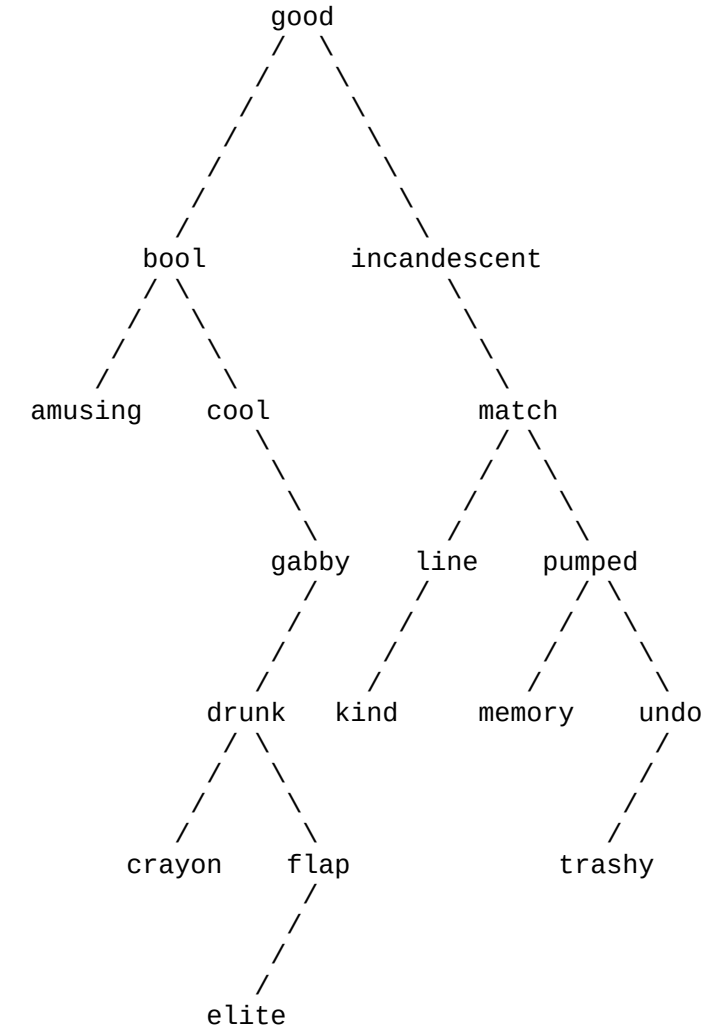
9.1. Pentru urmatoarea lista de cuvinte:
good, bool, cool, incandescent, gabby, drunk, match, pumped,
crayon, undo, flap, memory, line, trashy, elite, kind, amusing.
a) Generati Arborele Binar de Cautare
b) Scrieti parcurgerile: InOrdine, PreOrdine si PostOrdine
c) Descrieti succesiunea de stergere a nodului cu cheia drunk

//

a) Generati Arborele Binar de Cautare



//

b) Scrieti parcurgerile: InOrdine, PreOrdine si PostOrdine

```
  InOrdine: amusing,bool,cool,crayon,drunk,elite,flap,gabby,good,
            incandescent,kind,line,match,memory,pumped,trashy,undo
 PreOrdine: good,bool,amusing,cool,gabby,drunk,crayon,flap,elite,
            incandescent,match,line,kind,pumped,memory,undo,trashy
PostOrdine: amusing,crayon,elite,flap,drunk,gabby,cool,bool,kind,
            line,memory,trashy,undo,pumped,match,incandescent,good
```

//

c) Descrieti succesiunea de stergere a nodului cu cheia drunk

Pasul 1:
Se cauta succesorul InOrdine al nodului "drunk", acesta fiind "elite".
Pasul 2:
Se interschimba nodul "elite" cu nodul de sters "drunk".
Pasul 3:
Se sterge nodul.

//

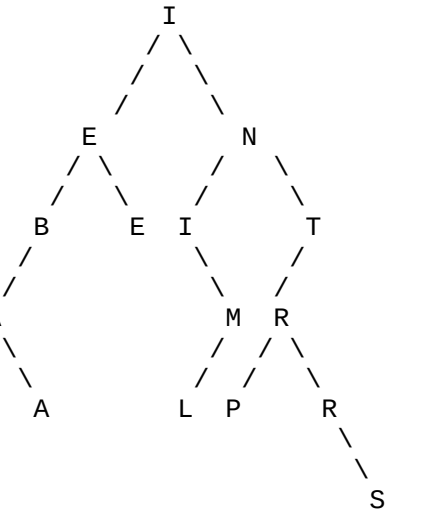
9.2. Pentru un Arbore Binar de Cautare avand cheia un numar intreg
(a) Scrieți o secventa de cod care returnează numărul elementelor (nodurilor) din arbore;
(b) Scrieți o secventa de cod care returnează suma tuturor cheilor din arbore;
(c) Scrieți o secventa de cod care returnează cheia cu valoarea maximă din arbore, sau -1 daca arborele este vid. Presupunem că toate valorile sunt pozitive;

//

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

typedef struct node{

    int n;
    struct node *next;

}node;

node *list=NULL;

//Subpunctul a;

int ContorTotal(node **list) {

    int contor = 0;
    if (*list == NULL) {
        return contor;
    }
    node *tmp = *list;
    while (tmp->next != NULL) {
        contor++;
        tmp = tmp->next;
    }

    return contor;
}

//Subpunctul b;

int SumaTotala(node **list) {

    int suma = 0;
    if (*list == NULL) {
```

```
        return suma;
    }
    node *tmp = *list;
    while (tmp->next != NULL) {
        suma += tmp->n;
        tmp = tmp->next;
    }
    return suma;

}

//Subpunctul c;

int MaximCheie(node **list) {

    int maxim = 0;
    if (*list == NULL) {
        return -1;
    }

    for (node *tmp = *list; tmp != NULL; tmp = tmp->next) {
        if (maxim < tmp->n) {
            maxim = tmp->n;
        }
        return maxim;
    }

}

int main(){

}

//////////////////////////////////////////////////////////////////////
```

9.3. Reprezentati Arborele Binar de Cautare pentru cheile reprezentand literele din sirul de caractere: INTREBARESIMPLA, in ordinea impusa considerand ca arborele este initial nul.



//