/* A Linked-List is an ordered set of data elements, each containing at least one data element and a link to its next node. Following the algorithm given on the right, write a C Program that will create a Linked-List, add nodes to the Linked-List and finally display the Linked-List nodes on the screen. */

#include <stdio.h> // needed to print anything on the screen
#include <malloc.h> // .h file to allocate additional memory when new node is created

void main() // main function returns nothing
{
    struct node // structure creates, adds and displays a node
    {
        int num; // any variable
        struct node *ptr; // pointer to the next node
    }
    typedef struct node NODE; // NODE declared to be of type struct node

    NODE *head, *first, *temp = 0; // *head, *first, *temp, variables of type struct node
    int count = 0; // just a counter
    int choice = 1; // choice variable
    first = 0; // points to root

    while (choice) // while choice is 1, keep creating new node
    {
        // head points to the beginning of the allocated new memory
        head = (NODE *)malloc(sizeof(NODE));

        // data entered & assigned to num in struct node
        printf("Enter the data item\n");
        scanf("%d", &head->num); // value entered from keyboard

        if (first != 0) // if the initial node exists
            {temp->ptr = head; // head point to allocated memory & assigned to temp->ptr
                temp = head; // head is stored in a temporary location called temp
            } else // if first == 0, no node exists, hence create a new node
                {first=temp=head; // first, temp & head points to nowhere (or zero memory)
                }

        fflush(stdin); // whatever entered from keyboard, display it on the screen

        printf("Do you want to continue(Type 0 or 1)?\n"); // continue to enter new node
        scanf("%d", &choice); // assign entered value to choice
    }

    temp->ptr = 0; // temp->ptr initially points to nowhere

    temp = first; // reset temp to the beginning, identified by first

    printf("\nstatus of the linked list is\n");

    while (temp != 0) // while temp points somewhere as beginning of allocated memory
    {
        printf("%d=>", temp->num); // print the value entered from the keyboard
        count++; // increment counter
        temp = temp->ptr; // store temp->ptr in a temporary location
    }

    printf("NULL\n"); // print null
    printf("No. of nodes in the list = %d\n", count);