ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД

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

typedef struct MusicalComposition {
    char* name;
    char* author;
    int year;
    struct MusicalComposition* prev;
    struct MusicalComposition* next;
} MusicalComposition;

MusicalComposition* createMusicalComposition(char* name, char* author, int
year) {
    MusicalComposition* ret =
(MusicalComposition*)malloc(sizeof(MusicalComposition));

    ret->name = (char*)malloc(strlen(name));
    strcpy(ret->name, name);

    ret->author = (char*)malloc(strlen(author));
    strcpy(ret->author, author);

    ret->year = year;
    ret->prev = NULL;
    ret->next = NULL;

    return ret;
}

MusicalComposition* createMusicalCompositionList(char** array_names, char**
array_authors, int* array_years, int n) {
    MusicalComposition* head =
(MusicalComposition*)malloc(sizeof(MusicalComposition));
    head = createMusicalComposition(array_names[0], array_authors[0],
array_years[0]);

    MusicalComposition* prev =
(MusicalComposition*)malloc(sizeof(MusicalComposition));
    prev = head;
    MusicalComposition* cur =
(MusicalComposition*)malloc(sizeof(MusicalComposition));
    for (int i = 1; i != n; ++i) {
        cur = createMusicalComposition(array_names[i], array_authors[i],
array_years[i]);
        cur->prev = prev;
        cur->prev->next = cur;
        cur->next = NULL;
        prev = cur;
    }

    return head;
}

void push(MusicalComposition* head, MusicalComposition* element) {
    if (head == NULL) {
        head = element;
        return;
    }

    MusicalComposition* cur = head;

    while (cur->next) { cur = cur->next; }
    element->prev = cur;
    cur->next = element;
    element->next = NULL;
}

int count(MusicalComposition* head) {
    int ret = 0;

    while (head != NULL) {
        ++ret;
        head = head->next;
    }

    return ret;
}

void print_names(MusicalComposition* head) {
    MusicalComposition* cur = head;

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

void swap_halfs(MusicalComposition** head) {
    if (head == NULL || *head == NULL) { return; }

    int list_len = count(*head);
    int half_index = list_len / 2;
    MusicalComposition* cur = *head;
    MusicalComposition* new_head = NULL;

    for (int i = 0; i != half_index; ++i) { cur = cur->next; }
    cur->prev->next = NULL;
    cur->prev = NULL;
    new_head = cur;

    for (int i = half_index; i != list_len - 1; ++i) { cur = cur->next; }
    cur->next = *head;
    (*head)->prev = cur;

    *head = new_head;
}

int main(){
    int length;
    scanf("%d\n", &length);

    char** names = (char**)malloc(sizeof(char*)*length);
    char** authors = (char**)malloc(sizeof(char*)*length);
    int* years = (int*)malloc(sizeof(int)*length);

    for (int i=0;i<length;i++) {
        char name[80];
        char author[80];

        fgets(name, 80, stdin);
        fgets(author, 80, stdin);
        fscanf(stdin, "%d\n", &years[i]);

        (*strstr(name,"\n"))=0;
        (*strstr(author,"\n"))=0;

        names[i] = (char*)malloc(sizeof(char*) * (strlen(name)+1));
        authors[i] = (char*)malloc(sizeof(char*) * (strlen(author)+1));
        strcpy(names[i], name);
        strcpy(authors[i], author);
    }

    MusicalComposition* head = createMusicalCompositionList(names, authors,
years, length);

    print_names(head);
    printf("\n");
    swap_halfs(&head);
    print_names(head);

    return 0;
}
```