

```

1  /*****
2  *          *** Arquivo Main ***          *
3  *****/
4
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include "somatorio.h"
8
9  int main(){
10     float num1, num2, res;
11     int x, a = 1;
12     Numeros *p = num_cria();
13     while(1){
14         system("cls");
15         printf("**** Calculadora com TAD ****\n\n\n");
16         printf("Entre com os 2 numeros para calculo: \n");
17         scanf("%f", &num1);
18         scanf("%f", &num2);
19         num_atribui(p, num1, num2);
20
21         printf("\n\n\n");
22         printf("**** Menu de operacoes ****\n\n\n");
23         printf("Digite 1 para soma\n");
24         printf("Digite 2 para subtracao\n");
25         printf("Digite 3 para multiplicacao\n");
26         printf("Digite 4 para divisao\n");
27         printf("Digite 5 para recuperar resultado anterior\n");
28         printf("Digite 0 para sair do programa\n");
29         scanf("%d", &x);
30         system("cls");
31
32         switch(x){
33             case 1:
34                 system("cls");
35                 res = num_soma(p);
36                 printf("O resultado da soma e: %.2f\n\n", res);
37                 system("pause");
38             break;
39             case 2:
40                 system("cls");
41                 res = num_sub(p);
42                 printf("O resultado da subtracao e: %.2f\n\n", res);
43                 system("pause");
44             break;
45             case 3:
46                 system("cls");
47                 res = num_mult(p);
48                 printf("O resultado da multiplicacao e: %.2f\n\n", res);
49                 system("pause");
50             break;
51             case 4:
52                 system("cls");
53                 res = num_div(p);
54                 printf("O resultado da divisao e: %.2f\n\n", res);
55                 system("pause");
56             break;
57             case 5:
58                 system("cls");
59                 res = num_acessa(p);
60                 printf("O resultado anterior e: %.2f\n\n", res);
61                 system("pause");
62             break;
63             case 0:
64                 num_libera(p);
65                 exit(1);
66             break;

```

```
67         default:
68             continue;
69     }
70 }
71 }
72
73
```

```
1  /*****
2  *          ***  Arquivo somatorio.h  ***
3  *****/
4
5  //Arquivo somatorio.h
6  //Atribui novo nome para struct ponto: Ponto
7  typedef struct numeros Numeros;
8
9  //Cria um novo Numero - somente ponteiro!!
10 Numeros *num_cria();
11
12 //Libera um Numero
13 void num_libera(Numeros *n);
14
15 //Acessa valor do ultimo calculo
16 float num_acessa(Numeros *n);
17
18 //Atribui os valores "x" e "y" a uma estrutura Numeros
19 void num_atribui(Numeros *n, float x, float y);
20
21 //executa a soma dos numeros
22 float num_soma(Numeros *p);
23
24 //executa a soma dos numeros
25 float num_sub(Numeros *p);
26
27 //executa a soma dos numeros
28 float num_mult(Numeros *p);
29
30 //executa a soma dos numeros
31 float num_div(Numeros *p);
32
33
34
35
36
37
38
39
```

```

1  /*****
2  *          ***  Arquivo somatorio.c  ***
3  *****/
4
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include <math.h>
8  #include "somatorio.h"//inclui os protótipos
9
10 //definição de tipos de dados
11 struct numeros{
12     float x;
13     float y;
14     float res;
15 };
16
17 Numeros *num_cria(){
18     Numeros *n = (Numeros*) malloc(sizeof(Numeros));
19     n->x = 0;
20     n->y = 0;
21     n->res = 0;
22     return n;
23 }
24
25 //Libera um Numero
26 void num_libera(Numeros *n){
27     free(n);
28 }
29 //Acessa valor do ultimo calculo
30 float num_acessa(Numeros *n){
31     return (n->res);
32 }
33
34 //Atribui os valores "x" e "y" a uma estrutura Numeros
35 void num_atribui(Numeros *n, float x, float y){
36     n->x = x;
37     n->y = y;
38 }
39
40 //executa a soma dos numeros
41 float num_soma(Numeros *n){
42     n->res = n->x + n->y;
43     return n->res;
44 }
45
46 //executa a subtração dos numeros
47 float num_sub(Numeros *n){
48     n->res = n->x - n->y;
49     return n->res;
50 }
51
52 //executa a multiplicação dos numeros
53 float num_mult(Numeros *n){
54     n->res = n->x * n->y;
55     return n->res;
56 }
57
58 //executa a divisão dos numeros
59 float num_div(Numeros *n){
60     n->res = n->x / n->y;
61     return n->res;
62 }

```