

Matrizes Especiais

Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, copie e cole o seu código-fonte com a resposta aqui mesmo neste documento, dentro dos espaços indicados para isso e preservando a indentação do código. Depois que terminar sua avaliação, não se esqueça de entregar sua atividade! Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

Uma matriz em banda tipo tridiagonal pode surgir em vários problemas numéricos. Considere a matriz tridiagonal a seguir que contém elementos na diagonal secundária, diagonal acima e abaixo e o restante dos elementos é zero. Considere que essa matriz pode ter no mínimo ordem 3.

$$T_{[5 \times 5]} = \begin{bmatrix} 0 & 0 & 0 & 3 & 4 \\ 0 & 0 & 7 & 8 & 9 \\ 0 & 11 & 12 & 13 & 0 \\ 15 & 16 & 17 & 0 & 0 \\ 20 & 21 & 0 & 0 & 0 \end{bmatrix}$$

Para essa atividade, utilize um TAD `MatrizEspecial` para representar a matriz por meio da representação linear com um único vetor (`vet`) tal que a quantidade de elementos armazenados seja mínima. E desenvolva:

- O construtor e o destrutor. Atente-se para dimensionar o `vet` de acordo com o tamanho mínimo necessário. O construtor deve preencher a matriz e garantir que a matriz tenha ordem no mínimo 3.

```
MatrizEspecial::MatrizEspecial(int ordem)
{
    if(ordem > 2)
    {
    }
    else
    {
        cout <<"Ordem invalida, tente novamente"<<endl;
    }
}
MatrizEspecial::~~MatrizEspecial()
{
    delete [] vetor;
```

```
}
```

- a) A operação `int getInd(int i, int j)` para verificar se os índices `i` e `j` da matriz são válidos e retornar o índice de `vet` (-2 se não precisa ser armazenado ou -1 caso não sejam válidos).

```
int MatrizEspecial::getInd(int i, int j)
{
    if(i>=0 && i<tam && j>=0 && j<tam)
    {
        if(i+j==tam-1 || i+j==tam-2 || i+j==tam) return indice;
        else return -2;
    }
    else return -1;
}
```

- b) A operação `int get(int i, int j)` que retorna o valor armazenado na posição `i, j` da matriz, zero caso o índice seja -2 e imprime uma mensagem de erro e sai do programa caso o índice seja inválido.

```
int MatrizEspecial::get(int i, int j)
{
    int q=getInd(i, j);

    if(q==-1)
    {
        cout << "indice invalido"<<endl;
        exit(1);
    }
    else
    {
        if(q==-2)
        {
            return 0.0;
        }else
        {
            return vetor[q];
        }
    }
}
```

- c) A operação `void set (int i, int j, int val)` que atribui o valor `val` à posição `i, j` da matriz e imprime mensagem de erro para índice incorreto.

```
void MatrizEspecial::set (int i, int j, int valor)
{
    int q=getInd(i, j);

    if(q<0)
    {
        cout << "índice inválido";
    }else
    {
        vetor[q]=valor;
    }
}
```