Atividade 9 - Árvore Binária de Busca (T2)

12 de novembro de 2025

Problema A

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct no{
    int chave;
    struct no *e, *d;
} No;

No* insere(No *raiz, int chave){
    if(!raiz){
        No* novo_no = (No*)malloc(sizeof(No));

        novo_no->chave = chave;
        novo_no->e = NULL;
        novo_no->d = NULL;
        return novo_no;
    }

    if(chave > raiz->chave){
        raiz->d = insere(raiz->d, chave);
    } else if (chave < raiz->chave){
        raiz->e = insere(raiz->e, chave);
    }
    return raiz;
}

void imprimePreOrdem(No* raiz){
    if(!raiz) return;
    printf(" %d", raiz->chave);
    imprimePreOrdem(raiz->e);
    imprimePreOrdem(raiz->d);
}

void imprimeEmOrdem(No* raiz){
    if(!raiz) return;
    imprimeEmOrdem(raiz->e);
    printf(" %d", raiz->chave);
    imprimeEmOrdem(raiz->d);
}

void imprimePosOrdem(No* raiz){
    if(!raiz) return;
```

```
43
44      imprimePosOrdem(raiz->e);
45      imprimePosOrdem(raiz->d);
46      printf(" %d", raiz->chave);
47  }
48
49  void impressao(No* raiz){
50      printf("Pre.:"); imprimePreOrdem(raiz); printf("\n");
51      printf("In..:"); imprimeEmOrdem(raiz); printf("\n");
52      printf("Post:"); imprimePosOrdem(raiz); printf("\n\n");
53  }
54
55  int main(){
56      int c; scanf("%d", &c);
57      for(int caso = 1; caso <= c; caso++){
58          printf("Case %d:\n", caso);
59          int n; scanf("%d", &n);
60          No* raiz = NULL;
61
62          for(int i = 0; i < n; i++){
63              int num; scanf("%d", &num);
64              raiz = insere(raiz, num);
65          }
66          impressao(raiz);
67      }
68      return 0;
69  }
```

Problema B

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>

typedef long long ll;

typedef struct No{
    char c;
    struct No *e, *d;
} No;

bool sp = false;

No* insert(No* root, char carac){
    if(!root){
        No* no = (No*) malloc (sizeof(No));
        no->e = NULL;
        no->d = NULL;
        no->c = carac;
        return no;
    }

    if(carac > root->c){
        root->d = insert(root->d, carac);
    } else {
        root->e = insert(root->e, carac);
    }
    return root;
}

bool search(No* root, char x){
    if(!root) return false;
    if(root->c == x) return true;

    if(x > root->c){
        return search(root->d, x);
    } else return search(root->e, x);
}

void printInfix(No *root){
    if(!root) return;

    printInfix(root->e);

    if(!sp) sp = true;
    else printf(" ");

    printf("%c", root->c);

    printInfix(root->d);
}

void printPost(No *root){
    if(!root) return;

```

```
57      printPost(root->e);
58      printPost(root->d);
59
60      if(!sp) sp = true;
61      else printf(" ");
62
63      printf("%c", root->c);
64  }
65
66  void printPre (No *root){
67      if(!root) return;
68
69      if(!sp) sp = true;
70      else printf(" ");
71
72      printf("%c", root->c);
73
74      printPre(root->e);
75      printPre(root->d);
76  }
77
78  int main(){
79      No* root = NULL;
80
81
82      char *o = (char*) malloc (sizeof(char) * 50);
83
84      while(scanf("%s\n", o) != EOF){
85          if(strcmp(o, "I") == 0){
86              char x; scanf("%c\n", &x);
87              root = insert(root, x);
88          } else if (strcmp(o, "INFIXA") == 0){
89              sp = false;
90              printInfix(root);
91              printf("\n");
92          } else if (strcmp(o, "PREFIXA") == 0){
93              sp = false;
94              printPre(root);
95              printf("\n");
96          } else if (strcmp(o, "POSFIXA") == 0){
97              sp = false;
98              printPost(root);
99              printf("\n");
100         } else {
101             char x; scanf("%c\n", &x);
102             if(search(root, x)) printf("%c existe\n", x);
103             else printf("%c nao existe\n", x);
104         }
105     }
106
107     return 0;
108 }
```