```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex6.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex12.c Ex13.c Ex14.c Ex15.c Ex16.c Ex17.c Ex18.c Ex19.c Ex20.c Ex21.c Ex22.c Ex23.c
       #include <stdio.h>
#include <stdlib.h>
                                                                 © C:\Users\claud\OneDrive - Fal × + v
       typedef int Item;
                                                                        4
  6 ☐ typedef struct arv {
            struct arv *esq;
            Item item;
            struct arv *dir:
                                                                Process exited after 0.05483 seconds with return value 0 Pressione qualquer tecla para continuar. . .
 10
 11
 12 ☐ Arv arv(Arv e, Item x, Arv d) {
13    Arv n = malloc(sizeof(struct arv));
 14
           n->esq = e;
n->item = x;
 15
 16
            n->dir = d;
           return n;
     L
 18
 20
 21 □ void exibe(Arv A,int n) {
          if( A==NULL ) return;
exibe(A->dir,n+1);
printf("%*s%d\n",3*n, "" ,A->item);
 22
 24
 25
            exibe(A->esq,n+1);
 26
 27
 28 ☐ int main(void) {
          Arv I = arv(arv(NULL,2,NULL),1,arv(NULL exibe(I,0);
 29
 31
          return 0;
```

Exercício 2

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.d
                                                      C:\Users\claud\OneDrive - Fat X
 1 #include <stdio.h>
       #include <stdlib.h>
       typedef int Item;
  4
                                                            5
  6  typedef struct arv {
          struct arv *esq;
Item item;
  7
8
                                                     Process exited after 0.06118 seconds with return value 0 Pressione qualquer tecla para continuar. . .
  9
           struct arv *dir;
    } *Arv;
 10
 11
 n->esq = e;
          n->item = x;
 15
           n->dir = d;
          return n;
    L }
 18
 20
21  void exibe(Arv A,int n) {
22  if( A==NULL ) return;
          exibe(A->dir,n+1);
printf("%*s%d\n",3*n, "" ,A->item);
 23
 25
           exibe(A->esq,n+1);
 26
 28 🗔
        int main(void) {
         Arv I = arv(arv(arv(NULL,4,NULL),2,arv
         exibe(I,0);
 30
 32
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
                                                 C:\Users\claud\OneDrive - Fat X + V
 3 #include <time.h>
                                                    16
    typedef int Item;
 7  typedef struct arv {
                                                 34
         struct arv *esq;
Item item;
                                                    68
10
         struct arv *dir;
                                                       28
   | } *Arv;
11
12
        arv(Arv e, Item x, Arv d) {
Arv n = malloc(sizeof(struct arv));
Process exited after 0.06771 seconds with return value 0
Pressione qualquer tecla para continuar. . .
13 ☐ Arv arv(Arv e, Item x, Arv d) {
14
         n->esq = e;
16
         n->item = x:
17
         n->dir = d;
18
         return n;
19 1
20
23
         return arv(completa(altura - 1), rand
24 L }
25
26 → void exibe(Arv A, int n) {
27    if (A == NULL) return;
        if (A == NULL) return;
exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->item)
28
29
30
         exibe(A->esq, n + 1);
31 L }
32
         srand(time(NULL));
34
         Arv A = completa(3);
exibe(A. 0):
35
36
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7
                                    C:\Users\claud\OneDrive - Fat × + v
     #include <stdio.h>
    #include <stdlib.h>
    #include <time.h>
                                         74
    typedef int Item;
                                         19
                                   76
 7 ☐ typedef struct arv {
 8
         struct arv *esq;
                                         42
         Item item;
                                      47
10
         struct arv *dir;
                                         95
   L } *Arv;
11
n->esq = e;
16
         n->item = x:
         n->dir = d;
18
         return n;
   L,
19
21 ☐ Arv completa(int altura) {
22
        if (altura <= 0) return N
23
         return arv(completa(altura
   L,
24
26  void exibe(Arv A, int n) {
27  if (A == NULL) return;
        exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n,
exibe(A->esq, n + 1);
28
29
30
31
32
```

Exercício 5

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6
                                    ©:\ C:\Users\claud\OneDrive - Fat X
     #include <stdio.h>
1
     #include <stdlib.h>
      #include <time.h>
                                      19
 4
                                          75
 5
     typedef int Item;
                                              45
                                   91
 7 ☐ typedef struct arv {
                                          91
 8
          struct arv *esq;
                                      56
 9
          Item item:
10
          struct arv *dir;
                                              48
11
    L } *Arv;
12
13 ☐ Arv arv(Arv e, Item x,
          arv(Arv e, Item x, A Process exited after 0.05762 seconds with return value 0 Pressione qualquer tecla para continuar. . .
14
15
          n->esq = e;
          n->item = x;
16
17
          n->dir = d;
18
          return n;
   L }
19
20
21 ☐ Arv aleatoria(int n) {
22
          if (n <= 0) return |
23
          return arv(aleatoria
24
25
26 → void exibe(Arv A, int n)
27 if (A == NULL) retur
          exibe(A->dir, n + 1)
printf("%*s%d\n", 3
28
29
30
          exibe(A\rightarrow esq, n+1
   L }
31
32
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex
                                        C:\Users\claud\OneDrive - Fat × +
   #include <stdio.h>
#include <stdlib.h>
 1
                                            20
    #include <time.h>
                                          42
                                             86
    typedef int Item;
                                        47
                                            28
 35
       struct arv *esq;
        Item item;
                                        Nos: 7
        struct arv *dir;
10
  | } *Arv;
11
                                          3
12
5
                                          2
15
16
        n->esq = e;
        n->item = x;
                                       Nos: 5
17
        n->dir = d;
        return n;
18
19 }
                                       Process exited after 0.05957 seconds with return value 0
20
                                       Pressione qualquer tecla para continuar. . . |
return arv(completa(altura - 1), ra
23 24 }
26 - void evibe/Anv A int n) /
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.
                                                                     C:\Users\claud\OneDrive - Fat X
 9
             Item item;
             struct arv *dir;
10
11
    L } *Arv;
12
13 ☐ Arv arv(Arv e, Item x, Arv d) {
14 | Arv n = malloc(sizeof(struct arv));
15
             n->esq = e;
16
             n->item = x:
17
             n->dir = d;
18
             return n;
                                                                    Process exited after 0.04903 seconds with return value 0 Pressione qualquer tecla para continuar. . . \mid
18 |
19 }
20
21
22 □ void exibe(Arv A, int n) {
            if (A == NULL) return;
exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->item
exibe(A->esq, n + 1);
23
24
25
26
27 }
29 = int soma(Arv A){
30
31
             if(A == NULL) return 0;
return A->item + soma(A->esq) + soma(A->dir);
32
```

