

```

1: // Programa de Agenda Telefonica
2: // Newton Gomes
3:
4: #include <stdio.h>
5: #include <conio.h>
6: #include <stdlib.h>
7: #include <string.h>
8:
9: #define TAMANHO_AGENDA 100
10:
11: // Estrutura do Contato da Agenda
12:
13: struct Contato
14: {
15:     char nome[50];
16:     char telefone[20];
17:     char email[50];
18: };
19:
20: // Declarações das funções
21:
22: void finalizar();
23: void pausa(char mensagem[]);
24: int ler_opcao();
25: void cadastrar(struct Contato ag[], int *qtd);
26: void consultar(struct Contato ag[], int qtd);
27: void alterar(struct Contato ag[], int qtd);
28: void listar(struct Contato ag[],int qtd);
29:
30: // Programa Principal
31:
32: main ()
33: {
34:     struct Contato agenda[TAMANHO_AGENDA];
35:     int quantidade=0;
36:     int opcao;
37:
38:     while (opcao != 5)
39:     {
40:         system("CLS");
41:         puts("* Agenda Eletronica de Contatos *\n");
42:         printf("%d contatos cadastrados na agenda eletronica\n\n",quantidade);
43:
44:         opcao = ler_opcao();
45:         system("CLS");
46:         switch (opcao)
47:         {
48:             case 1: // cadastrar
49:                 cadastrar(agenda, &quantidade);
50:                 break;
51:
52:             case 2: // consultar
53:                 consultar(agenda, quantidade);
54:                 break;
55:
56:             case 3: // alterar
57:                 alterar(agenda, quantidade);
58:                 break;
59:
60:             case 4: // listar
61:                 listar(agenda, quantidade);
62:                 break;

```

```

63:
64:     }
65: }
66:     finalizar();
67: }
68:
69:
70: // função para finalizar programa
71:
72: void finalizar()
73: {
74:     puts("# Fim do Programa #");
75:     pausa("Tecle Algo...");
76: }
77:
78: // função para mostrar mensagem com pausa
79:
80: void pausa(char mensagem[])
81: {
82:     printf("%s",mensagem);
83:     getch();
84: }
85:
86: // função para ler opção escolhida
87:
88: int ler_opcao()
89: {
90:     int x;
91:     printf("Opcoes:\n");
92:     printf("-----\n");
93:     printf("1-Cadastrar Contato\n");
94:     printf("2-Consultar Contato\n");
95:     printf("3-Alterar Contato\n");
96:     printf("4-Listar Cadastrados\n");
97:     printf("5-Finalizar Programa\n");
98:     printf(">>> "); scanf("%d",&x);
99:     printf("\n");
100:    return x;
101: }
102:
103: // função para cadastrar contato na agenda
104:
105: void cadastrar(struct Contato ag[], int *qtd)
106: {
107:     if (*qtd == TAMANHO_AGENDA)
108:     {
109:         puts("Agenda Cheia!");
110:     }
111:     else
112:     {
113:         printf("Posicao.: %d\n",*qtd+1);
114:         printf("Nome....: "); scanf("%s",ag[*qtd].nome);
115:         printf("Telefone: "); scanf("%s",ag[*qtd].telefone);
116:         printf("Email....: "); scanf("%s",ag[*qtd].email);
117:         (*qtd)++;
118:     }
119:     pausa("\nTecle algo...");
120: }
121:
122: // função para consultar contato na agenda
123:
124: void consultar(struct Contato ag[], int qtd)

```

```

125: {
126:     int pos;
127:     printf("Digite a posicao do contato:\n");
128:     scanf("%d",&pos);
129:     printf("\n");
130:     if (pos < 1 || pos > qtd )
131:         printf("Posicao Invalida ou Inexistente.\n");
132:     else
133:     {
134:         printf("Posicao.: %d\n",pos);
135:         printf("Nome....: %s\n",ag[pos-1].nome);
136:         printf("Telefone: %s\n",ag[pos-1].telefone);
137:         printf("Email....: %s\n",ag[pos-1].email);
138:     }
139:     pausa("\n>>> Pressione algo para continuar...");
140: }
141:
142: // função para alterar contato na agenda
143:
144: void alterar(struct Contato ag[], int qtd)
145: {
146:     int pos;
147:     char str[50];
148:     printf("Digite a posicao do contato:\n");
149:     scanf("%d",&pos);
150:     printf("\n");
151:     if (pos < 1 || pos > qtd )
152:         printf("Posicao Invalida ou Inexistente.\n");
153:     else
154:     {
155:         fflush(stdin); // limpar cache do teclado
156:         printf("Posicao.: %d\n",pos);
157:         printf("Nome....: %s\n>",ag[pos-1].nome);
158:         gets(str);
159:         if (str[0] != '\0')
160:             strcpy(ag[pos-1].nome,str);
161:         printf("Telefone: %s\n>",ag[pos-1].telefone);
162:         gets(str);
163:         if (str[0] != '\0')
164:             strcpy(ag[pos-1].telefone,str);
165:         printf("Email....: %s\n>",ag[pos-1].email);
166:         gets(str);
167:         if (str[0] != '\0')
168:             strcpy(ag[pos-1].email,str);
169:     }
170:     pausa("\n>>> Pressione algo para continuar...");
171: }
172:
173: // função para listar contatos da agenda
174:
175: void listar(struct Contato ag[],int qtd)
176: {
177:     if ( qtd > 0 )
178:     {
179:         for(int i=0; i < qtd; i++)
180:         {
181:             printf("---\nPosicao.: %d\n",i+1);
182:             printf("Nome....: %s\n",ag[i].nome);
183:             printf("Telefone: %s\n",ag[i].telefone);
184:             printf("Email....: %s\n",ag[i].email);
185:         }
186:     }

```

```
187:     else
188:         printf("Agenda Vazia!\n");
189:
190:     pausa("\n>>> Pressione algo para continuar...");
191: }
192:
```