Add linked list

```
#include <stdio.h>
#include <stdlib.h>
#include “function.h"

int main() {
    Node *head_a = NULL, *head_b = NULL;
    int data;
    while(1) { // read List a
        scanf("%d", &data);
        if (data >= 0) {
          Create_List(&head_a, data);
        } else break;
    }
    while(1) { // read List b
        scanf("%d", &data);
        if (data >= 0) {
          Create_List(&head_b, data);
        } else break;
    }
    Node *head = Add_List( head_a, head_b );
    Print_List( head_a );
    Print_List( head_b );
    Print_List( head );
    Free_List( head_a );
    Free_List( head_b );
    Free_List( head );
    return 0;
}
// function.h
typedef struct _Node {
    int data;
    struct _Node *next;
} Node;

void Create_List(Node** head, int data) {
    if (*head == NULL) {
        *head = malloc(sizeof(Node));
        (*(*head)).data = data;
    }
    else {
        Node *prev;
        Node *current = *head;
        while (current != NULL) {
            prev = current;
            current = current->next;
        }
        prev->next = malloc(sizeof(Node));
        prev->next->data = data;
    }
}
int Length(Node* head) {
    int len = 0;
    Node* y = head;
    while (y != NULL) {
        len++;
        y = y->next;
    }
    return len;
```

```
}
Node* Add_List(Node* head_a, Node* head_b){
  Node *a = head_a;
  Node *b = head_b;
  Node *head = malloc(sizeof(Node));
  int len_a = Length(a);
  int len_b = Length(b);
  if (len_a > len_b) {
    // case: longer list + shorter list
    // loop shorter length
    Node* current = head;
    for(int i=1; i<=len_a; i++){
      if(a->next != 0){
        current->next = malloc(sizeof(Node));
      }
        current->data = a->data;
        current = current->next;
        a = a->next;
    }
    current = head;
    for(int i=1; i<=len_b; i++){
      current->data += b->data;
      current = current->next;
      b = b->next;
    }
    return head;
  }
  else {
    Node* current = head;
    for(int i=1; i<=len_b; i++){
      if(b->next != 0){
        current->next = malloc(sizeof(Node));
      }
        current->data = b->data;
        current = current->next;
        b = b->next;
    }
    current = head;
    for(int i=1; i<=len_a; i++){
      current->data += a->data;
      current = current->next;
      a = a->next;
    }
    return head;
  }
}

void Print_List(Node* head){
  Node* current = head;
  while (current->next != NULL){
      printf("%d->", current->data);
      current = current->next;
  }
  printf("%d\n", current->data);
}
// This function is used to free the memory space.
void Free_List(Node* head){
  free(head);
}
```

the great depression

```
#include <stdio.h>
#include <stdbool.h>

typedef struct _Factory {
char name[21];  // factory name
//    int a;          // profit of producing car A
//    int b;          // profit of producing car B
int x;          // x = a-b
} Factory;

int main() {
    // init n, x, y
    int n;
    int x;
    int y;
    scanf("%d %d %d", &n, &x, &y);

    // init factories
    Factory factories[n];
    for (int i=0; i<n; i++) {
        int a, b;
        scanf("%s %d %d", factories[i].name, &a, &b);
        factories[i].x = a-b;
    }

    // sort factories by x
    for (int i=0; i<n-1; i++) {
        for (int j=0; j<n-1; j++) {
            if (factories[j].x < factories[j+1].x) {
                // swap
                Factory tmp = factories[j];
                factories[j] = factories[j+1];
                factories[j+1] = tmp;
            }
        }
    }

    // sort factories by name
    for (int i=0; i<x-1; i++) {
        for (int j=0; j<x-1; j++) {
            int k = 0;
            while (true) {
                if (factories[j].name[k] > factories[j+1].name[k]) {
                    // swap
                    Factory tmp = factories[j];
                    factories[j] = factories[j+1];
                    factories[j+1] = tmp;
                    break;
                }
                else if (factories[j].name[k] == factories[j+1].name[k]) k++;
                else break;
            }
        }
    }

    // print top x
    for (int i=0; i<x; i++) printf("%s\n", factories[i].name);
}
```

the monkey

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

typedef struct node{
    char input;
    struct node* prev;
    struct node* next;
} Node;

Node* load(char ch, Node* node){
    if(node->next){
        Node *letter = malloc(sizeof(Node*));
        letter->input = ch;
        letter->next = node->next;
        letter->prev = node;
        node->next = letter;
        return letter;
    } else {
        Node *letter = malloc(sizeof(Node*));
        letter->input = ch;
        letter->next = NULL;
        letter->prev = node;
        node->next = letter;
        return letter;
    }
}

void fresh(Node* node){
    node->input = '\0';
    node->next = NULL;
    node->prev = NULL;
    return;
}

int main(void){
    char ch;
    Node *output = malloc(sizeof(Node*));
    Node *cur_node = output;
    fresh(cur_node);
    while((ch = getchar()) != '\n'){
        switch(ch){
            case '>':
                if(cur_node->next) cur_node = cur_node->next;
                break;
            case '<':
                if(cur_node->prev) cur_node = cur_node->prev;
                break;
            case 'D':
                if(cur_node->next!=NULL){
                    cur_node = cur_node->next;
                    cur_node->prev->next = cur_node->next;
                    if(cur_node->next) cur_node->next->prev = cur_node->prev;
                    Node *tmp = cur_node;
                    cur_node = cur_node->prev;
                    free(tmp);
                }
                break;
            case EOF:
```

```
            break;
        default:
            cur_node = load(ch, cur_node);
        }
      if(ch == EOF) break;
  }
  while(output = output->next) putchar(output->input);
  putchar('\n');
  return 0;
}


/*
input

int<<<h>>>_used<D_list<<<<linked_

output

hint_use_linked_list
*/


/*
10 5 5
TOYOTA 10 100
GM 20 90
FORD 30 80
Volkswagen 40 70
Daimler 50 60
Honda 60 50
Nissan 70 40
PEUGEOT 80 30
FIAT 90 20
BMW 100 10


BMW
FIAT
Honda
Nissan
PEUGEOT
*/
```