Exercício 8

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c
                                 C:\Users\claud\OneDrive - Fat X
     #include <stdio.h>
 2
     #include <stdlib.h>
     #include <time.h>
                                       5
 5
     typedef int Item;
 7 ☐ typedef struct arv {
                                 Quantidade de folhas: 3
 8
         struct arv *esq;
 9
         Item item:
         struct arv *dir;
10
                                 Process exited after 0.05287 seconds with return value 0
   | } *Arv;
11
                                 Pressione qualquer tecla para continuar. . .
12
13 ☐ Arv arv(Arv e, Item x, Arv
14
         Arv n = malloc(sizeof(
15
         n->esq = e;
         n->item = x;
16
17
         n->dir = d;
18
         return n;
   L,
19
20
21
22 void exibe(Arv A, int n) {
23
         if (A == NULL) return;
24
         exibe(A->dir, n + 1);
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.
                                            C:\Users\claud\OneDrive - Fat × + ~
     #include <stdio.h>
#include <stdlib.h>
                                                  69
      #include <time.h>
                                              42
                                                  8
     typedef int Item:
                                                      18
                                           76
 7 ☐ typedef struct arv {
                                                  11
          struct arv *esq;
 9
          Item item;
                                                  90
          struct arv *dir;
10
11
                                           A |irvore exibida tem altura igual h|i: 4
12
13 Arv arv(Arv e, Item x, Arv d)
14 Arv n = malloc(sizeof(stru

