

## Računske vježbe 11

1. Napisati funkciju koja određuje broj čvorova drveta (težinu drveta).

```
struct drvo
{
    int a;
    struct drvo *left;
    struct drvo *right;
};

int tezina(struct drvo *root)
{
    if(root->left!=NULL && root->right!=NULL)
        return 1 + tezina(root->left) + tezina(root->right);
    else if(root->left!=NULL && root->right==NULL)
        return 1 + tezina(root->left);
    else if(root->left==NULL && root->right!=NULL)
        return 1 + tezina(root->right);
    else
        return 1;
}
```

2. Napisati funkciju koja određuje broj listova drveta.

```
int listovi(struct drvo *root)
{
    if(root->left!=NULL && root->right!=NULL)
        return listovi(root->left) + listovi(root->right);
    else if(root->left!=NULL)
        return listovi(root->left);
    else if(root->right!=NULL)
        return listovi(root->right);
    else
        return 1;
}
```

3. Svakom čvoru drveta pridružena je jedna riječ, dužine manje od 12 karaktera. Sastaviti funkciju koja određuje koliko puta se pojavljuje tražena riječ.

```
struct drvo
{
    char rijec[12];
    struct drvo *left;
    struct drvo *right;
};

int brojrijeci(struct drvo *root, char *s)
{
    int ind=0;
    if(strcmp(root->rijec,s)==0) ind=1;
    if(root->left!=NULL && root->right!=NULL)
        return ind + brojrijeci(root->right,s) + brojrijeci(root->left,s);
    else if(root->left!=NULL)
        return ind + brojrijeci(root->left,s);
    else if(root->right!=NULL)
        return ind + brojrijeci(root->right,s);
    else
        return ind;
}
```

4. Vršiti se obilazak binarnog drveta od korijena ka listovima i sabiraju se cijeli brojevi upisani u čvorovima drveta. Sastaviti funkciju koja pronalazi putanju sa najvećom sumom brojeva.

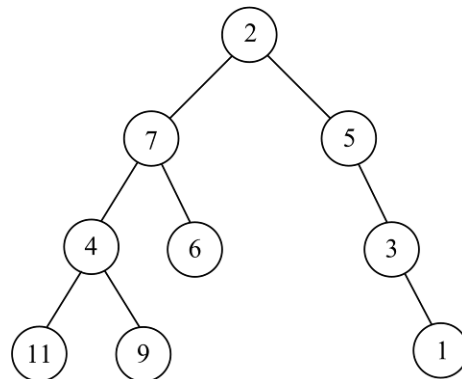
```
struct drvo
{
    int n;
    struct drvo *left;
    struct drvo *right;
};

int max_path(struct drvo *root)
{
    if(root->left!=NULL && root->right!=NULL)
        return root->n + max(max_path(root->left), max_path(root->right));
    else if(root->left!=NULL)
        return root->n + max_path(root->left);
    else if(root->right!=NULL)
        return root->n + max_path(root->right);
    else
        return root->n;
}

int max(int a, int b)
{
    return (a > b) ? a : b;
}
```

5. Dato je binarno drvo u čijim su čvorovima upisani cijeli brojevi. Odrediti redosljed štampanja tih brojeva ukoliko koristimo:

- inorder obilazak (lijevo poddrvo, korijen, desno poddrvo)
- preorder obilazak (korijen, lijevo poddrvo, desno poddrvo)
- postorder obilazak (lijevo poddrvo, desno poddrvo, korijen).



- 11, 4, 9, 7, 6, 2, 5, 3, 1
- 2, 7, 4, 11, 9, 6, 5, 3, 1
- 11, 9, 4, 6, 7, 1, 3, 5, 2