Examen ASD 2. Numele: _____Specializarea: _____ Nota Estimata: _____

1) (2p) Se da BSTNode un nod al unui arbore binar de cautare ce contine un membru te tip intreg. Scrieti codul unei functii noi care sa respecte urmatoarele specificatii. Metoda trebuie sa fie eficienta (nu vizitati noduri care nu sunt necesare)

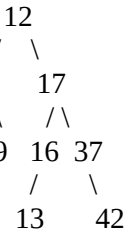
```
int predecesorulLui(BSTNode *nod)
// Preconditi: nod este adresa unui nod dintr-un arbore binar de cautare.
// Postconditi: returneaza valoarea nodului anterior, predecesorul nodului dat
```

2) (1p) Voi executa urmatorul cod, de 3 ori push si odata pop. Sa presupunem ca Q este reprezentata printr-o lista simplu inlantuita, Desenati starea cozii Q dupa executia codului de mai jos.

```
Queue Q = new Queue( );
Q.push(22);
Q.push(14);
Q.push(61);
Q.pop( );
```

3) (0.5p) Desenati un heap de maxim cu cel putin 13 noduri. Se va prezenta atat reprezentarea interna cat si sub forma de arbore.

4) (1p) Se da un arbore binar de cautare:



Parcurgeti arborele in preordine si afisati nodurile parcurse.

5) (1p) Care este inaltimea celui mai scund si celui mai inalalt arbore binar de cautare care poate fi construit cu N noduri distincte? Desenati exemple pentru a justifica raspunsul.

6) (1p) Care sunt pasii pentru a adauga un nod nou hash table? Descrieti in cuvinte fiecare pas.

7) (2p) Se da o secventa de intregi: 7, 2, 9, 3, 1, 6, 4, 5, 8
   a) Desenati arborele binar de cautare corespunzator secventei date.
   b) Desenati heapul de maxim corespunzator secventei date.

8) (1.5p) Scrieti codul sursa pentru functia “cateCheiPare” care returneaza numarul de noduri dintr-o lista circulara simplu inlantuita care contin chei pare. Nodul din lista are urmatoarea structura:

```
struct Node{
      int cheie;
      struct Node *next;
}
```

```
Sablonul functiei este
      int cateCheiPare(struct Node *head)
      // Preconditii: head este referinta catre primul nod din lista.
      // Lista poate sa fie vida sau nevida
      // Postconditii: Se returneaza numarul de noduri cu cheie para din lista
      // NOTA:Daca este vida lista se va returna 0
```