

```

1 // Pilha Dinâmica
2
3 #include<stdio.h>
4 #include<stdlib.h>
5 #include<string.h>
6
7
8 typedef struct livro
9 {
10     char nome[30];
11     int pag;
12 }Livro;
13
14 typedef struct Celula
15 {
16     Livro dados;
17     struct Celula *next;
18 }CELULA;
19
20 void inicializar(CELULA **pilha)
21 {
22     (*pilha)= NULL;
23 }
24
25 int Vazia(CELULA **pilha)
26 {
27     if((*pilha) == NULL)
28     {
29         printf("Pilha Vazia\n");
30         return 1;
31     }
32     return 0;
33 }
34
35
36 CELULA *criarCelula()
37 {
38     CELULA * nova = (CELULA*)malloc(sizeof(CELULA));
39     return nova;
40 }
41
42
43 int push(CELULA**pilha,Livro elemento)
44 {
45     CELULA *nova;
46     nova = criarCelula();
47     if(nova==NULL)
48     {
49         printf("Memória Cheia!\n");
50         return 0;
51     }
52     if(Vazia(pilha))
53     {
54         nova->dados = elemento;
55         nova->next = NULL;
56         (*pilha)=nova;
57         return 1;
58     }
59     nova->dados = elemento;
60     nova->next = (*pilha);
61     (*pilha) = nova;
62     return 1;
63 }
64
65 Livro pop(CELULA**pilha)
66 {
67     Livro removido;
68     CELULA * aux;
69     strcpy(removido.nome, "Inválido");
70     removido.pag = -1;
71     if(Vazia(pilha))
72     {
73         printf("Pilha Vazia\n");

```

```

74         return removido;
75     }
76     aux = (*pilha);
77     removido = aux->dados;
78     (*pilha) = (*pilha)->next;
79     free(aux);
80     return removido;
81 }
82
83 Livro topo(CELULA **pilha)
84 {
85     Livro topo;
86     CELULA * aux;
87     if(Vazia(pilha))
88     {
89         printf("Lista Vazia \n");
90         strcpy(topo.nome, "Inválido");
91         topo.pag = -1;
92         return topo;
93     }
94     aux = (*pilha);
95     topo = aux->dados;
96     return topo;
97 }
98
99
100 void imprimirtopo(CELULA **pilha)
101 {
102     if(Vazia(pilha))
103     {
104         printf("Pilha Vazia\n");
105         return ;
106     }
107
108     printf("Nome: %s", (*pilha)->dados.nome);
109     printf("Paginas: %d", (*pilha)->dados.pag);
110 }
111
112 void imrimir_desempilhando(CELULA**pilha)
113 {
114     if(Vazia(pilha))
115     {
116         printf("Lista Vazia");
117         return;
118     }
119
120     while((*pilha)!=NULL)
121     {
122         imprimirtopo(pilha);
123         pop(pilha);
124     }
125 }
126
127 void imprimir_sem_desempilhar(CELULA ** pilha1, CELULA** pilha2)
128 {
129     inicializar(pilha2);
130     Livro aux;
131     if(Vazia(pilha1))
132     {
133         printf("Pilha Vazia\n");
134         return;
135     }
136     while(!Vazia(pilha1))
137     {
138         aux = pop(pilha1);
139         printf("Nome: %s", aux.nome);
140         printf("Pag: %d", aux.pag);
141         push(pilha2,aux);
142     }
143     while(!Vazia(pilha2))
144     {
145         aux = pop(pilha2);
146         push(pilha1,aux);

```

```
147     }
148
149 }
150
151
152 int main()
153 {
154     return 0;
155 }
156
```