

Зв'язні списки



Лекція №7

Дисципліна «Програмування»

2-й семестр



Представлення даних

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

#define TSIZE    45    // розмір масиву для зберігання назви
#define FMAX     5    // максимальна кількість назв фільмів

struct Film
{
    char title[TSIZE];
    int rating;
};

char *s_gets(char *st, int n);

int main(void)
{
    struct Film movies[FMAX];
    int i = 0;
    int j;
```



Представлення даних

```
SetConsoleCP(1251);
SetConsoleOutputCP(1251);

puts("Введіть назву першого фільму:");

while(i < FMAX && s_gets(movies[i].title, TSIZE) != NULL
    && movies[i].title[0] != '\0')
{
    puts("Введіть своє значення рейтингу <0-10>:");
    scanf("%d", &movies[i++].rating);

    while(getchar() != '\n')
        continue;

    puts("Введіть назву наступного фільму\n"
        "(або порожній рядок для припинення вводу):");
}

if(i == 0) printf("\nДані не введені.");
else printf("\nСписок фільмів:\n");
```

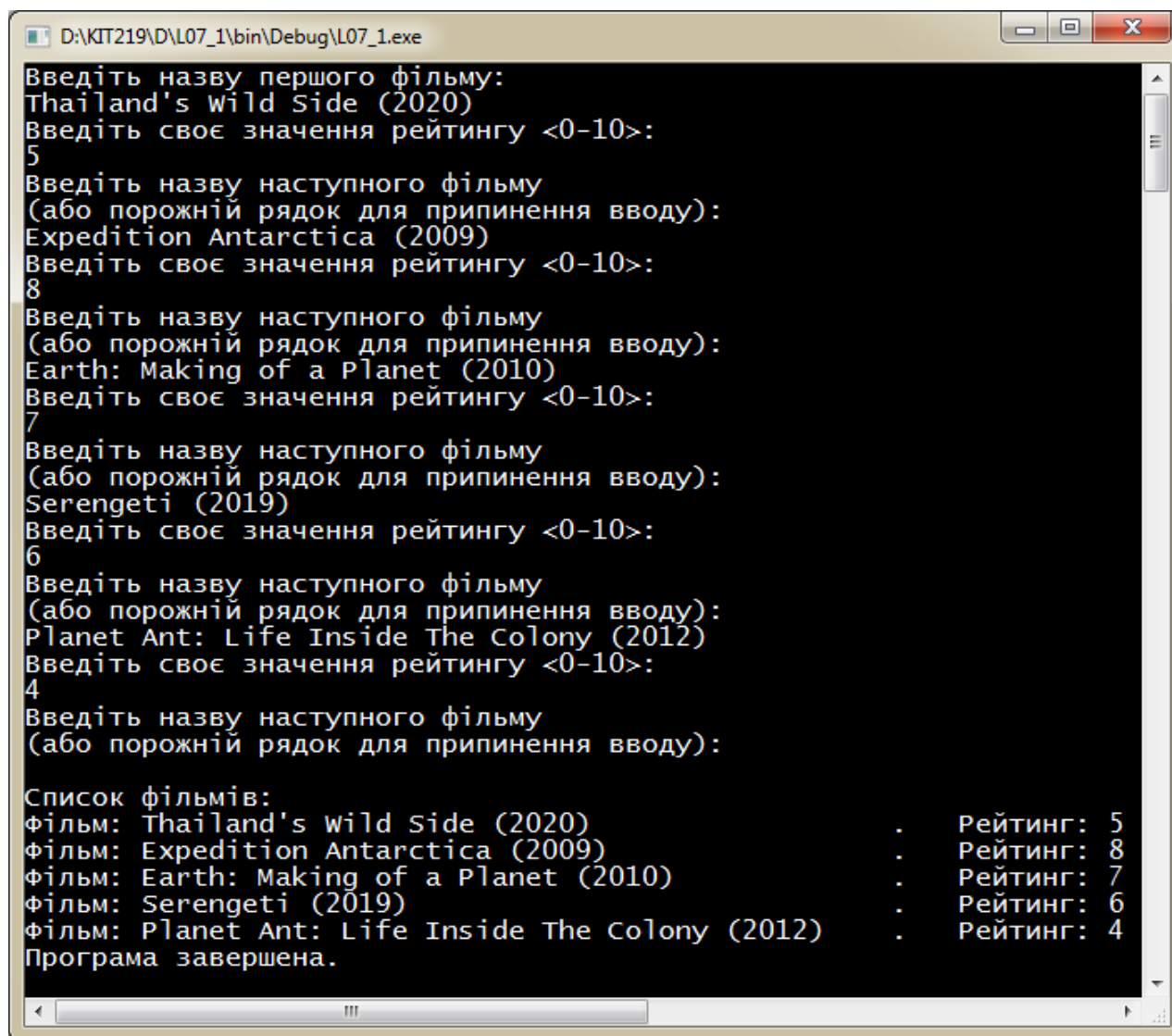


Представлення даних