Process exited after 0.06813 seconds with return value 0

Pressione qualquer tecla para continuar. . . |
15
          n->esq = e;
          n->item = x;
17
          n->dir = d;
          return n;
18
19 }
20
21 ☐ Arv aleatoria(int n) {
          if (n <= 0) return NULL;
22
          return arv(aleatoria(n /
24 L }
25
26 □ void exibe(Arv A, int n) {
```

```
Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c
                                                C:\Users\claud\OneDrive - Fat X
22
           if (n <= 0) return NULL;
23
           return arv(aleatoria(n / 2)
                                                      17
24 }
26 → void exibe(Arv A, int n) {
27    if (A == NULL) return;
                                                      99
                                               Ц9
           exibe(A->dir, n + 1);

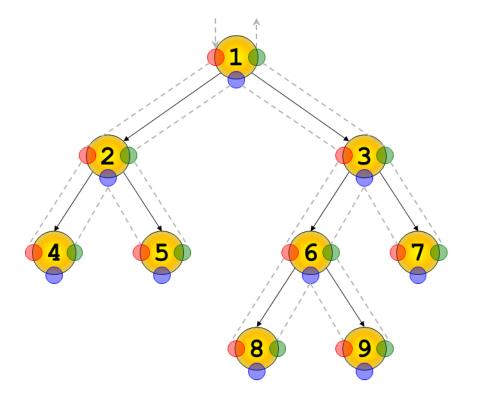
printf("%*s%d\n", 3 * n, "",

exibe(A->esq, n + 1);
28
                                                      85
29
                                                      69
31 }
                                                          60
32
                                               Clone da Arvore exibida:
14
33
                                                      99
35 ☐ Arv clone(Arv A) {
                                                          24
           if (A == NULL) return NULL;
Arv novo = malloc(sizeof(st)
36
37
                                                      85
38
           novo->item = A->item;
novo->esq = clone(A->esq);
                                                  31
39
                                                      69
40
           novo->dir = clone(A->dir);
                                                          60
41
43
                                               Process exited after 0.0503 seconds with return value 0
                                               Pressione qualquer tecla para continuar. . .
45
46 ☐ int main(void) {
           srand(time(NULL));
48
           Arv A= aleatoria(9);
           exibe(A, 0);
```

Exercício 11

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.
19 L }
                                                                       C:\Users\claud\OneDrive - Fat X
21 ☐ Arv aleatoria(int n) {
            if (n <= 0) return NULL:
22
                                                                              84
23 24 }
            return arv(aleatoria(n / 2), rand() % 100,
25
                                                                              83
26  void exibe(Arv A, int n) {
27  if (A == NULL) return;
                                                                              79
            exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->item);
exibe(A->esq, n + 1);
28
                                                                                 41
29
30
                                                                      Digite o item que deseja saber se pertence a Arvore: 83
O item 83 pertence a Arvore.
31
                                                                      Process exited after 4.764 seconds with return value 0 Pressione qualquer tecla para continuar. . .
return pertence(x, A->esq) || pertence(x,
36
37
38
40 ☐ int main(void) {
41    int x;
         srand(time(NULL));
Arv A= aleatoria(9);
exibe(A, 0);
42
43
44
```

