

Exercício 1

```
Ex1.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 typedef int Chave;
6 typedef char Valor[22];
7
8 typedef struct map {
9     Chave chave;
10    Valor valor;
11    struct map *prox;
12 } *Map;
13
14 Map no_map(Chave c, Valor v, Map p) {
15     Map n = malloc(sizeof(struct map));
16     n->chave = c;
17     strcpy(n->valor, v);
18     n->prox = p;
19     return n;
20 }
21
22 void insm(Chave c, Valor v, Map *M) {
23     while( *M && c>(*M)->chave )
24         M = &(*M)->prox;
25     if( *M && c==(*M)->chave )
26         strcpy((*M)->valor, v);
27     else *M = no_map(c, v, *M);
28 }
29
30 void exibem(Map M) {
31     printf("[");
32     while( M ) {
33         printf("(%d,%s)", M->chave, M->valor);
34         if( M->prox ) printf(",");
35         M = M->prox;
36     }
37     printf("]");
38 }
```

```
C:\Users\P08012631\Desktop\ X + v
[(15,Ivo),(29,Ana),(36,Leo),(42,Eva)]
[(15,Ivo),(29,Bia),(36,Leo),(42,Eva)]

-----
Process exited after 11.65 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 2

```
Ex1.c Ex2.c
20 }
21 return n;
22
23 void insmr(Chave c, Valor v, Map *M) {
24     if( *M == NULL || c < (*M)->chave ) {
25         *M = no_map(c, v, *M);
26     } else if( c == (*M)->chave ) {
27         strcpy((*M)->valor, v);
28     } else {
29         insmr(c, v, &(*M)->prox);
30     }
31 }
32
33 void exibem(Map M) {
34     printf("[");
35     while( M ) {
36         printf("(%d,%s)", M->chave, M->valor);
37         if( M->prox ) printf(",");
38         M = M->prox;
39     }
40     printf("]\n");
41 }
42
43 int main(void) {
44     Map I = NULL;
45     insmr(36, "Leo", &I);
46     insmr(15, "Ivo", &I);
47     insmr(42, "Eva", &I);
48     insmr(29, "Ana", &I);
49     exibem(I);
50     insmr(29, "Bia", &I);
51     exibem(I);
52     return 0;
53 }
```

```
C:\Users\P08012631\Desktop\ X + v
[(15,Ivo),(29,Ana),(36,Leo),(42,Eva)]
[(15,Ivo),(29,Bia),(36,Leo),(42,Eva)]

-----
Process exited after 12.11 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 3

```
Ex1.c Ex2.c Ex3.c
28 else *M = no_map(c,v,*M);
29
30
31 void remm(Chave c, Map *M) {
32     while( *M && c>(*M)->chave )
33         M = &(*M)->prox;
34     if( *M==NULL || c!=(*M)->chave ) return;
35     Map n = *M;
36     *M = n->prox;
37     free(n);
38 }
39
40 void exibem(Map M) {
41     printf("[");
42     while( M ) {
43         printf("(%d,%s)",M->chave,M->valor);
44         if( M->prox ) printf(",");
45         M = M->prox;
46     }
47     printf("]\n");
48 }
49
50 int main(void) {
51     Map I = NULL;
52     insm(36,"Leo",&I); insm(15,"Ivo",&I);
53     insm(42,"Eva",&I); insm(29,"Ana",&I);
54     exibem(I);
55     remm(29,&I);
56     exibem(I);
57     remm(42,&I);
58     exibem(I);
59     return 0;
60 }
61
62
```

```
C:\Users\P08012631\Desktop\
+ -
[[15,Ivo),(29,Ana),(36,Leo),(42,Eva)]
[[15,Ivo),(36,Leo),(42,Eva)]
[[15,Ivo),(36,Leo)]

-----
Process exited after 1.962 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 4

```
Ex1.c Ex2.c Ex3.c Ex4.c
22 while( *M && c>(*M)->chave )
23     M = &(*M)->prox;
24     if( *M && c==(*M)->chave )
25         strcpy((*M)->valor,v);
26     else *M = no_map(c,v,*M);
27 }
28
29 void remmr(Chave c, Map *M) {
30     if (*M == NULL) {
31         return;
32     }
33     if (c < (*M)->chave) {
34         return;
35     }
36     if (c == (*M)->chave) {
37         Map n = *M;
38         *M = n->prox;
39         free(n);
40         return;
41     }
42     remmr(c, &(*M)->prox);
43 }
44
45 void exibem(Map M) {
46     printf("[");
47     while( M ) {
48         printf("(%d,%s)",M->chave,M->valor);
49         if( M->prox ) printf(",");
50         M = M->prox;
51     }
52     printf("]\n");
53 }
54
55 int main(void) {
56     Map I = NULL;
57     insm(36,"Leo",&I); insm(15,"Ivo",&I);
58     insm(42,"Eva",&I); insm(29,"Ana",&I);
59     exibem(I);
60     remmr(29,&I);
61     exibem(I);
62     remmr(42,&I);
63     exibem(I);
64     return 0;
65 }
```