```
for(j = 0; j < i; j++)
    printf("Фільм: %s.    Рейтинг: %d\n",
           movies[j].title, movies[j].rating);
printf("Програма завершена.\n");
return 0;
}

char *s_gets(char *st, int n)
{
    char *ret_val;
    char *find;
    ret_val = fgets(st, n, stdin);
    if(ret_val)
    {
        find = strchr(st, '\n');
        if(find) *find = '\0';
        else while(getchar() != '\n') continue;
    }
    return ret_val;
}
```

Представлення даних



```
D:\KIT219\D\L07_1\bin\Debug\L07_1.exe
Введіть назву першого фільму:
Thailand's Wild Side (2020)
Введіть своє значення рейтингу <0-10>:
5
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Expedition Antarctica (2009)
Введіть своє значення рейтингу <0-10>:
8
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Earth: Making of a Planet (2010)
Введіть своє значення рейтингу <0-10>:
7
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Serengeti (2019)
Введіть своє значення рейтингу <0-10>:
6
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Planet Ant: Life Inside The Colony (2012)
Введіть своє значення рейтингу <0-10>:
4
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):

Список фільмів:
Фільм: Thailand's Wild Side (2020) . Рейтинг: 5
Фільм: Expedition Antarctica (2009) . Рейтинг: 8
Фільм: Earth: Making of a Planet (2010) . Рейтинг: 7
Фільм: Serengeti (2019) . Рейтинг: 6
Фільм: Planet Ant: Life Inside The Colony (2012) . Рейтинг: 4
Програма завершена.
```



Динамічний розподіл пам'яті

```
#define TSIZE 45 // розмір масиву для зберігання назви
struct Film
{
    char title[TSIZE];
    int rating;
};
...
int n, i;
struct Film *movies; // вказівник на структуру
...
printf("Вкажіть максимальну кількість фільмів: \n");
scanf("%d", &n);
movies = (struct Film *) malloc(n * sizeof(struct Film));
```

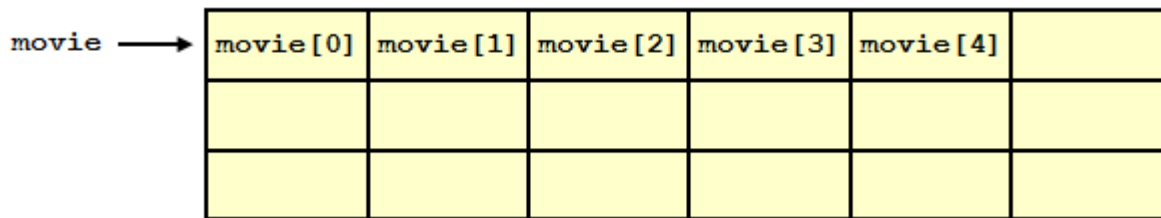
Вказівник **movies** можна застосовувати таким чином, ніби він є ім'ям масиву:

