

```

/* Grupo:
 * Igor Gomes Oliveira Ramos RA: 2920481911008
 * Leandro de Oliveira Lozano RA: 2920481911043
 * Michel de Lima Soares RA: 2920481911045
 */
public class Grafo {
    private ListaLigada[] tabela;

    /**
     * Construtor da classe, define a quantidade de vértices da tabela
     * @param qtdeVertices
     */
    public Grafo(int qtdeVertices) {
        tabela = new ListaLigada[qtdeVertices];
    }

    /**
     * Função de espalhamento da tabela
     * @param chave
     * @return int
     */
    public int fEspalha(int chave) {
        return chave - 1;
    }

    /**
     * Adiciona v1 ao índice v2 e v2 ao índice v1 com a modificação da
     * função de espalhamento
     * @param v1
     * @param v2
     */
    public void adiciona(int v1, int v2) {
        if (v1 > 0 && v1 <= tabela.length) {
            int i = fEspalha(v1);

            if (tabela[i] == null)
                tabela[i] = new ListaLigada();

            tabela[i].adicionaFinal(v2);
        }

        if (v2 > 0 && v2 <= tabela.length) {
            int i = fEspalha(v2);

            if (tabela[i] == null)
                tabela[i] = new ListaLigada();

            tabela[i].adicionaFinal(v1);
        }
    }
}

```

@Override

```
public String toString() {  
    String msg = "|";  
  
    for (int i = 0; i < tabela.length; ++i) {  
        msg += String.format(" [%d] -> %s |", i + 1, tabela[i] != null?tabela[i]:"null");  
    }  
  
    return msg;  
}  
}
```

```
/**
 * Grupo:
 * Igor Gomes Oliveira Ramos RA: 2920481911008
 * Leandro de Oliveira Lozano RA: 2920481911043
 * Michel de Lima Soares RA: 2920481911045
 */
```

```
public class ListaLigada {
    private No inicio, fim;
```

```
/**
 * Adiciona um novo nó no início da lista
 * @param e
 */
```

```
public void adicionaInicio(int e) {
    No novo = new No(e);
```

```
    if (inicio == null) {
        inicio = novo;
        fim = novo;
```

```
    }
    else {
        novo.prox = inicio;
        inicio.ant = novo;
        inicio = novo;
```

```
    }
```

```
}
```

```
/**
 * Adiciona um novo nó no final da lista
 * @param e
 */
```

```
public void adicionaFinal(int e) {
    No novo = new No(e);
```

```
    if (inicio == null) {
        inicio = novo;
        fim = novo;
```

```
    } else {
        fim.prox = novo;
        novo.ant = fim;
        fim = novo;
```

```
    }
```

```
}
```

```
/**
 * Remove um nó do início da lista
 * @return int removido
 */
```

```
public int removeInicio() {
    if (inicio == null)
        return -1;
```

```

        int r = inicio.dado;
        inicio = inicio.prox;
        inicio.ant = null;
        return r;
    }

    /**
     * Remove um nó do final da lista
     * @return int removido
     */
    public int removeFinal() {
        if (inicio == null)
            return -1;

        int r = fim.dado;
        fim.ant = fim;
        fim.prox = null;
        return r;
    }

    /**
     * Retorna uma String com a tabela de espalhamento
     * @return String
     */
    @Override
    public String toString() {
        String msg = "";

        No aux = inicio;

        while (aux.prox != null) {
            msg += aux.dado + " -> ";
            aux = aux.prox;
        }

        return msg + aux.dado;
    }
}

```

```
/**
 * Grupo:
 * Igor Gomes Oliveira Ramos RA: 2920481911008
 * Leandro de Oliveira Lozano RA: 2920481911043
 * Michel de Lima Soares RA: 2920481911045
 */
public class No {
    public No prox;
    public No ant;
    public int dado;

    public No(int dado) {
        this.dado = dado;
    }
}
```