#include <stdio.h>

#include <string.h>

#include <stdlib.h>

typedef struct NODE {

int data;

struct NODE \*next;

};

/\* --------------------- Node Creation ------------------------ \*/

NODE \*makeNode(int key)

{

NODE \*newNode = (NODE\*)malloc(sizeof(NODE));

if (!newNode)

return NULL;

newNode->data = key;

newNode->next = NULL;

return newNode;

}

//adds a new item to the end of the list

NODE \* add\_last(NODE \*head, NODE\* newItem)

{

NODE \*currItem;

if (!head)

return newItem;

currItem = head;

while (currItem->next)

currItem = currItem->next;

currItem->next = newItem;

return head;

}

NODE \*add\_middle(NODE \*head, NODE\* newItem)

{

NODE \*currItem;

if (!head)

return newItem;

currItem = head;

while (currItem->next) {

if (currItem->data < newItem->data && currItem->next->data > newItem->data)

{

newItem->next = currItem->next;

currItem->next = newItem;

}

currItem = currItem->next;

}

return head;

}

NODE \*splitListToEven(NODE \*head, NODE \*lstEven)

{

NODE \*currItem;

NODE \*currItemeven = NULL;

if (!head)

return NULL;

currItem = head;

while (currItem) {

if (currItem->data % 2 == 0)

{

if (lstEven == NULL)

{

lstEven = makeNode(currItem->data);

currItemeven = lstEven;

}

else

{

currItemeven->next = makeNode(currItem->data);

currItemeven = currItemeven->next;

}

}

currItem = currItem->next;

}

return lstEven;

}

NODE \*splitListToOdd(NODE \*head, NODE \*lstOdd)

{

NODE \*currItem;

NODE \*currItemodd = NULL;

if (!head)

return NULL;

currItem = head;

while (currItem) {

if (currItem->data % 2 == 1)

{

if (lstOdd == NULL)

{

lstOdd = makeNode(currItem->data);

currItemodd = lstOdd;

}

else

{

currItemodd->next = makeNode(currItem->data);;

currItemodd = currItemodd->next;

}

}

currItem = currItem->next;

}

return lstOdd;

}

int main() {

struct NODE \*head;

struct NODE \*lstEven = NULL;

struct NODE \*lstOdd = NULL;

struct NODE \*temp\_node;

struct NODE \*middle;

head = makeNode(0);

middle = makeNode(3);

for (int i = 2; i < 8; i += 2)

{

temp\_node = makeNode(i);

add\_last(head, temp\_node);

}

add\_middle(head, middle);

middle = makeNode(5);

add\_middle(head, middle);

NODE \*temp = head;

while (temp) {

printf("NODE DATA : %d\n", temp->data);

temp = temp->next;

}

lstEven = splitListToEven(head,lstEven);

lstOdd = splitListToOdd(head, lstOdd);

NODE \*temp\_even = lstEven;

NODE \*temp\_odd = lstOdd;

while (temp\_even) {

printf("EVEN NODE DATA : %d\n", temp\_even->data);

temp\_even = temp\_even->next;

}

while (temp\_odd) {

printf("ODD NODE DATA : %d\n", temp\_odd->data);

temp\_odd = temp\_odd->next;

}

getchar();

return 0;

}