Exercício 12



Percursos diretos

Pré-ordem: 1 2 4 5 3 6 8 9 7

Em-ordem: 4 2 5 1 6 3 7 8 9

Pós-ordem: 452689731

Percursos inversos

Pré-ordem: 1 3 2 7 6 9 8 5 4

Em-ordem: 7 2 8 6 1 9 5 4 3

Pós-ordem: 7 8 9 2 5 4 3 6 1

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c
 1
     #include <stdio.h>
                                                             © C:\Users\claud\OneDrive - Fat × + ∨
     #include <stdlib.h>
#include <time.h>
                                                                   30
                                                               75
    typedef int Item:
                                                                      58
                                                            12
 struct arv *esq;
Item item;
                                                                   97
           struct arv *dir:
10
                                                                   36
11 \ } *Arv;
                                                            A sequencia pre-ordem da Arvore exibida eh: 12 6 36 0 97 75 22 58 30
12
Process exited after 0.05434 seconds with return value 0
Pressione qualquer tecla para continuar. . .
15
          n->esq = e;
n->item = x;
16
17
          n->dir = d;
18
19 }
          return n;
21 ☐ Arv aleatoria(int n) {
22
23
          if (n <= 0) return NULL;
return arv(aleatoria(n / 2), rand() % 1</pre>
24 L }
26 □ void exibe(Arv A. int n) {
          if (A == NULL) return;
exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->item);
aviha/A->asa n ± 1).
27
28
29
```

Exercício 14

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c Ex10.c Ex11.c Ex13.c Ex13.c Ex15.c Ex15.c Ex15.c Ex17.c Ex18.c Ex19.c Ex20.c Ex21.c Ex22.c Ex23.c Ex12.c
     #include <stdio.h>
#include <stdlib.h>
 1
                                                    © C:\Users\claud\OneDrive - Fat X + v
     #include <time.h>
     typedef int Item;
                                                      65
                                                          15
                                                              57
                                                   43
 8
          struct arv *esq;
          Item item;
                                                          33
10
          struct arv *dir:
                                                      38
   | } *Arv;
11
12
                                                              90
13 ☐ Arv arv(Arv e, Item x, Arv d) {
                                                   A sequencia em-ordem da arvore exibida eh: 90 3 38 33 43 57 15 65 7
14
          Arv n = malloc(sizeof(struct ar
                                                   Process exited after 0.05281 seconds with return value 0 Pressione qualquer tecla para continuar. . . \mid
15
          n->esq = e;
16
           n->item =
17
          n->dir = d;
   L
19
21 □ Arv aleatoria(int n) {
          if (n <= 0) return NULL;
          return arv(aleatoria(n / 2), ran
23
25
26 □ void exibe(Arv A, int n) {
         if (A == NULL) return;
exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->item);
exiba(A->ee n + 1).
27
28
29
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex
    #include <stdio.h>
1
                                           © C:\Users\claud\OneDrive - Fat × + ∨
     #include <stdlib.h>
     #include <time.h>
                                             85
     typedef int Item;
5
                                                    70
 6
                                          49
 7 ☐ typedef struct arv {
8
         struct arv *esq;
         Item item;
                                             34
                                                69
         struct arv *dir:
10
                                                   82
                                           A sequencia pos-ordem da arvore exibida eh: 82 69 9 34 70 36 23 85 49
12
13 ☐ Arv arv(Arv e, Item x, Arv d) {
                                          Process exited after 0.05077 seconds with return value 0
         Arv n = malloc(sizeof(struct
14
                                          Pressione qualquer tecla para continuar. . .
15
         n->esq = e;
16
         n->item = x;
17
         n->dir = d;
         return n;
19 }
21 ☐ Arv aleatoria(int n) {
22    if (n <= 0) return NULL;
23
24 }
         return arv(aleatoria(n / 2),
26  void exibe(Arv A, int n) {
```

```
Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.d
                                                  C:\Users\claud\OneDrive - Fat X
     #include <stdio.h>
#include <stdlib.h>
      #include <time.h>
                                                        19
     typedef int Item;
                                                            80
                                                 97
 7 

typedef struct arv {
                                                        56
 8
          struct arv *esq;
Item item;
                                                        50
10
          struct arv *dir;
                                                           14
11 } *Arv;
                                                 A sequencia em-ordem inversa da arvore exibida eh: 14 50 92 56 97 80 19 0 42
12
13 ☐ Arv arv(Arv e, Item x, Arv d) {
                                                Process exited after 0.0459 seconds with return value 0 Pressione qualquer tecla para continuar. . .
          Arv n = malloc(sizeof(struct a n->esq = e;
14
15
          n->item = x;
17
          n->dir = d:
18
   L }
19
21 ☐ Arv aleatoria(int n) {
          if (n <= 0) return NULL;</pre>
23
          return arv(aleatoria(n / 2), ra
26 □ void exibe(Arv A. int n) {
```

