

```

1  #define ex6
2  #include "stdio.h"
3  #include <stdlib.h>
4  #include <conio.h>
5  #include <math.h>
6
7  #ifndef ex1
8  /*
9  1 - Escreva um programa que tem uma estrutura de dados com os membros abaixo.
10     receba dados via teclado e imprima estes conteudos no video no seguinte
11     formato.
12         char, int, long, float, double
13         unsigned char, unsigned int, unsigned long,
14
15         10      20      30      40      50      60      70
16     1234567890123456789012345678901234567890123456789012345678901234567890
17         char      int      long      float      double
18         unsigned char      unsigned int      unsigned long
19 */
20
21     struct data{
22         char c;
23         int i;
24         long l;
25         float f;
26         double d;
27         unsigned char uc;
28         unsigned int ui;
29         unsigned long ul;
30     };
31
32     int main()
33     {
34         struct data dados;
35
36         printf("Digite um char: ");
37         scanf("%c", &dados.c);
38         fflush(stdin);
39         printf("Digite um numero inteiro: ");
40         scanf("%d", &dados.i);
41         fflush(stdin);
42         printf("Digite um numero long: ");
43         scanf("%ld", &dados.l);
44         fflush(stdin);
45         printf("Digite um numero Float: ");
46         scanf("%f", &dados.f);
47         fflush(stdin);
48         printf("Digite um numero Double: ");
49         scanf("%lf", &dados.d);
50         fflush(stdin);
51         printf("Digite um unsigned char: ");
52         scanf("%u", &dados.uc);
53         fflush(stdin);
54         printf("Digite um unsigned int: ");
55         scanf("%u", &dados.ui);
56
57         printf("Digite um unsigned long: ");
58         scanf("%lu", &dados.ul);
59
60         printf("      10      20      30      40      50      60      70\n");
61         printf("123456789012345678901234567890123456789012345678901234567890\n");
62
63         printf("      %c", dados.c);
64         printf("      %-6d", dados.i);
65         printf("      %-10ld", dados.l);
66         printf("      %-11le", dados.f);
67         printf("      %-11le\n", dados.d);
68         printf("      %-3u", dados.uc);
69         printf("      %-6u", dados.ui);
70         printf("      %-11lu", dados.ul);
71     }
72 #endif // ex1
73
74 #ifndef ex2
75 /*
76 2 - Escreva um programa com a estrutura abaixo. Defina um vetor de estruturas
77     de 4 elementos. Receba os dados pelo teclado e imprima-os no video. Faça um
78     menu. (vetor de estruturas)
79         nome, end, cidade, estado, cep
80 */
81
82     struct pessoa{
83         char nome[255];
84         char end[255];

```

```

85     char cidade[255];
86     char estado[3];
87     char cep[10];
88 };
89
90 int main()
91 {
92     int i=0;
93     struct pessoa cliente[4];
94
95     for(i; i < 4; i++){
96         printf("Digite o %d nome: ", i+1);
97         scanf("%[^\\n]s", &cliente[i].nome);
98         fflush(stdin);
99         printf("Digite o %d endereco: ", i+1);
100        scanf("%[^\\n]s", &cliente[i].end);
101        fflush(stdin);
102        printf("Digite o %d cidade: ", i+1);
103        scanf("%[^\\n]s", &cliente[i].cidade);
104        fflush(stdin);
105        printf("Digite o %d estado: ", i+1);
106        scanf("%s", &cliente[i].estado);
107        fflush(stdin);
108        printf("Digite o %d cep: ", i+1);
109        scanf("%s", &cliente[i].cep);
110        fflush(stdin);
111    }
112    printf("MENU ?\\n");
113    printf("_____\\n");
114    for(i=0; i < 4; i++){
115        printf("Nome %d: %s\\n", i+1, cliente[i].nome);
116        printf("Endereco %d: %s\\n", i+1, cliente[i].end);
117        printf("Cidade %d: %s\\n", i+1, cliente[i].cidade);
118        printf("Estado %d: %s\\n", i+1, cliente[i].estado);
119        printf("Cep %d: %s\\n", i+1, cliente[i].cep);
120        printf("_____\\n");
121    }
122 }
123 #endif // ex2
124
125 #ifdef ex3
126
127 /*
128 3 - Crie uma estrutura para representar as coordenadas de um ponto no plano
129      (posicoes X e Y). Em seguida, declare e leia do teclado um ponto e exiba a
130      distancia dele ate' a origem das coordenadas, isto é, posição (0, 0). Para
131      realizar o cálculo, utilize a fórmula a seguir:
132      
$$d = \text{raiz quadrada de } (X_B - X_A)^2 + (Y_B - Y_A)^2$$

133      Em que:
134      d = distância entre os pontos A e B
135      X = coordenada X em um ponto
136      Y = coordenada Y em um ponto
137 */
138 struct coord{
139     int x;
140     int y;
141 };
142 int main()
143 {
144     int i =0;
145
146     struct coord coordenada[2];
147     int distancia=0;
148     for(i=0; i < 2; i++){
149         if(i == 0){
150             printf("Digite a coordenada X do ponta A: ");
151             scanf("%d", &coordenada[i].x);
152             printf("Digite a coordenada Y do ponta A: ");
153             scanf("%d", &coordenada[i].y);
154         }else{
155             printf("Digite a coordenada X do ponta B: ");
156             scanf("%d", &coordenada[i].x);
157             printf("Digite a coordenada Y do ponta B: ");
158             scanf("%d", &coordenada[i].y);
159         }
160     }
161     distancia = sqrt(pow((coordenada[1].x - coordenada[0].x),2)) + pow((coordenada[1].y -
162     coordenada[0].y),2);
163     printf("A distancia entre os pontos e %d", distancia);
164 }
165 #endif // ex3
166

```