```
C:\Users\P08012631\Desktop\
+ -
[[15,Ivo),(29,Ana),(36,Leo),(42,Eva)]
[[15,Ivo),(36,Leo),(42,Eva)]
[[15,Ivo),(36,Leo)]

-----
Process exited after 0.7023 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 5

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c
25 strcpy((*M)->valor,v);
26 else *M = no_map(c,v,*M);
27
28
29 int pertm(Chave c, Valor v, Map M) {
30     while( M && c>M->chave )
31         M = M->prox;
32     if( M && c==M->chave )
33         strcpy(v,M->valor);
34     return (M && c==M->chave);
35 }
36
37 void exibem(Map M) {
38     printf("[");
39     while( M ) {
40         printf("(%d,%s)",M->chave,M->valor);
41         if( M->prox ) printf(",");
42         M = M->prox;
43     }
44     printf("]\n");
45 }
46
47 int main(void) {
48     Valor w;
49     Map I = NULL;
50     insm(36,"Leo",&I); insm(15,"Ivo",&I); insm(42,"Eva",&I);
51     exibem(I);
52     if( pertm(42,w,I) ) printf("Valor da chave 42: %s\n",w);
53     else puts("Chave 42 inexistente!");
54     if( pertm(10,w,I) ) printf("Valor da chave 10: %s\n",w);
55     else puts("Chave 10 inexistente!");
56     return 0;
57 }
58
59
60
```

```
C:\Users\P08012631\Desktop\
[[15,Ivo),(36,Leo),(42,Eva)]
Valor da chave 42: Eva
Chave 10 inexistente!

-----
Process exited after 1.738 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 6

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c
26 else *M = no_map(c,v,*M);
27
28
29 int pertmr(Chave c, Valor v, Map M) {
30     if( M == NULL ) {
31         return 0;
32     }
33     if( c < M->chave ) {
34         return 0;
35     }
36     if( c == M->chave ) {
37         strcpy(v, M->valor);
38         return 1;
39     }
40     return pertmr(c, v, M->prox);
41 }
42
43 void exibem(Map M) {
44     printf("[");
45     while( M ) {
46         printf("(%d,%s)",M->chave,M->valor);
47         if( M->prox ) printf(",");
48         M = M->prox;
49     }
50     printf("]\n");
51 }
52
53 int main(void) {
54     Valor w;
55     Map I = NULL;
56     insm(36,"Leo",&I); insm(15,"Ivo",&I); insm(42,"Eva",&I);
57     exibem(I);
58     if( pertmr(42,w,I) ) printf("Valor da chave 42: %s\n",w);
59     else puts("Chave 42 inexistente!");
60     if( pertmr(10,w,I) ) printf("Valor da chave 10: %s\n",w);
61     else puts("Chave 10 inexistente!");
62 }
```

```
C:\Users\P08012631\Desktop\
[[15,Ivo),(36,Leo),(42,Eva)]
Valor da chave 42: Eva
Chave 10 inexistente!

-----
Process exited after 1.624 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 7

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c
41 }
42
43 void destroim(Map *M) {
44     while (*M) {
45         Map temp = *M;
46         *M = (*M)->prox;
47         free(temp);
48     }
49 }
50
51 void exibem(Map M) {
52     printf("[");
53     while( M ) {
54         printf("(%d,%s)", M->chave, M->valor);
55         if( M->prox ) printf(",");
56         M = M->prox;
57     }
58     printf("]\n");
59 }
60
61 int main(void) {
62     Valor w;
63     Map I = NULL;
64     insm(36,"Leo",&I); insm(15,"Ivo",&I); insm(42,"Eva",&I);
65     exibem(I);
66     if( pertmr(42,w,I) ) printf("Valor da chave 42: %s\n",w);
67     else puts("Chave 42 inexistente!");
68     if( pertmr(10,w,I) ) printf("Valor da chave 10: %s\n",w);
69     else puts("Chave 10 inexistente!");
70
71     destroim(&I);
72     printf("Mapeamento depois da destruicao:\n");
73     exibem(I);
74     return 0;
75 }
```

```
C:\Users\P08012631\Desktop\ X + v
[[15,Ivo),(36,Leo),(42,Eva)]
Valor da chave 42: Eva
Chave 10 inexistente!
Mapeamento depois da destruicao:
[]

