

# EE 453/717: Advanced Computing for Electrical Engineers

Instructors: Saravanan Vijayakumaran, Sachin Patkar

Indian Institute of Technology Bombay

Autumn 2011

Assignment 2 : 15 points

Due date: August 29, 2011

1. Write a C++ function to detect the existence of a loop in a singly linked list using only a pointer to beginning of the list. The function should have the signature `void detectLoop(node* head);`. Start with the program available at <http://www.ee.iitb.ac.in/~sarva/EE453717/Autumn2011/loop.cpp>. The program contains a `main()` function which constructs a linked list using the following input.

- **Input:** The `main()` function reads the following from the console
  - A line containing a positive integer  $N$  corresponding to the number of nodes in the singly linked list. The line ends with a newline character. There will be  $N$  nodes in the list which will be numbered 1 to  $N$ . The numbering will be used to specify the links in the linked list (see below).
  - $N$  lines each containing a pair of nonnegative integers separated by a space. Each number in the pair is less than or equal to  $N$ . The line ends with a newline character. A pair  $i j$  implies that node  $i$  points to node  $j$ . If  $j$  is zero, then node  $i$  points to NULL.

Example: The following console input corresponds to the linked list in Figure 1. Note that it contains the loop  $2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2$ .

```
5
1 2
5 4
4 3
2 5
3 2
```

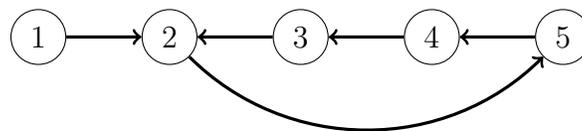


Figure 1: A linked list with a loop

The `loop.cpp` file has an empty `detectLoop` function. You have to complete its implementation such that it gives the following output. **Please upload the completed `loop.cpp` file to the online judge.**

- **Output:** Output the number 1 to the console followed by a newline character if a loop exists. Otherwise output a 0 followed by a newline character.

2. Write a C++ program which labels the edge pixels of the components in an image with integers starting from 2. A pixel which is part of a component is not an edge pixel if it is in the interior of the component, i.e. it has component pixels to the north, south, east, and west of itself.

- **Input:** Read the following from the console
  - A line containing a positive integer  $N$  corresponding to the number of rows and columns in the image. The image is assumed to be square. The line ends with a newline character.
  - $N$  lines containing a sequence of  $N$  bits separated by spaces. Each line ends with a newline character. The ones correspond to component pixels and the zeros correspond to blank pixels.

An example input will look like the following.

```
7
0 0 1 1 0 0 0
0 0 1 1 1 0 0
0 0 1 1 1 0 0
0 0 0 1 1 0 0
1 1 0 0 1 0 0
1 1 1 0 0 0 0
1 1 1 0 0 0 0
```

- **Output:**  $N$  lines each containing  $N$  integers separated by spaces. Each line ends with a newline character. The edge pixels of a component are represented by the same integer. The first component which is encountered when the pixels are scanned by rows from left to right is labeled with the integer 2. The next component is labeled with the integer 3 and so on.

```
0 0 2 2 0 0 0
0 0 2 0 2 0 0
0 0 2 0 2 0 0
0 0 0 2 2 0 0
3 3 0 0 2 0 0
3 0 3 0 0 0 0
3 3 3 0 0 0 0
```