

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include "stdio.h"
3 #include "stdlib.h"
4 #include "malloc.h"
5 #include "string.h"
6
7
8 typedef struct Film
9 {
10     int durata;
11     char* titlu;
12 }Film;
13
14 Film creareFilm(int durata, const char* titlu)
15 {
16     Film f;
17     f.durata = durata;
18     f.titlu = (char*)malloc(sizeof(char) * (strlen(titlu) + 1));
19     strcpy(f.titlu, titlu);
20     return f;
21 }
22
23
24 void afisareFilm(Film film)
25 {
26     printf("\nDurata filmului %s este %d",
27         film.titlu, film.durata);
28 }
29
30
31 typedef struct Nod
32 {
33     Film info;
34     struct Nod* next;
35     struct Nod* prev;
36 }Nod;
37
38
39 Nod* creareNod(Film f, Nod* next, Nod* prev)
40 {
41     Nod* nod = (Nod*)malloc(sizeof(Nod));
42     nod->info = creareFilm(f.durata, f.titlu);
43     nod->next = next;
44     nod->prev = prev;
45     return nod;
46 }
47
48 typedef struct DLL
49 {
```

```
50     Nod* prim;
51     Nod* ultim;
52 }DLL;
53
54 DLL inserareInceput(DLL lista, Film f)
55 {
56     Nod* nou = creareNod(f, NULL, NULL);
57     if (lista.prim)
58     {
59         nou->next = lista.prim;
60         lista.prim->prev = nou;
61         lista.prim = nou;
62     }
63     else
64     {
65         lista.prim = lista.ultim = nou;
66     }
67     return lista;
68 }
69
70 DLL inserareFinal(DLL lista, Film f)
71 {
72     Nod* nou = creareNod(f, NULL, NULL);
73     if (lista.ultim)
74     {
75         nou->prev = lista.ultim;
76         lista.ultim->next = nou;
77         lista.ultim = nou;
78     }
79     else
80     {
81         lista.prim = lista.ultim = nou;
82     }
83     return lista;
84 }
85
86 void afisareInceputFinal(DLL lista)
87 {
88     if (lista.prim)
89     {
90         Nod* aux = lista.prim;
91         while (aux)
92         {
93             afisareFilm(aux->info);
94             aux = aux->next;
95         }
96     }
97 }
98
```

```
99 void afisareFinalInceput(DLL lista)
100 {
101     if (lista.ultim)
102     {
103         Nod* aux = lista.ultim;
104         while (aux)
105         {
106             afisareFilm(aux->info);
107             aux = aux->prev;
108         }
109     }
110 }
111
112 void dezaolocareListaDubla(DLL* lista)
113 {
114     if ((*lista).prim)
115     {
116         while ((*lista).prim)
117         {
118             Nod* aux = (*lista).prim;
119             (*lista).prim = (*lista).prim->next;
120             free(aux->info.titlu);
121             free(aux);
122         }
123         (*lista).ultim = NULL;
124     }
125 }
126
127
128 DLL inserareInceputListaDublaCirculara(DLL lista, Film f)
129 {
130     Nod* nou = creareNod(f, NULL, NULL);
131     if(lista.prim) {
132         nou->next = lista.prim;
133         lista.prim->prev = nou;
134         lista.ultim->next = nou;
135         nou->prev = lista.ultim;
136         lista.prim = nou;
137     }
138     else {
139         lista.prim = lista.ultim = nou;
140     }
141     return lista;
142 }
143 //lista=insterare(lista)
144 //inserarelista(&lista)//in functie ai * lista
145 DLL inserareFinalListaDublaCirculara(DLL lista, Film f)
146 {
147     Nod* nou = creareNod(f, NULL, NULL);
```

```
148     if (lista.ultim) {
149         nou->next = lista.prim;
150         lista.ultim->next = nou;
151         lista.prim->prev = nou;
152         nou->prev = lista.ultim;
153         lista.ultim = nou;
154     }
155     else {
156         lista.prim = lista.ultim = nou;     nou.prev=nou;
157     }                                       nou.next=nou;
158     return lista;
159 }
160
161 void afisareListaDublaCircularaInceputFinal(DLL cap)
162 {
163     if (cap.prim) {
164         afisareFilm(cap.prim->info);
165         Nod* aux = cap.prim->next;
166         while (aux != cap.prim) {
167             afisareFilm(aux->info);
168             aux = aux->next;
169         }
170     }
171 }
172
173 void afisareListaDublaCircularaFinalInceput(DLL cap) {
174     if (cap.ultim) {
175         afisareFilm(cap.ultim->info);
176         Nod* aux = cap.ultim->prev;
177         while (aux != cap.ultim) {
178             afisareFilm(aux->info);
179             aux = aux->prev;
180         }
181     }
182 }
183
184 void main()
185 {
186     DLL lista;
187     lista.prim = lista.ultim = NULL;
188     //200 120 200 180 120
189
190     lista = inserareInceput(lista, creareFilm(200, "Film1"));
191     lista = inserareInceput(lista, creareFilm(120, "Film2"));
192     lista = inserareInceput(lista, creareFilm(200, "Film3"));
193     lista = inserareFinal(lista, creareFilm(180, "Film4"));
194     lista = inserareFinal(lista, creareFilm(120, "Film5"));
195
196     printf("\n Afisare lista inceput -> final: ");
```

```
197     afisareInceputFinal(lista);
198     printf("\n Afisare lista final -> inceput: ");
199     afisareFinalInceput(lista);
200
201     DLL list;
202     list.prim = list.ultim = NULL;
203     list = inserareInceputListaDublaCirculara(list, creareFilm(200, "film1"));
204     list = inserareInceputListaDublaCirculara(list, creareFilm(150, "film2"));
205     list = inserareInceputListaDublaCirculara(list, creareFilm(180, "film3"));
206     list = inserareInceputListaDublaCirculara(list, creareFilm(260, "film4"));
207     list = inserareInceputListaDublaCirculara(list, creareFilm(300, "film5"));
208
209     printf("\n Afisare lista dubla circulara inceput - final: ");
210     afisareListaDublaCircularaInceputFinal(list);
211
212     printf("\n Afisare lista dubla circulara final - inceput: ");
213     afisareListaDublaCircularaFinalInceput(list);
214 }
```