Hacettepe University
Department of Computer Science & Engineering

BiL235 Programming/Software Laboratory
Experiment II

Subject	: Linked List
Date Due	: November 14, 2011
Programming Language	: ANSI-ISO C (GCC 4.1.2 20080704 (Red Hat 4.1.2-50))
Advisors	: R.A. Tahir BIÇAKCI, Dr. Sevil Şen

I. INTRODUCTION

This assignment is designed to improve your knowledge on linked list algorithms and to help you practice with pointers in C. For this purpose, you are expected to develop a system which finds the silhouette of a city.

Linked List:

In computer science, a linked list is a data structure that consists of a sequence of data records such that in each record there is a field that contains a reference (*i.e.*, a link) to the next record in the sequence [1, 2]. They occur often enough to have a place in our work on science, mathematics and information theory.

Silhouette of buildings:

The silhouette of a city is shown in the Figure 1. The 2-dimensional view of a city is given on the left side of the Figure 1. On the right side of the figure, the silhouette of the city is demonstrated.

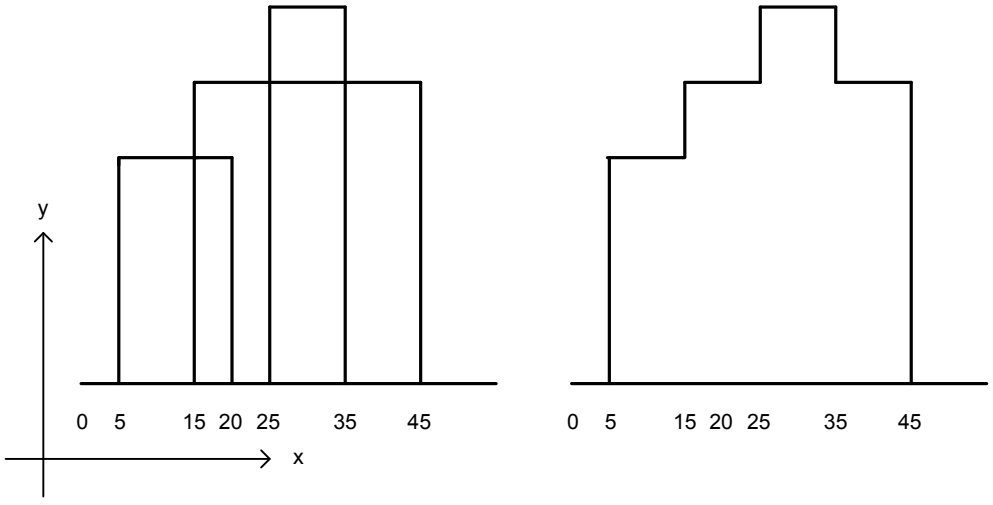

Figure 1. Two dimensional view of a city and its silhouette

II. PROBLEM

In this experiment, you are supposed to develop a program which gives the silhouette of a city given. You also need to take into account the streets as the third dimension. The three dimensional view of a city is shown in Figure 2. (x, y and z—>street).

