Exercício 1

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.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

```c
#include <stdio.h>
#include <stdlib.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq  = e;
    n->item = x;
    n->dir  = d;
    return n;
}


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);
}

 int main(void) {
  Arv I = arv(arv(NULL,2,NULL),1,arv(NULL
  exibe(I,0);
  return 0;
  }
```

C:\Users\claud\OneDrive - Fat

```
      4
   3
1
   2

--------------------------------
Process exited after 0.05483 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 2

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c Ex9.c

```c
#include <stdio.h>
 #include <stdlib.h>

 typedef int Item;

 typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
 } *Arv;

 Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq  = e;
    n->item = x;
    n->dir  = d;
    return n;
 }


 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);
 }

  int main(void) {
   Arv I = arv(arv(arv(NULL,4,NULL),2,arv
   exibe(I,0);
   return 0;
   }
```

C:\Users\claud\OneDrive - Fat

```
   3
1
      5
   2
      4

--------------------------------
Process exited after 0.06118 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 3

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.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

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv completa(int altura) {
    if (altura <= 0) return NULL;
    return arv(completa(altura - 1), rand
}

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);
}

int main(void) {
    srand(time(NULL));
    Arv A = completa(3);
    exibe(A, 0);
```

C:\Users\claud\OneDrive - Fat

```
      53
   16
      13
34
      4
   68
      28

--------------------------------
Process exited after 0.06771 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 4

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d)
    Arv n = malloc(sizeof(stru
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv completa(int altura) {
    if (altura <= 0) return NU
    return arv(completa(altura
}

void exibe(Arv A, int n) {
    if (A == NULL) return;
    exibe(A->dir, n + 1);
    printf("%*s%d\n", 3 * n,
    exibe(A->esq, n + 1);
}
```

C:\Users\claud\OneDrive - Fat

```
        34
      74
    2
      19
76
        98
      42
    47
      95

--------------------------------
Process exited after 0.05448 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 5

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, A
    Arv n = malloc(sizeo
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return N
    return arv(aleatoria
}

void exibe(Arv A, int n)
    if (A == NULL) retur
    exibe(A->dir, n + 1)
    printf("%*s%d\n", 3
    exibe(A->esq, n + 1)
}
```

C:\Users\claud\OneDrive - Fat

```
      96
    19
      75
        45
91
      91
    56
      61
        48

--------------------------------
Process exited after 0.05762 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 6

Ex1.c Ex2.c Ex3.c Ex4.c Ex5.c Ex6.c Ex7.c Ex8.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv completa(int altura) {
    if (altura <= 0) return NULL;
    return arv(completa(altura - 1), ra
}
```

C:\Users\claud\OneDrive - Fat

```
      20
    42
      86
47
      28
    35
      57
Nos: 7

    3
1
      5
    2
      4
Nos: 5

--------------------------------
Process exited after 0.05957 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 7

```c
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}


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);
}

int soma(Arv A){
    if(A == NULL) return 0;
    return A->item + soma(A->esq) + soma(A->dir);
```

```
   3
1
      5
   2
      4
Soma: 15

--------------------------------
Process exited after 0.04903 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 8

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv
    Arv n = malloc(sizeof(
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}


void exibe(Arv A, int n) {
    if (A == NULL) return;
    exibe(A->dir, n + 1);
```

```
   3
1
      5
   2
      4
Quantidade de folhas: 3

--------------------------------
Process exited after 0.05287 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 9

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d)
    Arv n = malloc(sizeof(stru
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2
}

void exibe(Arv A, int n) {
```

```
      69
   42
      8
         18
76
      11
   9
      90
         64
A árvore exibida tem altura igual há: 4
--------------------------------
Process exited after 0.06813 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 10

```c
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2),
}

void exibe(Arv A, int n) {
    if (A == NULL) return;
    exibe(A->dir, n + 1);
    printf("%*s%d\n", 3 * n, "",
    exibe(A->esq, n + 1);
}



