

Relatório do Trabalho III

(Matrizes Esparsas)

Leonardo Cerce Guioto
10716640

Sumário

| | |
|----------------------------|---------|
| Sumário | pág. 01 |
| Introdução | pág. 02 |
| Descrição do projeto | pág. 02 |
| Tutorial | pág. 05 |
| Outras informações | pág. 06 |

Introdução

O objetivo deste trabalho foi o de criar um programa, na Linguagem C, que permita a criação e manipulação de matrizes esparsas. Ele deve ser capaz de criar as matrizes, adicionar, consultar e fazer a soma dos valores presentes na mesma, e apagá-la.

Equipe: Leonardo Cerce Guioto - 10716640

Descrição do projeto

O projeto contém 3 arquivos: `Matrizes.c`, `funcoes.c` e `funcoes.h`. No primeiro arquivo está a `main{}`, enquanto no segundo e terceiro estão as funções em si, e as declarações das mesmas, respectivamente. O programa foi desenvolvido numa máquina cujo sistema operacional é o Windows 10 Home de 64 bits, através do programa Code::Blocks. O compilador usado foi o GNU GCC, com as seguintes flags: `-Wall` exibe todos os 'warnings' e `-g` foi usado para o 'debugging' do programa. A seguir serão mostradas as funções usadas no programa.

► Função #1

```
int linhas(void)
{
    int num_linhas;
    printf("Adicione o numero de linhas: ");
    for(;;)
    {
        fflush(stdin);
        fscanf(stdin, "%d", &num_linhas);
        if(num_linhas>0)
        {
            break;
        }
        else
        {
            printf("Numero de linhas invalido.\n");
        }
    }
    return num_linhas;
}
```

Esta função obtém do usuário o número de linhas que a matriz terá. A função que obtém o número de linhas é idêntica a essa e não será mostrada aqui.

► Função #2

```

int testav(char k)
{
    int i;

    i=k-'0';

    if(i<1||i>8)
    {
        for(;;)
        {
            printf("Opcao invalida.Tente novamente.\n");
            scanf("%c",&k);
            i=k-'0';
            if(i>=1&&i<=8)
            {
                i=0;
            }
        }
    }
    return i;
}

```

Esta função verifica se a opção seleccionada no menu é válida e retorna o número correspondente, que será usado no switch().

► Função #3

```

void addvalor(matriz* nova, int numlin, int numcol, int nmax,int numat)
{
    while(i)
    {
        printf("Digite o numero da linha: ");
        for(;;)
        {
            scanf("%d",&nlin);    nlin=nlin-1;
            if(nlin<numlin&& nlin>=0)
            {
                break;
            }
            printf("Valor invalido. Tente novamente. ");
        }

        .
        .
        .

        printf("Digite o valor a ser inserido: ");
        fflush(stdin);
        scanf("%d",&valor);
        nova[numat].coluna=ncol;nova[numat].linha=nlin;nova[numat].valor=valor;
        printf("Deseja adicionar mais algum valor? (S para sim e N para nao): ");
        scanf("%c",&teste);
        if(teste=='N' || teste=='n')
        {
            i=0;}}}}

```

Esta função controla a inserção de valores na matriz criada.

► Função #4

```
void consultar(matriz** achar, int numlin, int numcol, int maxi)
{
    while(i)
    {
        system("cls");
        printf("Digite o numero da linha: ");
        for(;;)
        {
            fflush(stdin);
            scanf("%d",&nlin); nlin=nlin-1;
            if(nlin<numlin&&nlin>=0)
            {
                break;
            }
            printf("Valor invalido. Tente novamente. ");
        }
        printf("Digite o numero da coluna: ");
        for(;;)
        {
            fflush(stdin);
            scanf("%d",&ncol); ncol=ncol-1;
            if(ncol<numcol&&ncol>=0)
            {
                break;
            }
            printf("Valor invalido. Tente novamente. ");
        }
        system("cls");

        .
        .
        .
        printf("Deseja consultar mais alguma posicao? (S para sim e N para
nao): ");
        fflush(stdin);
        scanf("%c",&teste);
        if(teste=='N' || teste=='n')
        {
            i=0;
        }
    }
}
```

A função acima controla as consultas dos valores que estão na matriz criada, exibindo na tela o valor de determinada posição que foi pesquisada pelo usuário.

► Função #5

```

void somalin(matriz* somando, int nlinmax, int ncol, int atual)
{
    int nlin;
    int soma=0;

    printf("Digite o numero da linha que deseja obter a soma dos valores: ");
    for(;;)
    {
        fflush(stdin);
        scanf("%d",&nlin); nlin=nlin-1;
        if(nlin>=0&& nlin<nlinmax)
        {
            break;
        }
        printf("Linha invalida. Tente novamente. ");
    }
    for(int i=0;i<atual;i++)
    {
        if(somando[i].linha==nlin){soma+=somando[i].valor;}
    }
    printf("Somatorio dos valores da linha %d: %d",nlin,soma);
}

```

A função acima mostra na tela o somatório dos valores que estão em determinada linha, que o usuário especifica. A função que mostra o somatório dos valores de uma coluna não será mostrada, pois é idêntica a esta.

Tutorial

Os arquivos .c devem ser compilados juntos usando o comando “gcc Matrizes.c funcoes.c -o Matrizes.out” e executados por “./Matrizes.out”. Caso esteja utilizando o Code::Blocks, apenas abra o arquivo “Matrizes.cbp” e selecione a opção “Build” para compilar o programa, e em seguida “Run” para executá-lo, ou ainda, abra o arquivo “Matrizes_Esparsas.exe”.

A seguir serão mostrados screenshots do programa com exemplos de criação de matrizes, inserção e consulta de valores e somatórios e exclusão da matriz criada.

► Exemplos



Figura 1 - Tela de ‘carregamento’

```
<Matriz Esparsa>
1 - Criar matriz
2 - Adicionar valor em uma posicao
3 - Excluir matriz
4 - Consultar valores
5 - Consultar soma de uma linha
6 - Consultar soma de uma coluna
7 - Sair

Escolha uma opcao:
```

Figura 2 - Menu de opções

```
Adicione o numero de linhas: 1000
Adicione o numero de colunas: 1000
Matriz esparsa criada!

Pressione qualquer tecla para continuar. . .
```

Figura 3 - Criação de matriz

```
Digite o numero da linha: 995
Digite o numero da coluna: 123
Digite o valor a ser inserido: 10
Deseja adicionar mais algum valor? (S para sim e N para nao):
```

Figura 4 - Adicionando um valor à matriz

```
Digite o numero da linha: 995
Digite o numero da coluna: 123

Valor na linha 995 e coluna 123 = 10

Pressione qualquer tecla para continuar. . .

Deseja consultar mais alguma posicao? (S para sim e N para nao):
```

Figuras 5 e 6 - Consulta de valor de uma posição da matriz

```
Digite o numero da linha que deseja obter a soma dos valores: 995
Digite o numero da coluna que deseja obter a soma dos valores: 123
Somatorio dos valores da coluna 123: 10

Pressione qualquer tecla para continuar. . .
```

Figuras 7 e 8 - Consulta do somatório dos valores de determinada linha ou coluna

```
Matriz esparsa apagada.

Pressione qualquer tecla para continuar. . .
```

Figura 9 - Exclusão da matriz criada

Outras informações

Bugs e limitações

[Limitação] - O programa tem baixa portabilidade no Linux devido às funções utilizadas para fazê-lo.

[Limitação] - A quantidade máxima de valores que podem ser salvos na matriz é determinada pelo hardware do usuário.