ASSIGNMENT 2 - C CODE

Akhil P Dominic
MT23013

3. Write a c program tail -n which will print last n lines of the input. The program should behave rationally no matter how much the value of n should be. Do not store the lines in 2-dimentional arrays of fixed sizes.
For correct code and execution

tail.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct node
{
   char *cur_line;
   struct node *next;
};

struct node *root = NULL;
struct node *cur = NULL;

void insert_node(char *line)
{
   if (root == NULL)
   {
      struct node *new_node = (struct node *)malloc(sizeof(struct node *));
      new_node->cur_line = strdup(line);
      new_node->next = NULL;
      root = new_node;
   }
   else
   {
      struct node *new_node = (struct node *)malloc(sizeof(struct node *));
      new_node->cur_line = strdup(line);
      new_node->next = NULL;

      cur = root;
      while (cur->next != NULL)
      {
```

```
            // cout<<cur->cur_line<<endl;
            cur = cur->next;
        }
        cur->next = new_node;
    }
}

void printnlines(struct node *Node, int n)
{
    int counter = 0;
    cur = root;
    while (cur != NULL)
    {
        cur = cur->next;
        counter = counter + 1;
    }
    cur = root;
    int i = 0;
    while (i < (counter - n))
    {
        cur = cur->next;
        i++;
    }

    while (cur != NULL)
    {
        printf("%s \n", cur->cur_line);
        cur = cur->next;
    }
}

int main(int argc, char *argv[])
{
    if (argc < 2){
        printf("Error, no arg value passed\n");
        printf("Correct usage: tail -n number");
        return -1;
    }

    size_t d=100;
    char *input = (char*)(malloc(d));
    int inp_len;
    printf("Enter the input : \n");
```

```
    while((getdelim(&input, &d, '\n', stdin) > 1)){
            size_t len = strlen(input);
             insert_node(input);
    }

    // cout<<"Enter value for n : ";
    int n = atoi(argv[2]);
    // cin>>n;
            printf("\n Last %d lines are : \n",n);

    printnlines(root, n);
}
```