Exercício 17

```
zi 🗆 Arv aleatoria(int n) 1
           if (n <= 0) return NULL;
           return arv(aleatoria(n / 2), ran
23
24 }
25
                                                  © C:\Users\claud\OneDrive - Fat × + v
                                                        50
26 ☐ void exibe(Arv A, int n) {
          a exibe(Arv A, int n) {
if (A == NULL) return;
exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n, "", A->
exibe(A->esq, n + 1);
                                                     49
27
28
                                                            69
29
30
                                                        66
31
                                                     99
32
33 ☐ Arv poda(Arv *A){
        if("A == NULL) return NULL;
if(("A)->esq == NULL && (*A)->dir
Arv temp = *A;
34
35
36
                                                     49
37
           *A = NULL;
                                                        13
38
           free(temp):
                                                 98
39
           return NULL;
                                                     99
40
                                                        54
41
        (*A)->esq = poda(&(*A)->esq);
42
43
        (*A)->dir = poda(&(*A)->dir);
        return *A;
Process exited after 0.0501 seconds with return value 0 Pressione qualquer tecla para continuar. . .
48
           srand(time(NULL));
49
           Arv A= aleatoria(9);
           printf("A arvore exibida sem suas folhas eh: \n\n");
51
```

```
21 ☐ Arv aleatoria(int n) {
22    if (n <= 0) return NULL;
        23
25
                                                     0
89
                                                        60
                                                     0
31 }
33 F int destroi(Arv *A) {
                                                  A arvore foi destruida com sucesso!
         if (*A == NULL) return 0;
int n = destroi(&(*A)->esq);
35
36
         int m = destroi(&(*A)->dir);
free(*A);
                                                  Process exited after 0.04939 seconds with return value 0 Pressione qualquer tecla para continuar. . . |
37
         *A = NULL:
38
         return n + m + 1;
40 1
41
42 ☐ int main(void) {
43
         int x:
         srand(time(NULL));
45
         Arv A= aleatoria(9):
         exibe(A, 0);
destroi(&A);
         printf("A arvore foi destruida com sucesso!\n");
48
```

```
23 24 }
            return arv(aleatoria(n / 2), rand() % 100, aleatoria(n - n / 2 - 1));
25
                                                          C:\Users\claud\OneDrive - Fat X + ~
26 → void exibe(Arv A, int n) {
27   if (A == NULL) return;
                                                            15
           exibe(A->dir, n + 1);
printf("%*s%d\n", 3 * n,
exibe(A->esq, n + 1);
28
                                                                 50
29
                                                                     37
    L 3
                                                         39
31
                                                                34
32
33
                                                            82
                                                                47
34 ☐ int conta(Item x, Arv A) {
                                                                    50
35
            if (A == NULL)
                                                        Qual elemento gostaria de saber se esta na arvore? 34
O elemento 34 aparece 1 vezes na arvore.
36
                return 0:
37
            else if (A->item == x)
38
                return 1 + conta(x, A->esq) +
                Process exited after 4.185 seconds with return value 0 return conta(x, A->esq) + cont Pressione qualquer tecla para continuar. . .
39
40
    L
41
42
43
44 ☐ int main(void) {
         int x;
Arv *B;
45
46
47
            srand(time(NULL));
48
            Arv A= aleatoria(9):
49
            exibe(A, 0);
           printf("Qual elemento gostaria de saber se esta na arvore? ");
scanf("%d", &x);
50
51
            conta(x,A);
52
```