Arv clone(Arv A) {
    if (A == NULL) return NULL;
    Arv novo = malloc(sizeof(st
    novo->item = A->item;
    novo->esq = clone(A->esq);
    novo->dir = clone(A->dir);

    return novo;
}


int main(void) {
    srand(time(NULL));
    Arv A= aleatoria(9);
    exibe(A, 0);
```

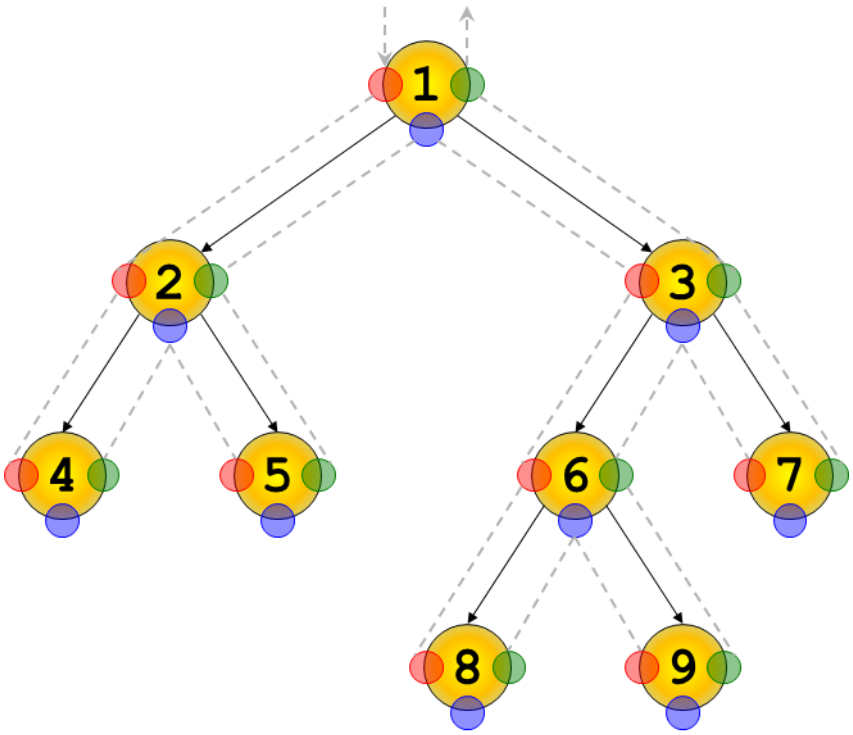
```
C:\Users\claud\OneDrive - Fal

      17
   14
      99
         24
49
      85
   31
      69
         60
Clone da Arvore exibida:      17
   14
      99
         24
49
      85
   31
      69
         60

--------------------------------
Process exited after 0.0503 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 11

```c
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), rand() % 100,
}

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);
}

int pertence(Item x, Arv A){
    if(A == NULL) return 0;
    if(x == A->item) return 1;
    return pertence(x, A->esq) || pertence(x,

}

int main(void) {
  int x;
  srand(time(NULL));
  Arv A= aleatoria(9);
  exibe(A, 0);
```

```
C:\Users\claud\OneDrive - Fal

      9
   16
      84
         38
82
      83
   10
      79
         41
Digite o item que deseja saber se pertence a Arvore: 83
O item 83 pertence a Arvore.
--------------------------------
Process exited after 4.764 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

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: 4 5 2 6 8 9 7 3 1

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

Exercício 13

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv));
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), rand() % 1
}

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);
```

```
C:\Users\claud\OneDrive - Fat

      30
   75
      22
         58
12
      97
   6
      36
         0
A sequencia pre-ordem da Arvore exibida eh: 12 6 36 0 97 75 22 58 30
--------------------------------
Process exited after 0.05434 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 14

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct arv
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), ran
}

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);
```

```
C:\Users\claud\OneDrive - Fat

      7
   65
      15
         57
43
      33
   38
      3
         90
A sequencia em-ordem da arvore exibida eh: 90 3 38 33 43 57 15 65 7
--------------------------------
Process exited after 0.05281 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 15

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct a
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), r
}

void exibe(Arv A, int n) {
```

```
C:\Users\claud\OneDrive - Fat

      23
   85
      36
         70
49
      9
   34
      69
         82
A sequencia pos-ordem da arvore exibida eh: 82 69 9 34 70 36 23 85 49
--------------------------------
Process exited after 0.05077 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 16

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

typedef int Item;

typedef struct arv {
    struct arv *esq;
    Item item;
    struct arv *dir;
} *Arv;

Arv arv(Arv e, Item x, Arv d) {
    Arv n = malloc(sizeof(struct ar
    n->esq = e;
    n->item = x;
    n->dir = d;
    return n;
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), ra
}

void exibe(Arv A, int n) {
```

```
C:\Users\claud\OneDrive - Fat

    42
  0
    19
      80
97
    56
  92
    50
      14
A sequencia em-ordem inversa da arvore exibida eh: 14 50 92 56 97 80 19 0 42
--------------------------------
Process exited after 0.0459 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 17

