64     printf("\n\nXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n\n");  
65  
66     int matrix[N][M];  
67     generateMatrix(matrix);  
68  
69     if(verifyCondition(matrix)){  
70         printf("\nCondicao satisfeita.\n");  
71     }  
72     else{  
73         printf("\nCondicao nao satisfeita.\n");  
74     }  
75  
76  
77     printf("\n\nXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n\n");  
78     return 0;  
79 }  
80
```