```
while (i < FMAX && gets(movies[i].title) != NULL &&
        movies[i].title[0] != '\0')
```

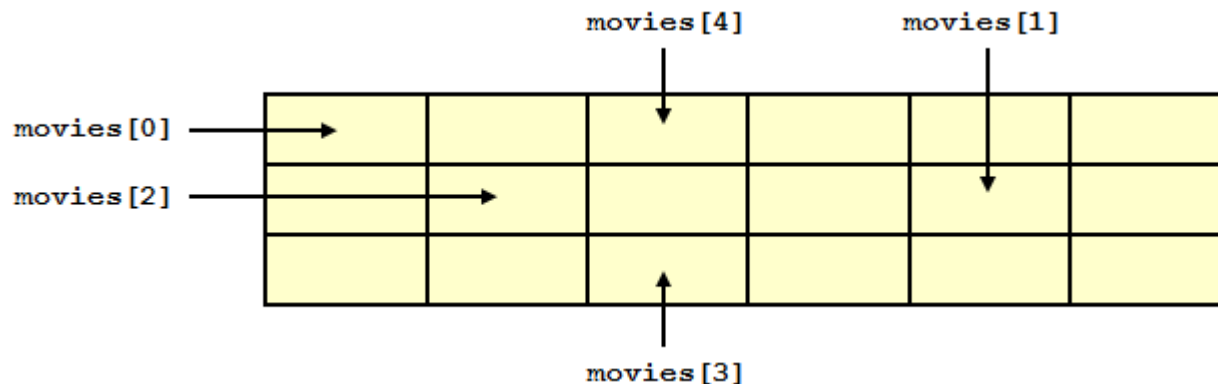


Виділення пам'яті під структури

```
struct Film *movie;  
movie = (struct Film *) malloc(5 * sizeof(struct Film));
```



```
int i;  
struct Films *movies[5];  
for(i = 0; i < 5; i++)  
    movies[i] = (struct Films *) malloc(sizeof(struct Films));
```





Створення масиву вказівників

```
#define TSIZE 45          // розмір масиву для збереження назв
#define FMAX 500         // максимальна кількість назв фільмів

struct Film
{
    char title[TSIZE];
    int rating;
};

...

struct Film *movies[FMAX]; // масив вказівників на структури
int i;

...

movies[i] = (struct Film *) malloc(sizeof(struct Film));
```




Перехід до зв'язного списку

При кожному виклику функції `malloc()` для виділення пам'яті під нову структуру одночасно можна виділяти пам'ять і для нового вказівника.

Перевизначимо структуру таким чином, щоб вона містила вказівник на наступну структуру. Тоді при кожному створенні нової структури її адресу можна буде зберігати в попередній структурі.

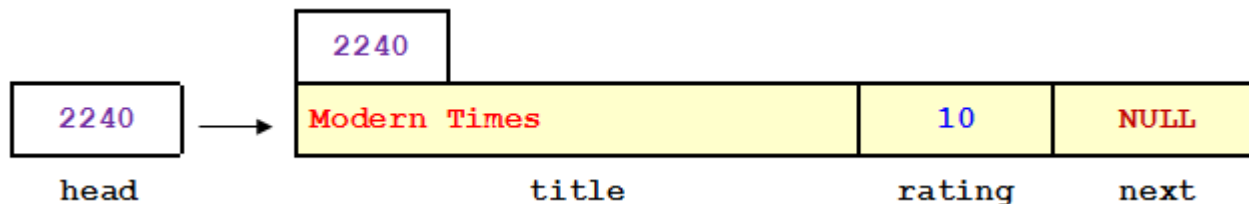
```
#define TSIZE 45    // розмір масиву для зберігання назв