Exercício 20

```
princi( ""5%a\n , ) "
exibe(A->esq, n + 1);
                                     ⊤ n, , A->1tem/;
30
  L 3
                                                                      C:\Users\claud\OneDrive - Fat × + ~
33 ☐ int iguais(Arv A, Arv B) {
                                                                             80
34
35
36
                                                                         ДΘ
           if (A == NULL && B == NULL)
          return 1;
else if (A == NULL || B == NULL)
return 0;
37
38
                                                                     63
           else if (A->item != B->item)
                                                                         78
39
10
11
                                                                             46
           else
12
                return iguais(A->esq, B->esq) && iguais
   L 3
                                                                         22
45
                                                                             90
18
        srand(time(NULL));
        Arv A= aleatoria(9);
Arv B= aleatoria(9);
19
                                                                             92
        exibe(A, 0);
printf("\n\n");
51
52
53
54
        if(!iguais(B,A)){
        printf("\n\nAs arvores sao diferentes");
} else{
55
56
57
                                                                     As arvores sao diferentes
                                                                     Process exited after 0.05396 seconds with return value 0 Pressione qualquer tecla para continuar. . . |
          printf("\n\nAs arvores sao iguais");
58
59
50
        return 0;
```

```
20
                                                             © C:\Users\claud\OneDrive - Fat × + v
21 ☐ Arv aleatoria(int n) {
            if (n <= 0) return NULL;
                                                             Arvore aleatoria:
22
23
24
            return arv(aleatoria(n / 2), rand
    L ,
                                                                66
25
26 □ void exibe(Arv A, int n) {
            a exibe(Arv A, int n) {
   if (A == NULL) return;
   exibe(A->dir, n + 1);
   printf("%*s%d\n", 3 * n, "", A->i*
   exibe(A->esq, n + 1);
                                                                     56
28
                                                                     90
30
    L }
                                                             Arvore espelhada:
32
33 ☐ Arv espelho(Arv A){
                                                                     90
34
         if(A == NULL) return NULL;
         return arv(espelho(A->dir), A->item
                                                                 66
                                                                     56
36
37
    L }
                                                             67
38
                                                                         6
40
                                                                66
41 ☐ int main(void) {
         srand(time(NULL));
printf("Arvore aleatoria:\n");
42
            Arv A= aleatoria(9);
exibe(A, 0);
                                                            Process exited after 0.0986 seconds with return value 0 Pressione qualquer tecla para continuar. . . \mid
11
45
         printf("Arvore espelhada:\n");
Arv B = espelho(A);
46
48
            exibe(B, 0);
49
            return 0;
```

```
EXI.C EXS.C 
  29
30
31 }
  |ürvore A:
                                                                                                                                                                                                                                                                                  42
  36
37 }
 49
                                                                                                                                                                                                                                                                                                              59
                                                                                                                                                                                                                                                                    lürvore B
  43 44 }
   45
                                                                                                                                                                                                                                                                                 49
   46
                                                                                                                                                                                                                                                                                                56
 46
47 int main(void) {
48     srand(time(NULL));
49     printf("Ã rvore A:\n");
50     Arv A= aleatoria(9);
51     cvibe(A 0);
  49
50
51
                                                                                                                                                                                                                                                                                                58
                                exibe(A, 0);

printf("Ã rvore B:\n");

Arv B = espelho(A);

exibe(B, 0);
   52
                                                                                                                                                                                                                                                                       As Arvores sao espelhos.
  53
54
   55
                                                                                                                                                                                                                                                                     Process exited after 0.04974 seconds with return value 0 Pressione qualquer tecla para continuar. . . |
  56 E
                                 if(espelho_ou_nao(A, B)){
  printf("As Arvores sao espelhos.\n");
   58
                                 }else{
  59
60
                                          printf("As Arrvores nao sao espelhos.\n");
```

```
29 printf("%*s%d\n", 3 * n, "", A->item);
exib(A->esq, n + 1);

30 c(A)Users\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\claud\c
30
31 }
                                                                                                                                                                                                               © C:\Users\claud\OneDrive - Fat × + ~
                                                                                                                                                                                                             Arvore aleatoria:
70
 32
69
36
37
                                                                                                                                                                                                                                              50
                           return r;
                                                                                                                                                                                                             53
38 }
                                                                                                                                                                                                                      93
40 □ void preenche_vetor(Arv A, Item *v, int *i) {
                                  if (A == NULL) return;
preenche_vetor(A->esq, v, i);
41
42
                                                                                                                                                                                                                                              52
                                                                                                                                                                                                             Arvore balanceada:
43
44
45 }
                                    v[(*i)++] = A->item;
                                  preenche_vetor(A->dir, v, i);
                                                                                                                                                                                                                                   69
46
47
48 □ int main(void) {
                                                                                                                                                                                                                                   50
                                                                                                                                                                                                            53
                                  srand(time(NULL));
printf("Arvore aleatoria:\n");
Arv A = aleatoria(9);
49
50
                                                                                                                                                                                                                                   93
                                                                                                                                                                                                                       38
 51
52
53
54
55
56
                                    exibe(A, 0);
                                                                                                                                                                                                           Process exited after 0.05241 seconds with return value 0 Pressione qualquer tecla para continuar. . .
                                    Item v[9];
                                    int i = 0;
57
58
59
                                    preenche_vetor(A, v, &i);
                                    printf("Arvore balanceada:\n");
60
61
                                   Arv B = balanceada(v, 0, 8);
exibe(B, 0);
                                    return 0;
```