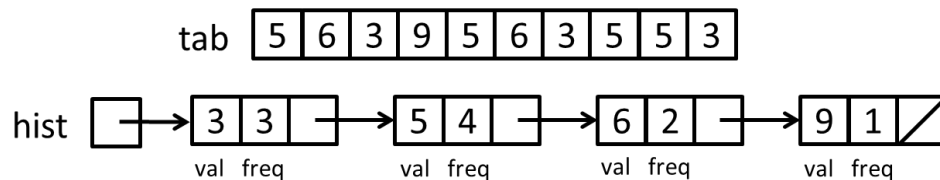


Exercise I

A histogram is a presentation of data that shows the frequencies with which each value occurs in an array. We consider integer values.

Example



1. Define the data type "cel" to represent a histogram as a linked list of values (val) and frequencies (freq).
2. Write the "histo" which, given an array of values, returns its histogram. Note that the given array is not sorted; however, the resulting histogram must be sorted according to ascending values.
3. Write the function "histoTest" that demonstrates the use of the function "histo".

Exercise II

Consider the data types Point3d, Point2d and Node, defined below. A binary file named "points3d" contains data of type point3d.

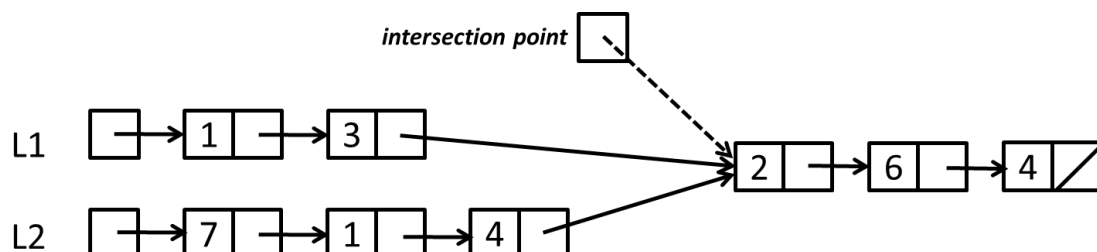
```
typedef struct Point3d { double x , y , z ; } point3d ;
typedef struct Point2d { double x , y ; } point2d ;
typedef struct Node { point2d p ; struct Node * next ; } node ;
```

4. Write the function "file2list" which reads the file "points3d" and copies its content in a linked list of type node *. Obviously, the z-coordinate will be neglected. The order of points in the file should be maintained in the list. The file and the list must be traversed (iterated), each, only once.

Exercise III

5. Given two linked lists of integers, write the function "intPoint" which finds and returns their intersection point. Note that this is not the set-intersection but common nodes.

Example



The End