```c
Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), ran
}

void 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);
}

Arv poda(Arv *A){
  if(*A == NULL) return NULL;
  if((*A)->esq == NULL && (*A)->dir
    Arv temp = *A;
    *A = NULL;
    free(temp);
    return NULL;
  }
  (*A)->esq = poda(&(*A)->esq);
  (*A)->dir = poda(&(*A)->dir);
  return *A;
}

int main(void) {
  Arv *B;
    srand(time(NULL));
    Arv A= aleatoria(9);
    exibe(A, 0);
    printf("A arvore exibida sem suas folhas eh: \n\n");
```

```
C:\Users\claud\OneDrive - Fat

    50
  49
    13
      69
98
    66
  99
    54
      41
A arvore exibida sem suas folhas eh:

  49
    13
98
  99
    54

--------------------------------
Process exited after 0.0501 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 18

```c
Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), rand() % 10
}

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);
}

int destroi(Arv *A) {
    if (*A == NULL) return 0;
    int n = destroi(&(*A)->esq);
    int m = destroi(&(*A)->dir);
    free(*A);
    *A = NULL;
    return n + m + 1;
}

int main(void) {
    int x;
    srand(time(NULL));
    Arv A= aleatoria(9);
    exibe(A, 0);
    destroi(&A);
    printf("A arvore foi destruida com sucesso!\n");
```

```
C:\Users\claud\OneDrive - Fat

    62
  0
    4
      89
17
    60
  0
    21
      50
A arvore foi destruida com sucesso!

--------------------------------
Process exited after 0.04939 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 19

```c
    return arv(aleatoria(n / 2), rand() % 100, aleatoria(n - n / 2 - 1));
}

void exibe(Arv A, int n) {
    if (A == NULL) return;
    exibe(A->dir, n + 1);
    printf("%*s%d\n", 3 * n, "", A->it
    exibe(A->esq, n + 1);
}


int conta(Item x, Arv A) {
    if (A == NULL)
        return 0;
    else if (A->item == x)
        return 1 + conta(x, A->esq) +
    else
        return conta(x, A->esq) + cont
}


int main(void) {
    int x;
   Arv *B;
    srand(time(NULL));
    Arv A= aleatoria(9);
    exibe(A, 0);
    printf("Qual elemento gostaria de saber se esta na arvore? ");
    scanf("%d", &x);
    conta(x,A);
    printf("O elemento %d aparece %d vezes na arvore ", x, conta(x,A));
```

```
C:\Users\claud\OneDrive - Fal
      52
   15
      50
         37
39
      34
   82
      47
         50
Qual elemento gostaria de saber se esta na arvore? 34
O elemento 34 aparece 1 vezes na arvore.
--------------------------------
Process exited after 4.185 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 20