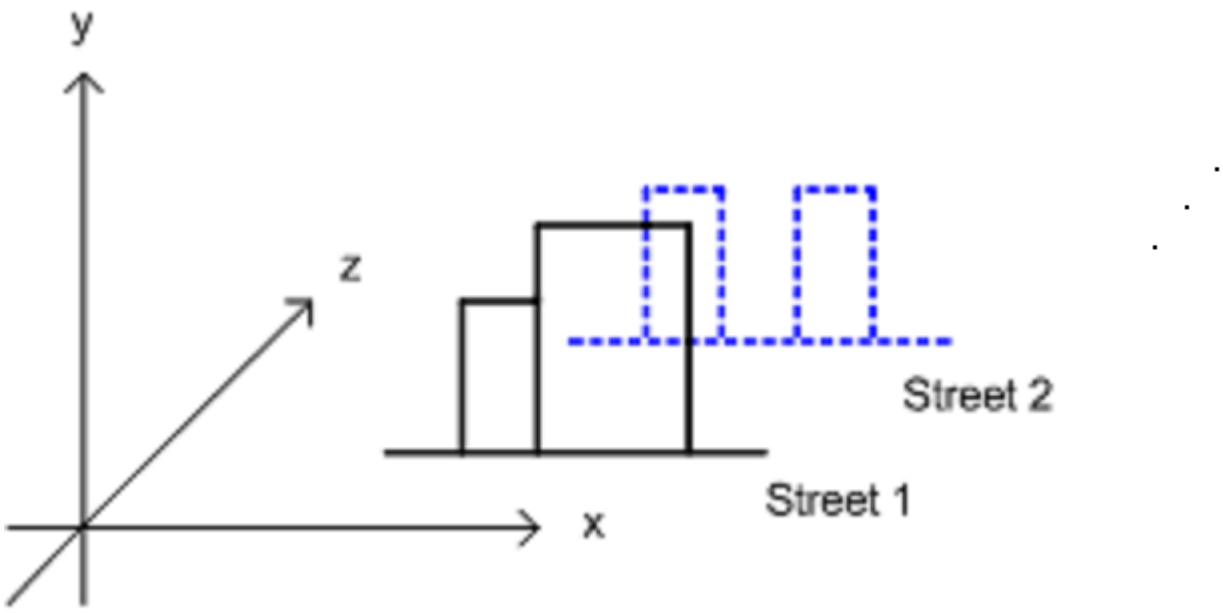

Figure 2. Three-dimensional of a city

When the city is looked from the "Street 1", the silhouette covers all possible buildings (buildings on both Street 1 and Street 2). But when we look from the "Street 2", the silhouette covers only the buildings on the "Street 2". For example, when we look towards to the Street 2 from a point between the Street 1 and Street 2, only dashed lines can be seen.

III. DESIGN PROPERTIES

- You must consider each building as an object.
- You can't clone any building.
- Use the linked list structures to hold buildings and to solve the problems.
- The number of streets is dynamic.
- According to the input, the buildings on the same street do not cross one another, they can only be contiguous.
- There might be no buildings on a street.

Your work has to be compiled in the system dev.cs.hacettepe.edu.tr . The compiler installed on the system is GCC 4.1.2 20080704 (Red Hat 4.1.2-50).

IV. INPUT AND OUTPUT FORMAT

1. Input file

Input file will be error free. The commands will be given any order.

```
insert <street no> (<Left> <Height> <Right>) (<Left> <Height> <Right>)
draw <street no>
remove <street no> (<Left> <Height> <Right>)
...
```

List 1. Input file format

insert: This command represents the buildings on the street. "street no" is the number of the street. "(<Left> <Height> <Right>)" symbolizes a building, "Left"

and "Right" are left and right coordinates, respectively. "Height" is the height of a building.

draw: With this command you must draw the silhouette of city by using "street no". For example "draw 4" command tells you to draw the silhouette of the city from the point of view of street 4. In other words, you have to draw the buildings on the streets 4, 5, 6, ... and n. "draw 1" command tells you to draw all buildings.

remove: This command removes the building given with its number from the city. If the given building number cannot be found in the list, you should not give any error message. Undefined messages will prevent to evaluate your work automatically.

```
İnsert 1 (0 10 3) (3 5 6) (10 12 11)
draw 1
draw 1
remove 1 (3 5 6)
insert 3 (2 3 5) (12 20 20) (6 3 12) (21 2 25)
draw 1
draw 2
insert 2 (1 5 7)
draw 3
remove 2 (1 5 7)
remove 2 (1 5 7)
draw 1
draw 40
```

List 2. A sample input file

2. Output file

Your application will be executed by a Linux script, and evaluated automatically. It is important to apply the rules of output format. **Misformatted outputs will not be evaluated. There will be no toleration in the evaluation process.** The format file and sample file are given below.

For every "draw" command, you must write the silhouette of the city to the output file. The output list contains only the starting point of each horizontal line. There will be no comma between the coordinates.

```
<x-coordinate> <height> <x-coordinate> <height> <x-coordinate> <height> <x-coordinate> <height>
```

List 3. Output file format

```
0 10 3 5 6 0 10 12 11 0
0 10 3 5 6 0 10 12 11 0
0 10 3 3 5 0 6 3 10 12 11 3 12 20 20 0 21 2 25 0
2 3 5 0 6 3 12 20 20 0 21 2 25 0
2 3 5 0 6 3 12 20 20 0 21 2 25 0
0 10 3 3 5 0 6 3 10 12 11 3 12 20 20 0 21 2 25 0
0 0
```

List 4. A sample output file

V. LAST REMARKS:

- You will use online submission system to submit your experiments. https://submit.cs.hacettepe.edu.tr/ (No other submission method such as diskette, CD or email will be accepted)
- Your submission code file structure must implement this template:

```
<student_id>
      <src>
      <report>
```

- Submission time for deadline is: 17:00.
- Regardless of the length, use **UNDERSTANDABLE** names to your variables, procedures and functions.
- Do not submit any file via e-mail related with this assignment.
- **SAVE** all your work until the assignment is graded.
- The assignment must be original, **INDIVIDUAL** work. Duplicate or very similar assignments are both going to be punished. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms or source codes.
- You can ask your questions through course's news group: news://news.cs.hacettepe.edu.tr/dersler.bil235 And you are supposed to be aware of everything discussed in the newsgroup.

VI. REFERENCES:

1. Ellis Horowitz, Fundamentals of Data Structures
2. http://en.wikipedia.org/wiki/Linked_list