

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct cadastro{
6      char nome[41];
7      int idade;
8      char endereco[31];
9  }Cadastro;
10
11  Cadastro *aloca(int n);
12  Cadastro *realoca(Cadastro *cad, int n);
13  Cadastro coleta();
14  void imprimeDados(Cadastro *cad, int n);
15  void liberaDados(Cadastro *cad);
16
17  int main (){
18      int i, n = 0, lidos = 0, new_qtd;
19      Cadastro *cad, c;
20      FILE *f;
21      f = fopen("arquivo.bin", "rb");
22      if(f == NULL){ //se arq de dados nao existir, então cria-o
23          f = fopen("arquivo.bin", "wb");
24          printf("Informe a quantidade de elementos para Cadastro:\n");
25          scanf("%d", &new_qtd);
26          cad = aloca(new_qtd); //cria bloco com a qtd informada, e segue para coletar dados
27      }else{
28          //se arq dados existir, então abre e lê conteúdo. Para isso deve existir
29          //espaço para pelo menos 1 elemento no vetor dinâmico, então aloca com espaço para 1
30          elemento
31          cad = aloca(1);
32          printf("Arquivo de dados encontrado, efetuando leitura...");
33          while(!feof(f)){ //enquanto nao for o fim...
34              n = fread(&c, sizeof(Cadastro), 1, f);
35              //esse if detecta quando o fread não conseguir ler do arquivo porque é o seu fim.
36              Mesmo depois
37              //de ler a ultima linha, ainda não terá chegado ao fim do arq.
38              //então aqui, impedimos a duplicação da última linha do arquivo, ou estrutura.
39              if(n == 1){
40                  lidos++;
41                  cad = realoca(cad, lidos);
42                  cad[lidos - 1] = c;
43              }
44          }
45          fclose(f);
46
47          printf("\nforam lidos %d registros no arquivo de dados\n", lidos);
48          printf("Entre com a nova quantidade de elementos para os dados: ");
49          scanf("%d", &new_qtd);
50          system("cls");
51
52          //insere novos elementos a partir da última posição lida, se nova qtd não for maior
53          //do que itens lidos no arquivo, esse "for" não é executado "i < new_qtd" <- false!!
54          for(i = lidos; i < new_qtd; i++){
55              cad[i] = coleta();
56          }
57
58          imprimeDados(cad, new_qtd);
59          getchar();
60          //salva todos os dados e encerra programa
61          f = fopen("arquivo.bin", "wb");
62          fwrite(cad, sizeof(Cadastro), new_qtd, f);
63          fclose(f);
64          liberaDados(cad);
65          return 0;
66      }
67
68  Cadastro *aloca(int n){
69      int i;
70      Cadastro *cad = (Cadastro*) malloc(n * sizeof(Cadastro));
71      return cad;
72  }
73
74  Cadastro *realoca(Cadastro *cad, int n){ // realoca 1 posição a mais a cada leitura no arquivo
75      cad = (Cadastro*) realloc(cad, n * sizeof(Cadastro) + sizeof(Cadastro));
76      return cad;
77  }
78
79  Cadastro coleta(){ //coleta dados e devolve 1 estrutura preenchida
80      Cadastro cad;
81      printf("Digite o nome da pessoa:\n");
82      fflush(stdin);
83      fgets(cad.nome, 40, stdin);
84      //substitui o \n no final da string por \0

```

```

83     cad.nome[strlen(cad.nome) - 1] = '\0';
84     printf("Digite a idade da pessoa:\n");
85     scanf("%d", &cad.idade);
86     printf("Digite o endereco da pessoa:\n");
87     fflush(stdin);
88     fgets(cad.endereco, 30, stdin);
89     //substitui o \n no final da string por \0
90     cad.endereco[strlen(cad.endereco) - 1] = '\0';
91     system("cls");
92     return cad;
93 }
94
95 void imprimeDados(Cadastro *cad, int n){
96     int i;
97     for(i = 0; i < n; i++){
98         printf("Cadastro numero %d\n", i + 1);
99         printf("Nome: %s\n", cad[i].nome);
100        printf("Idade : %d\n", cad[i].idade);
101        printf("Endereco: %s\n", cad[i].endereco);
102
103        printf("*****\n\n");
104    }
105 }
106
107 void liberaDados(Cadastro *cad){
108     printf("Liberando memoria...\n\n");
109     free(cad);
110 }
111

```