struct Film
{
    char title[TSIZE];
    int rating;
    struct Film *next;
};
```



Елемент зв'язного списку

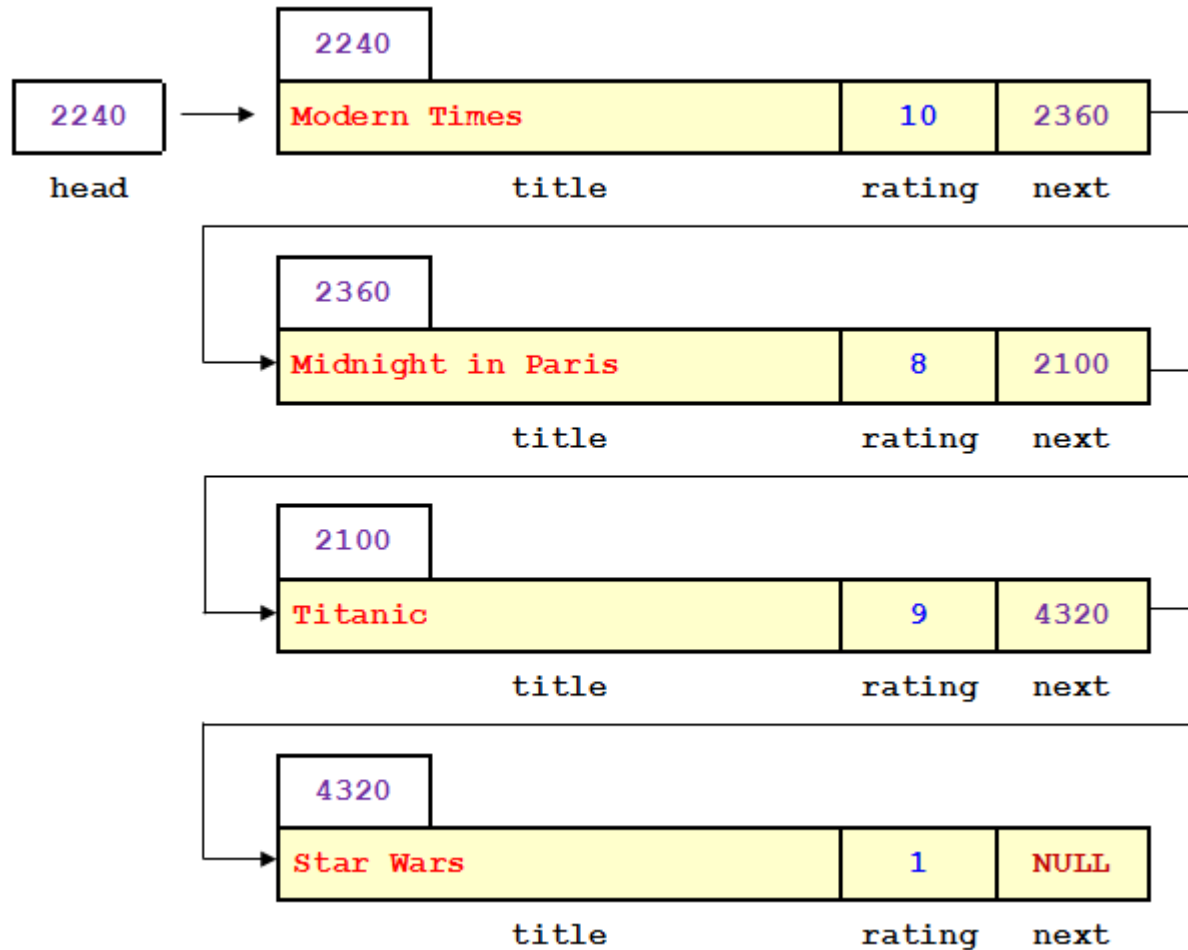
```
#define TSIZE 45           // розмір масиву для зберігання назв  
struct Film  
{  
    char title[TSIZE];    // назва фільму  
    int rating;             // рейтинг  
    struct Film *next;     // вказівник на наступну структуру  
};                          // зв'язного списку  
struct File *head;        // вказівник на заголовок списку
```



Перший елемент у зв'язному списку



Елементи зв'язного списку



Зв'язний список з чотирма елементами



Використання зв'язного списку

```
#include <stdio.h>
#include <windows.h>
#include <stdlib.h>           // містить прототип функції malloc()
#include <string.h>          // містить прототип функції strcpy()

#define TSIZE 45             // розмір масиву для зберігання назв

struct Film
{
    char title[TSIZE];
    int rating;
    struct Film *next;       // вказує на наступну структуру
};                           // в списку

char *s_gets(char *st, int n);
```



Використання зв'язного списку

```
int main(void)
{
    struct Film *head = NULL;
    struct Film *prev, *current;
    char input[TSIZE];

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    // Збирання та збереження інформації
    puts("Введіть назву першого фільму: ");
    while(s_gets(input, TSIZE) != NULL && input[0] != '\0')
    {
        current = (struct Film *) malloc(sizeof(struct Film));
        if(head == NULL) // перша структура
            head = current;
        else // наступні структури
            prev->next = current;
        current->next = NULL;
    }
}
```



Використання зв'язного списку

```
strcpy(current->title, input);
puts("Введіть своє значення рейтингу <0-10>:");
scanf("%d", &current->rating);

while(getchar() != '\n')
    continue;
puts("Введіть назву наступного фільму\n"
      "(або порожній рядок для припинення вводу):");
prev = current;
}

