

```
/*
Um professor mantém as notas dos alunos das turmas em que leciona em tabelas, onde as linhas
correspondem aos alunos e as colunas correspondem às avaliações. O número de alunos e número de
avaliações não são conhecidos previamente e podem variar a cada turma.
```

Solicite o tamanho da turma e número de avaliações e implemente as funções a seguir usando alocação dinâmica:

- maior_nota: retorna a maior nota obtida pela turma, além do número do aluno e o número da avaliação correspondente à maior nota da turma;
- media_turma: retorna a média das notas obtidas pela turma, considerando todas as avaliações realizadas.

Obs: a tabela de notas dos alunos da turma deve ser gerada aleatoriamente com números de 0 a 10.

```
*/
```

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

```
/*
Função que retorna o maior elemento de uma matriz
*/
```

```
float maior(float ** M,int lin, int col)
{
    int i,j;
    float m;
    m = 0;
    for (i = 0; i < lin; i++)
    {
        for (j = 0; j < col; j++)
        {
            if (M[i][j] > m)
            {
                m = M[i][j];
            }
        }
    }
    return m;
}
```

```
/*
Função que retorna a média dos valores armazenados em uma matriz
*/
```

```
float media(float ** M, int lin, int col)
{
    int i,j;
    float soma;
    soma = 0;
    for (i = 0; i < lin; i++)
    {
        for (j = 0; j < col; j++)
        {
            soma += M[i][j];
        }
    }
    return (soma/(lin*col));
}
```

```
/*
Função que aloca memória para uma matriz dinamicamente e inicializa com valores aleatórios de 0 a
10
*/
```

```
float** criaMatriz(int lin,int col)
{
    int i,j;
    float** M;
    //Alocação dinamica de memória para a tabela de notas,
    //considerando o número de alunos e avaliações informadas.
```

```

M = (float**) calloc(lin, sizeof (float*));

//Verifica se a alocação foi realizada com sucesso.
if (M == NULL)
{
    printf("Erro durante a alocação de memória");
    exit(1);
}
for (i = 0; i < lin; i++)
{
    M[i] = (float *) calloc(col, sizeof (float));

    //Verifica se a alocação foi realizada com sucesso.
    if (M[i] == NULL)
    {
        printf("Erro durante a alocação de memória");
        exit(1);
    }
    for (j = 0; j < col; j++)
    {
        M [i][j] = (rand() % 101) / 10.0;
    }
}

return M;
}

/*
Função que imprime valores de uma matriz
*/
void imprime(float** M, int lin, int col)
{

    int i, j;
    for (i = 0; i < lin; i++)
    {
        printf("%3d", (i+1));
        for (j = 0; j < col; j++)
            printf("%10.1f", M[i][j]);
        printf("\n");
    }
}

/*
Função que libera memória alocada dinamicamente para uma matriz
*/

void liberaMemoria(float ** M, int lin){
    int i;
    for (i = 0; i < lin; i++)
    {
        free(M[i]);
    }
    free(M);
}

int main(void)
{
    int nalunos, navaliacoes;
    int numal, numav;
    int i,j;
    float med, m;
    float** tabela;

    printf("Informe o número de alunos da turma: ");
    scanf("%d", &nalunos);
    printf("Informe o número de avaliações da turma: ");
    scanf("%d", &navaliacoes);

```

```
tabela = criaMatriz(nalunos, navaliacoes);

printf("\nTabela de notas: \n");
imprime(tabela,nalunos,navaliacoes);

m = maior(tabela,nalunos,navaliacoes);
med = media(tabela,nalunos,navaliacoes);
printf("\nMaior nota da turma: %10.1f", m);
printf("\nMédia da turma: %10.1f", med);

//Liberação da memória
liberaMemoria(tabela,nalunos);

return 0;
}
```