/\* This program swaps the nodes of linked list rather

than swapping the field from the nodes.\*/

#include<stdio.h>

#include<stdlib.h>

/\* A linked list node \*/

struct Node

{

int data;

struct Node \*next;

};

/\* Function to swap nodes x and y in linked list by

changing links \*/

void swapNodes(struct Node \*\*head\_ref, int x, int y)

{

// Nothing to do if x and y are same

if (x == y) return;

// Search for x (keep track of prevX and CurrX

struct Node \*prevX = NULL, \*currX = \*head\_ref;

while (currX && currX->data != x)

{

prevX = currX;

currX = currX->next;

}

// Search for y (keep track of prevY and CurrY

struct Node \*prevY = NULL, \*currY = \*head\_ref;

while (currY && currY->data != y)

{

prevY = currY;

currY = currY->next;

}

// If either x or y is not present, nothing to do

if (currX == NULL || currY == NULL)

return;

// If x is not head of linked list

if (prevX != NULL)

prevX->next = currY;

else // Else make y as new head

\*head\_ref = currY;

// If y is not head of linked list

if (prevY != NULL)

prevY->next = currX;

else // Else make x as new head

\*head\_ref = currX;

// Swap next pointers

struct Node \*temp = currY->next;

currY->next = currX->next;

currX->next = temp;

}

/\* Function to add a node at the begining of List \*/

void push(struct Node\*\* head\_ref, int new\_data)

{

/\* allocate node \*/

struct Node\* new\_node =

(struct Node\*) malloc(sizeof(struct Node));

/\* put in the data \*/

new\_node->data = new\_data;

/\* link the old list off the new node \*/

new\_node->next = (\*head\_ref);

/\* move the head to point to the new node \*/

(\*head\_ref) = new\_node;

}

/\* Function to print nodes in a given linked list \*/

void printList(struct Node \*node)

{

while(node != NULL)

{

printf("%d ", node->data);

node = node->next;

}

}

/\* Druver program to test above function \*/

int main()

{

struct Node \*start = NULL;

/\* The constructed linked list is:

1->2->3->4->5->6->7 \*/

push(&start, 7);

push(&start, 6);

push(&start, 5);

push(&start, 4);

push(&start, 3);

push(&start, 2);

push(&start, 1);

printf("\n Linked list before calling swapNodes() ");

printList(start);

swapNodes(&start, 4, 3);

printf("\n Linked list after calling swapNodes() ");

printList(start);

return 0;

}