// Відображення списку фільмів
if(head == NULL)
    printf("Дані не введені.");
else
    printf("Список фільмів:\n");
current = head;
```



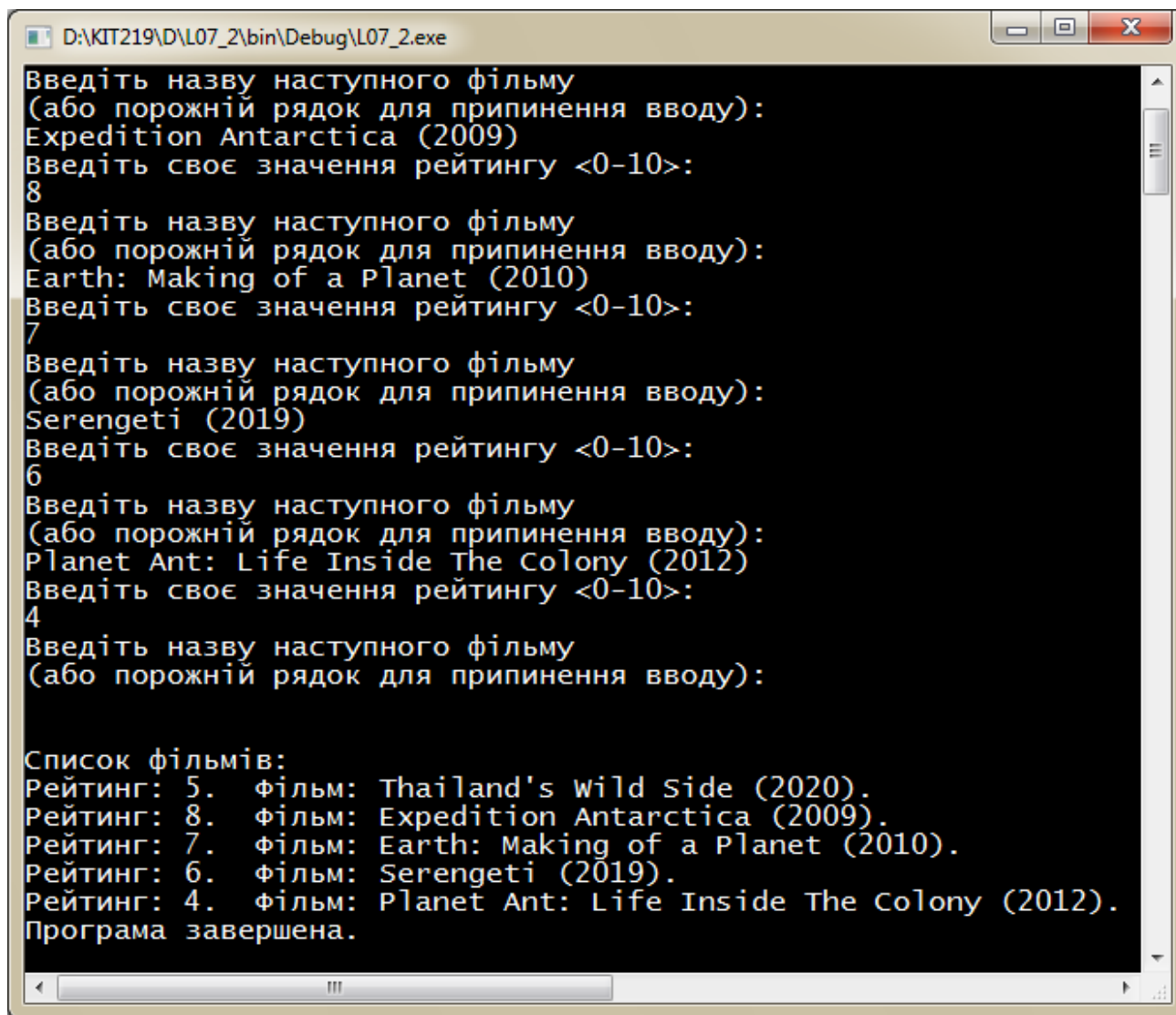
Використання зв'язного списку

```
while (current != NULL)  
{  
    printf("Рейтинг: %d. Фільм: %s. \n",  
          current->rating, current->title);  
    current = current->next;  
}
```

```
// Програма виконана, тому можна звільнити пам'ять  
current = head;
```

```
while (current != NULL)  
{  
    free(current);  
    current = current->next;  
}  
printf("Програма завершена.\n");  
return 0;  
}
```

Використання зв'язного списку