-----
Process exited after 0.6603 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 8

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c
44 if (*M != NULL) {
45     destroimr(&(*M)->prox);
46     free(*M);
47     *M = NULL;
48 }
49 }
50
51 void exibem(Map M) {
52     printf("[");
53     while( M ) {
54         printf("(%d,%s)", M->chave, M->valor);
55         if( M->prox ) printf(",");
56         M = M->prox;
57     }
58     printf("]\n");
59 }
60
61 int main(void) {
62     Valor w;
63     Map I = NULL;
64     insm(36,"Leo",&I); insm(15,"Ivo",&I); insm(42,"Eva",&I);
65     exibem(I);
66     if( pertmr(42,w,I) ) printf("Valor da chave 42: %s\n",w);
67     else puts("Chave 42 inexistente!");
68     if( pertmr(10,w,I) ) printf("Valor da chave 10: %s\n",w);
69     else puts("Chave 10 inexistente!");
70
71     destroimr(&I);
72     printf("Mapeamento depois da destruicao:\n");
73     exibem(I);
74     return 0;
75 }
76
77
78 }
```

```
C:\Users\P08012631\Desktop\ X + v
[[15,Ivo),(36,Leo),(42,Eva)]
Valor da chave 42: Eva
Chave 10 inexistente!
Mapeamento depois da destruicao:
[]

-----
Process exited after 1.209 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 9

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c [*] exerc 9 - mapeamento[1].c
22 void insm(Chave c, Valor v, Map *M) {
23     while( *M && c->(*M)->chave )
24         M = &(*M)->prox;
25     if( *M && c==(*M)->chave )
26         (*M)->valor=v;
27     else *M = no_map(c,v,*M);
28 }
29
30 void exibem(Map M) {
31     printf("[");
32     while( M ) {
33         printf("(%d,%d)",M->chave,M->valor);
34         if( M->prox ) printf(",");
35         M = M->prox;
36     }
37     printf("]\n");
38 }
39
40
41 int main(void) {
42     Map I = NULL;
43     int tamanho, num_chave;
44     Valor valor;
45
46     printf("Digite o a quantidade de chaves que o mapeamento tera (Somente Numeros): ");
47     scanf("%d", &tamanho);
48
49     for(int i = 1; i <= tamanho; i++){
50
51         printf("Digite a %da. chave:\n", i);
52         scanf("%d", &num_chave);
53         |
54         printf("Digite o valor referente ao %do. valor:\n", i);
55         scanf("%d", &valor);
56         insm(num_chave, valor, &I);
57     }
58 }
59
60
```

```
C:\Users\P08012631\Desktop\ X + v
Digite o a quantidade de chaves que o mapeamento tera (Somente Numeros): 3
Digite a 1a. chave:
5
Digite o valor referente ao 1o. valor:
6
Digite a 2a. chave:
7
Digite o valor referente ao 2o. valor:
8
Digite a 3a. chave:
9
Digite o valor referente ao 3o. valor:
10
[(5,6),(7,8),(9,10)]

-----
Process exited after 10.33 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 10

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c
26     strcpy((*M)->valor, v);
27
28     *M = no_map(c, v, *M);
29 }
30
31 void exibem(Map M) {
32     printf("[");
33     while( M ) {
34         printf("(%s,%s)", M->chave, M->valor);
35         if( M->prox )
36             printf(",");
37         M = M->prox;
38     }
39     printf("]\n");
40 }
41
42 int main(void) {
43     Map I = NULL;
44     int tamanho;
45     Chave chave;
46     Valor valor;
47
48     printf("Digite a quantidade de chaves que o mapeamento tera (Somente Numeros): ");
49     scanf("%d", &tamanho);
50
51     for (int i = 1; i <= tamanho; i++) {
52         printf("Digite a %da. chave:\n", i);
53         scanf("%s", &chave);
54
55         printf("Digite o valor referente ao %do. valor:\n", i);
56         scanf("%s", &valor);
57         insm(chave, valor, &I);
58     }
59 }
60
```

```
C:\Users\P08012631\Desktop\ X + v
Digite a quantidade de chaves que o mapeamento tera (Somente Numeros): 3
Digite a 1a. chave:
aviao
Digite o valor referente ao 1o. valor:
asa
Digite a 2a. chave:
carro
Digite o valor referente ao 2o. valor:
roda
Digite a 3a. chave:
moto
Digite o valor referente ao 3o. valor:
guidao
[(aviao,asa),(carro,roda),(moto,guidao)]

-----
Process exited after 20.08 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```