

### Lista de Exercícios 4 – Listas

A lista de exercícios deverá ser feita individualmente e enviada para o e-mail “[jorgehpopae@gmail.com](mailto:jorgehpopae@gmail.com)” com o assunto “LISTA\_4 - <PRIMEIRONOME>\_<NºUSP>”, por exemplo, “LISTA\_4 – Jorge\_9999999”. O e-mail deve conter no anexo apenas 1 arquivo compactado, “LISTA4\_<NOME>\_<NºUSP>.zip”, por exemplo, “LISTA4\_Jorge\_9999999.zip”. Neste arquivo, devem estar os exercícios resolvidos em arquivos separados, com nomes “exercicio1.c”, “exercicio2.c”, etc. A data limite para entrega é **20/10**.

1) Considere um TAD **lista ordenada duplamente encadeada** que armazene strings e a estrutura:

```
typedef struct bloco{
    struct bloco *esq, *dir;
    char str[100];
}no;
```

Implemente as operações:

```
//erro retorna 1 se ocorreu um erro e 0 caso contrário
void insereOrdenado(no *lista, char str[100],int erro); //insere a string str ordenada na lista
char *acessa(no *lista,int i,int erro); // retorna o i-ésimo elemento da lista
int busca(no *lista,char str[100],int erro); //retorna a posição relativa (primeiro, segundo, ...) do elemento que
contenha o texto str na lista. Retorna (-1) caso tal elemento não exista
```

2) Deseja-se criar uma **lista circular** heterogênea (de valores inteiros ou reais). Considere a estrutura abaixo e implemente as funções Cria, Esvazia, EntraInicio e SaiFim;

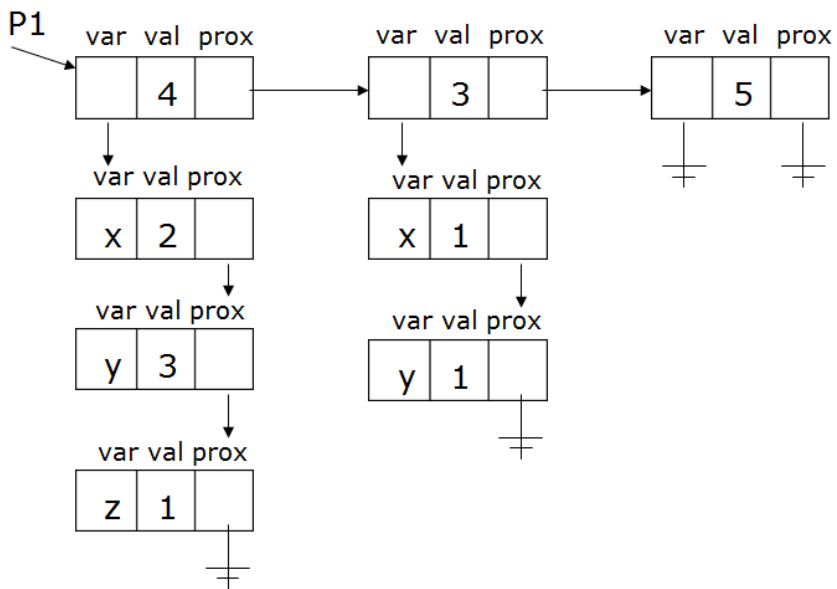
```
typedef union {
    int i;
    float r;
} elem;
typedef struct bloco{
    struct bloco *prox;
    elem x;
    enum tipo {
        inteiro, real
    } Tipo;
} no;
```

```
void Cria(no* lista, enum tipo* t);
void Esvazia(no* lista);
void EntraInicio(no* lista, elem* e, enum tipo* t);
void SaiFim(no* lista, elem* e, enum tipo* t);
```

3) Implemente um programa que lê um polinômio em uma lista generalizada e calcula o resultado do polinômio dado um valor para as variáveis **x, y, z, etc.**

Por exemplo:

Ex:  $P1 = 4x^2y^3z + 3xy + 5$



Para os valores de  $x = 1$   $y = 2$   $z = 3$ , o resultado retornado será 107.