```
D:\KIT219\D\L07_2\bin\Debug\L07_2.exe
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Expedition Antarctica (2009)
Введіть своє значення рейтингу <0-10>:
8
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Earth: Making of a Planet (2010)
Введіть своє значення рейтингу <0-10>:
7
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Serengeti (2019)
Введіть своє значення рейтингу <0-10>:
6
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):
Planet Ant: Life Inside The Colony (2012)
Введіть своє значення рейтингу <0-10>:
4
Введіть назву наступного фільму
(або порожній рядок для припинення вводу):

Список фільмів:
Рейтинг: 5. фільм: Thailand's wild Side (2020).
Рейтинг: 8. фільм: Expedition Antarctica (2009).
Рейтинг: 7. фільм: Earth: Making of a Planet (2010).
Рейтинг: 6. фільм: Serengeti (2019).
Рейтинг: 4. фільм: Planet Ant: Life Inside The Colony (2012).
Програма завершена.
```




Вивід списку на екран

Встановлюємо вказівник **current** таким чином, щоб він посилався на першу структуру, тобто на **head**:

```
current = head;
```

Потім за допомогою форми запису з вказівником звертаємося до членів цієї структури:

```
printf("Рейтинг: %d. Фільм: %s. \n",  
       current->rating, current->title);
```

Вказівник **current** встановлюємо на наступну структуру:

```
current = current->next;
```

Цей процес необхідно повторити в циклі, поки вказівник **current** не буде встановлено в **NULL**:

```
while(current != NULL)  
{  
    printf("Рейтинг: %d. Фільм: %s. \n",  
           current->rating, current->title);  
    current = current->next;  
}
```



Створення списку

Створення списку передбачає виконання наступних дій:

- використання функції `malloc()` для виділення достатнього простору під зберігання структури;
- збереження адреси структури;
- копіювання до структури коректної інформації.

Не має сенсу створювати структуру, якщо вона поки непотрібна, тому для прийому від користувача інформації про назву фільму в програмі застосовується тимчасове сховище (масив `input`).

Якщо користувач відтворює за допомогою клавіатури символ `EOF` або вводить порожній рядок, цикл вводу завершується:

```
while(s_gets(input, TSIZE) != NULL && input[0] != '\0')
```



Створення списку

Далі програма запитує простір для структури і присвоює її адресу змінній типу вказівника **current**:

```
current = (struct Film *) malloc(sizeof(struct Film));
```

Адреса самої першої структури повинна бути збережена у змінній типу вказівника **head**.

Адреса кожної наступної структури повинна зберігатися в члені **next** попередньої структури.

Необхідно вказівнику **head** присвоїти значенням **NULL** на початку програми. Потім можна використовувати значення вказівника **head** для прийняття рішення про подальші дії:

```
if (head == NULL)                // перша структура
    head = current;
else                             // наступні структури
    prev->next = current;
```

prev – вказівник на попередню структуру.



Створення списку

Далі член структури **next** повинен бути встановлений в **NULL** для позначення того, що поточна структура є останньою в списку. Назву фільму з масиву **input** треба скопіювати до члену структури **title** і отримати значення для члену **rating**:

```
current->next = NULL;  
strcpy(current->title, input);  
puts("Введіть своє значення рейтингу <0-10>:");  
scanf("%d", &current->rating);
```

Потім треба підготувати програму до наступного циклу вводу. Зокрема, вказівник **prev** необхідно встановити так, щоб він посилався на поточну структуру, оскільки після вводу назви наступного фільму та розподілу наступної структури поточна структура стане попередньою:

```
prev = current;
```



Звільнення пам'яті

Для того, щоб звільнити всю пам'ять, яка була виділена для зв'язного списку, необхідно в циклі застосувати функцію `free()` до кожної виділеної структури:

```
current = head;
while (current != NULL)
{
    free(current);
    current = current->next;
}
```

Спочатку вказівник `current` встановлюється на початок зв'язного списку: `current = head;`

Потім в циклі `while`, поки не знайдемо останню структуру (`current != NULL`), видаляємо пам'ять під поточну структуру: `free(current);`

та встановлюємо поточний вказівник на наступну структуру: `current = current->next;`