Άσκηση 1. *Στην άσκηση αυτή πρέπει να υλοποιήσετε ένα δυαδικό δέντρο αναζήτησης. Το δυαδικό δέντρο αναζήτησης πρέπει να έχει την εξής διασύνδεση (**bst.h**):*

```
#ifndef _BST_H
#define _BST_H

typedef long bstdataT;                      // type for data
typedef long bstkeyT;                       // type for key

typedef struct bst_rep* bst;                // binary search tree (bst)
typedef struct bstnode_rep* bstnode;        // bst node

bst bst_create();                           // create an empty tree
void bst_destroy( bst B );                  // free memory of bst

void bst_insert( bst, bstkeyT, bstdataT );  // insert element in bst
bstnode bst_search( bst, bstkeyT );         // search key in bst

bstdataT bst_data( bstnode );               // get data of bst node
void     bst_setdata( bstnode, bstdataT );  // set data of bst node
bstkeyT  bst_key( bstnode );                // get key of  node

void bst_inorder( bst, void (*)(bstnode) ); // traverse bst inorder and run
                                            // function for each node
#endif
```

*Για λόγους απλότητας δεν περιέχονται πολλές βασικές συναρτήσεις όπως η διαγραφή στοιχείου, η επιστροφή του μεγέθους του δέντρου, κ.τ.λ. Γράψτε την υλοποίηση σας στο αρχείο **bst.c**.*

Άσκηση 2. *Χρησιμοποιώντας το δέντρο δυαδικής αναζήτησης της προηγούμενης άσκησης πρέπει να γράψετε ένα πρόγραμμα σε C που να υπολογίζει ιστογράμματα. Ονομάστε το αρχείο σας **histogram.c**. Το πρόγραμμα σας πρέπει να διαβάζει από την τυπική είσοδο ακέραιους αριθμούς τύπου long (χωρισμένους με whitespace) και να μετράει τον αριθμό των φορών που κάθε αριθμός έχει εμφανιστεί. Η έξοδος πρέπει να ακολουθεί την εξής μορφή:*
• κάθε γραμμή πρέπει να περιέχει έναν αριθμό και την συχνότητα εμφάνισης του αριθμού αυτού, χωρισμένα με κενό 2
• οι γραμμές πρέπει να είναι ταξινομημένες σε αύξουσα σειρά όσο αναφορά τους αριθμούς (όχι τις συχνότητες).

Για παράδειγμα στην είσοδο:

4 5 4 1 1 1 1 1 7 12 12 12

το πρόγραμμα σας πρέπει να τυπώσει:

1 5
4 2
5 1
7 1
12 3