

```

1 // F9 -> Compilar          F10 -> Executar          F11 -> Compilar e Executar
2
3 //BIBLIOTECAS
4 #include <stdio.h>          // Funções de Entrada
5 #include <stdlib.h>         // Funções Padrão
6 #include <locale.h>         // Função de texto setlocale()
7
8 // CORPO DO PROGRAMA
9 int main(){ ou void main() { //Inicio
10
11     setlocale(LC_ALL, "Portuguese"); // Permite utilizar caracteres
12     especiais como ex: á, ç, é, dentre outros.
13
14     /*DECLARAÇÃO DE VARIÁVEIS*/
15
16     Sintaxe:
17
18     tipo nome;
19
20     /*|   VARIÁVEIS           |N° DE BITS| FORMATO |
21     | void                   |         |         |
22     | Char                   |      8  |      %c |
23     | Unsigned Char         |      8  |      %c |
24     | Int                    |     32  |      %d |
25     | Unsigned int          |     32  |      %u |
26     | Short int             |     16  |     %hi |
27     | Unsigned short int    |     16  |     %hu |
28     | Long int              |     32  |     %li |
29     | Unsigned long int     |     32  |     %lu |
30     | Float                 |     32  |      %f |
31     | Double                |     64  |     %lf |
32     | Long Double           |     96  |         |
33     */
34
35     //Ex:
36     int x1, y1, result1;      // Declarando as Variáveis globais
37     do tipo Int
38     int x = 1, y = 1, result = 0; // Declaração de Variáveis globais
39     atribuindo Valores iniciais
40     float result2;           // Variável Global do tipo float
41
42     //Obs: Ex: " i-- " -> i = i - 1 (Decrementa) ou " i++ " -> i = i + 1 (Incrementa)
43
44     // ENTRADA OU LEITURA DE DADOS
45     scanf("%d", &x1);         // Ler Variável x1
46     scanf("%d", &y1);         // ler Variável y1
47     scanf("%d %d", &x1, &y1); // ler Variável x1 e y1
48     gets(x1);
49
50     //BLOCOS DE COMANDOS
51     result = x + y;           // OBS: " = " -> Receber ou
52     atribuir " == " -> igual
53     result1 = x1 + y1;
54     float result2 = (result1 / 3);
55
56     // ESTRUTURAS DE DECISÃO " if " ou " if else "
57     if(/*Condição 1*/){      // se
58         // Blocos de Comandos
59     }else if(/*condição 2*/){ // senão se
60         // Blocos de Comandos
61     }else{                   // senão
62         // Blocos de Comandos
63     }
64
65     // if e else sem {}
66     if(/*Condição 1*/)
67         // Um Comando
68     else
69         // Um Comando
70
71     // ESTRUTURAS DE REPETIÇÃO " WHILE "
72
73     while(/*Condição*/) {    // Enquanto a condição for

```

```

70     verdadeira o while vai rodar
71     // Bloco de Comandos
72 }
73 // ESTRUTURAS DE REPETIÇÃO " FOR "
74
75 for (i = 0; i > 0; i++ /*i--*/){ // for (var = inicio; condição;
76     incremento) // Bloco de Comandos
77     break; // Caso necessário
78 }
79
80 //Switch case
81
82 int operador;
83 printf("Qual operdaor deseja escolher?\n");
84 scanf("%d", &operdador); //
85 switch(operador){
86     case 1:
87         // Bloco de Comandos
88         break;
89     case 2:
90         // Bloco de Comandos
91         break;
92     case 2:
93         // Bloco de Comandos
94         break;
95     default: // Caso nenhuma das opções seja
96         escolhida // Bloco de Comandos
97 }
98
99 // SAÍDA OU ESCRITA DE DADOS
100 printf("Calculadora de soma: \n"); // Escrever (\n -> Pular linha)
101 printf("O valor da soma eh: %d ", result1); // Escrever
102 printf("O valor da soma eh: %f ", result2); // Escrever
103 printf("O valor da soma eh: %.2f ", result2); // Escrever
104 printf("cont = %d|soma = %d|ValoraSomar = %d\n", cont, soma, ValoraSomar); //
105 // Passo a passo
106
107 // CRIAR E CHAMAR FUNÇÃO
108 void teste(int b) //tipo_da_funcao NomeDaFuncao (Lista_de_Parametros)
109 {
110     // corpo da função
111 }
112
113 int teste(int b){
114     // Bloco de comandos
115     return(tipo_da_funcao a)
116 }
117
118 int main(){
119     TESTE = teste();
120 }
121
122 //COMANDO DE PARADA DO SISTEMA
123 system("pause");
124
125 //UTILIZAR VARIÁVEL TIPO Char
126
127 char a;
128 printf ( "Entre com um caractere : " ) ;
129 scanf ( "%c" , &a ) ;
130 printf ( "\nCaractere Digitado: %c\n" , a ) ;
131 printf ( "\nCaractere Digitado : %c (Codigo ASCII = %d)\n" , a , a ) ;
132
133 //CRIAR VETOR
134 /* Um vetor é uma seqüência de vários valores do mesmo tipo, armazenados
135 seqüencialmente na memória, e fazendo uso de um mesmo nome de variável
136 para acessar esses valores. Um vetor também pode ser entendido logicamente

```

```
139     como uma lista de elementos de um mesmo tipo. */
140
141     tipo variável[indice];
142
143     int indice = 10;
144     int vetor[indice];
145
146     scanf("%d", &vetor[indice] );
147     printf("%d", vetor[indice]);
148
149     int indice;
150     int vetor[100];
151     ...
152     for (indice = 0; indice < 100; indice++) {
153         // Bloco de comandos
154     }
155
156     printf("%d", vetor[indice]);
157     scanf("%d", &valores[indice] );
158
159 } //FIM
160
```