

PUC-Rio – Programação 2 – INF1007
Prova 3 – Soluções – Turma 33C

- Questões 1a e 1b: (Isabel)

```
struct no {
    int pot;
    float coef;
    struct no *prox;
}
typedef struct no No;

No *lista_insere(No *lst, int pot, float coef) {
    No *novo;

    /* aloca um novo elemento (nó) da lista */
    novo = (No *) malloc(sizeof(No));
    novo->pot = pot;
    novo->coef = coef;

    /* testa se lista vazia: retorna o novo nó */
    if (lst == NULL) {
        novo->prox = NULL;
        return novo;
    }

    /* insere o novo nó no inicio da lista */
    novo->prox = lst;
    return novo;
}
```

- Questão 1c: (Maria Beatriz)

```
No *derivada(No *l) {
    int pot;
    float coef;
    No *aux, *d = NULL;

    /* lista vazia? */
    if (l == NULL) return NULL;

    /* percorre a lista calculando a derivada de cada termo e inserindo um nó
    correspondente na lista da derivada */
    aux = l;
    while (aux != NULL) {
        if (aux->pot != 0) {
            coef = aux->pot * aux->coef;
            pot = aux->pot - 1;
            d = lista_insere(d, pot, coef);
        }
        aux = aux->prox;
    }
    return d;
}
```

- Questão 2 (Leandro):

```

int palindromo(char *nome) {
    Pilha *aux = pilha_cria();
    char p;
    int i = 0;

    /* empilha todos os caracteres da cadeia */
    while (nome[i] != '\0') {
        pilha_push(aux, nome[i]);
        i++;
    }

    /* como a pilha a partir do topo tem a cadeia invertida,
       compara cada caractere da cadeia com o topo da pilha */

    i = 0;
    while (!pilha_vazia(aux)) {
        p = pilha_pop(aux);
        if (p != nome[i]) {
            pilha_libera(aux);
            return 0;
        }
        i++;
    }
    pilha_libera(aux);
    return 1;
}

```

- Questão 3: (Isabella)

```

void mostraPalindromo(NoArv *a, char *nome) {
    if (a == NULL) return;

    /* mostra os palindromos da subarvore da esquerda:
       todas as palavras nessa subarvore antecedem a da raiz */
    mostraPalindromo(a->esq, nome);

    /* raiz só interessa se sua palavra antecede à dada */
    if (strcmp(a->nome, nome) < 0) {

        /* e só imprime se é um palindromo */
        if (palindromo(a->nome))
            printf("%s\n", a->nome);

        /* se raiz antecede, há chance de palindromos na subárvore da direita */
        mostraPalindromo(a->dir, nome);
    }
}

```