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

int iguais(Arv A, Arv B) {

    if (A == NULL && B == NULL)
        return 1;
    else if (A == NULL || B == NULL)
        return 0;
    else if (A->item != B->item)
        return 0;
    else
        return iguais(A->esq, B->esq) && iguais
}



int main(void) {
   srand(time(NULL));
   Arv A= aleatoria(9);
   Arv B= aleatoria(9);
   exibe(A, 0);
   printf("\n\n");
   exibe(B, 0);
   if(!iguais(B,A)){
     printf("\n\nAs arvores sao diferentes");
   } else{
     printf("\n\nAs arvores sao iguais");
   }

   return 0;
}
```

```
C:\Users\claud\OneDrive - Fal
      80
   40
      58
         1
63
      61
   78
      46
         82


      95
   22
      90
         99
95
      92
   5
      9
         66


As arvores sao diferentes
--------------------------------
Process exited after 0.05396 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 21

```c
}

Arv aleatoria(int n) {
    if (n <= 0) return NULL;
    return arv(aleatoria(n / 2), rand
}

void 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);
}

Arv espelho(Arv A){
  if(A == NULL) return NULL;
  return arv(espelho(A->dir), A->item

}



int main(void) {
    srand(time(NULL));
  printf("Arvore aleatoria:\n");
    Arv A= aleatoria(9);
    exibe(A, 0);
  printf("Arvore espelhada:\n");
    Arv B = espelho(A);
    exibe(B, 0);
    return 0;
}
```

```
C:\Users\claud\OneDrive - Fal
Arvore aleatoria:
      27
   66
      14
         6
67
      56
   66
      90
         39
Arvore espelhada:
         39
      90
   66
      56
67
         6
      14
   66
      27


--------------------------------
Process exited after 0.0986 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 22

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

Arv espelho(Arv A){
  if(A == NULL) return NULL;
  return arv(espelho(A->dir), A->item, espelho(A-

}

int espelho_ou_nao(Arv A, Arv B){
  if(A == NULL && B == NULL) return 1;
  if(A == NULL || B == NULL) return 0;
  if(A->item != B->item) return 0;
  return espelho_ou_nao(A->esq, B->dir) && espel
}


int main(void) {
    srand(time(NULL));
  printf("Ã rvore A:\n");
    Arv A= aleatoria(9);
    exibe(A, 0);
  printf("Ã rvore B:\n");
    Arv B = espelho(A);
    exibe(B, 0);

  if(espelho_ou_nao(A, B)){
    printf("As Arvores sao espelhos.\n");
  }else{
    printf("As Arrvores nao sao espelhos.\n");
  }

    return 0;
}
```

```
C:\Users\claud\OneDrive - Fat
Ã rvore A:
        53
    42
        58
            7
76
        56
    49
        16
            59
Ã rvore B:
            59
        16
    49
        56
76
            7
        58
    42
        53
As Arvores sao espelhos.

--------------------------------
Process exited after 0.04974 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 23

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

Arv balanceada(Item *v, int p, int u){
  if(p>u) return NULL;
  int q = (p+u)/2;
  Arv r = arv(balanceada(v,p,q-1), v[q], balar
  return r;
}

void preenche_vetor(Arv A, Item *v, int *i) {
    if (A == NULL) return;
    preenche_vetor(A->esq, v, i);
    v[(*i)++] = A->item;
    preenche_vetor(A->dir, v, i);
}


int main(void) {
    srand(time(NULL));
    printf("Arvore aleatoria:\n");
    Arv A = aleatoria(9);
    exibe(A, 0);


    Item v[9];
    int i = 0;
    preenche_vetor(A, v, &i);

    printf("Arvore balanceada:\n");
    Arv B = balanceada(v, 0, 8);
    exibe(B, 0);
    return 0;
}
```

```
C:\Users\claud\OneDrive - Fat
Arvore aleatoria:
        70
    69
        8
            50
53
        47
    93
        38
            52
Arvore balanceada:
        70
        69
    8
        50
53
            47
        93
    38
        52

--------------------------------
Process exited after 0.05241 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```