

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include "stdio.h"
3 #include "stdlib.h"
4 #include "malloc.h"
5 #include "string.h"
6
7 typedef struct Student
8 {
9     int varsta;
10    char* nume;
11    float medie;
12 }Student;
13
14 struct Student creareStudent(int v, const char* n, float m)
15 {
16     struct Student s;
17     s.varsta = v;
18     s.nume = (char*)malloc(sizeof(char) * (strlen(n) + 1));
19     strcpy(s.nume, n);
20     s.medie = m;
21     return s;
22 }
23
24 void afisareStudent(struct Student s)
25 {
26     printf("\n Studentul cu varsta %d are numele %s si media %5.2f",
27         s.varsta, s.nume, s.medie);
28 }
29
30
31 typedef struct Node
32 {
33     struct Student info;
34     struct Node* next;
35 }Node;
36
37 struct Node* creareNod(struct Student info, struct Node* next)
38 {
39     struct Node* nou = (struct Node*)malloc(sizeof(struct Node));
40     nou->info = creareStudent(info.varsta, info.nume, info.medie);
41     nou->next = next;
42     return nou;
43 }
44
45
46 struct Node* inserareInceput(struct Node* cap, struct Student s)
47 {
48     struct Node* nou = creareNod(s, NULL);
49     if (cap)
```

```
50     {
51         nou->next = cap;
52         cap = nou;
53     }
54     else {
55         cap = nou;
56     }
57     return cap;
58 }
59
60 struct Node* inserareFinal(struct Node* cap, struct Student s)
61 {
62     struct Node* nou = creareNod(s, NULL);
63     if (cap) {
64         struct Node* p = cap;
65         while (p->next) {
66             p = p->next;
67         }
68         p->next = nou;
69     }
70     else {
71         cap = nou;
72     }
73     return cap;
74 }
75
76 void afisareLista(struct Node* cap)
77 {
78     if (cap) {
79         struct Node* p = cap;
80         while (p) {
81             afisareStudent(p->info);
82             p = p->next;
83         }
84     }
85 }
86
87 struct Node* stergereLista(struct Node* cap)
88 {
89     if (cap) {
90         while (cap) {
91             struct Node* aux = cap;
92             cap = (cap)->next;
93             free(aux->info.ume);
94             free(aux);
95         }
96     }
97     return cap;
98 }
```

```
99 //se modifica in timp real
100 void dezaolocareLista(struct Node** cap)
101 {
102     if (*cap) {
103         while (*cap) {
104             struct Node* aux = *cap;
105             *cap = (*cap)->next;
106             free(aux->info.nume);
107             free(aux);
108         }
109     }
110 }
111 // " BT"
112
113 void afisareListaCirculara(Node* head)
114 {
115     if (head) {
116         afisareStudent(head->info);
117         Node* p = head->next;
118         while (p != head)
119         {
120             afisareStudent(p->info);
121             p = p->next;
122         }
123     }
124 }
125
126 Node* inserareInceputListaCirculara(Node* head, Student s)
127 {
128     Node* nou = creareNod(s, NULL);
129     if (head) {
130         nou->next = head;
131         Node* p = head;
132         while (p->next != head) {
133             p = p->next;
134         }
135         p->next = nou;
136         head = nou;
137     }
138     else {
139         nou->next = nou;
140         head = nou;
141     }
142     return head;
143 }
144
145 Node* inserareLaFinalListaCirculara(Node* head, Student s)
146 {
147     Node* nou = creareNod(s, NULL);
```

```
148     if (head) {
149         nou->next = head;
150         Node* p = head;
151         while (p->next != head) {
152             p = p->next;
153         }
154         p->next = nou;
155     }
156     else {
157         nou->next = nou;
158         head = nou;
159     }
160     return head;
161 }
162
163 void main()
164 { //18 13 23 20 16 22
165     struct Node* cap = NULL;
166     cap = inserareInceputListaCirculara(cap, creareStudent(20, "abc", 10));
167     cap = inserareInceputListaCirculara(cap, creareStudent(23, "bca", 9));
168     cap = inserareInceputListaCirculara(cap, creareStudent(13, "cba", 8));
169     cap = inserareInceputListaCirculara(cap, creareStudent(18, "cab", 7));
170     cap = inserareLaFinalListaCirculara(cap, creareStudent(16, "Marcel", 5));
171     cap = inserareLaFinalListaCirculara(cap, creareStudent(22, "RT", 6));
172     afisareListaCirculara(cap);
173     //dezalocareLista(&cap);
174     printf("\n afisare lista dupa dezalocare");
175     //afisareLista(cap);
176     //cap = stergereLista(cap);
177 }
178
179
180
```