

Exercício 1

Ex1.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void troca(int v[], int i, int j) {
5     int x = v[i];
6     v[i] = v[j];
7     v[j] = x;
8 }
9
10 void bsort(int v[], int n) {
11     for(int i=1; i<n; i++)
12         for(int j=0; j<n-i; j++)
13             if(v[j]>v[j+1])
14                 troca(v,j,j+1);
15 }
16
17 void exibe(int v[], int n){
18     for(int i=0; i<n; i++){
19         printf("%d ", v[i]);
20     }
21 }
22
23 int main(void){
24     int v[10] = {83,31,91,46,27,20,96,25,96,80};
25     bsort(v,10);
26     exibe(v,10);
27     return 0;
28 }
```

```
C:\Users\P08012631\Desktop\ x + v
20 25 27 31 46 80 83 91 96 96
-----
Process exited after 0.03084 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 2

Ex2.c

```
1 #include <stdio.h>
2
3 void empurra(int v[], int PosicaoDestino){
4     int NumeroMaior, IndiceMaior;
5     for(int i = 0; i<PosicaoDestino; i++){
6         if(v[i]>NumeroMaior){
7             IndiceMaior = i;
8             NumeroMaior = v[i];
9         }
10    }
11
12    for(int i = IndiceMaior; i<PosicaoDestino; i++){
13        v[i] = v[i+1];
14    }
15    v[PosicaoDestino] = NumeroMaior;
16 }
17
18 void troca(int v[], int i, int j) {
19     int x = v[i];
20     v[i] = v[j];
21     v[j] = x;
22 }
23
24 void bsort(int v[], int n) {
25     for(int i=1; i<n; i++)
26         for(int j=0; j<n-i; j++)
27             if(v[j]>v[j+1])
28                 troca(v,j,j+1);
29 }
30
31 void exibe(int v[], int n){
32     for(int i=0; i<n; i++){
33         printf("%d ", v[i]);
34     }
35 }
36
37 int main(void) {
38     int v[9] = {51,82,38,99,75,19,69,46,27};
39     empurra(v,8);
40     exibe(v,9);
41     return 0;
42 }
43
44
```

```
C:\Users\cloud\OneDrive - Fat x + v
51 82 38 75 19 69 46 27 99
-----
Process exited after 0.07521 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Exercício 3

```
[*] Ex2.c  Ex1.c  Ex3.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void troca(int v[], int i, int j) {
5      int temp = v[i];
6      v[i] = v[j];
7      v[j] = temp;
8  }
9
10 void empurra(int v[], int u) {
11     if (u <= 0) return;
12
13     int max_idx = 0;
14     for (int i = 1; i <= u; i++) {
15         if (v[i] > v[max_idx]) {
16             max_idx = i;
17         }
18     }
19     troca(v, max_idx, u);
20     empurra(v, u - 1);
21 }
22
23 void bsr(int v[], int n) {
24     if (n <= 1) return;
25
26     empurra(v, n - 1);
27     bsr(v, n - 1);
28 }
29
30 void exibe(int v[], int n) {
31     printf("{");
32     for (int i = 0; i < n; i++) {
33         printf("%d", v[i]);
34         if (i < n - 1) printf(",");
35     }
36     printf("}\n");
37 }
38
39 int main(void) {
40     int v[9] = {51, 82, 38, 99, 75, 19, 69, 46, 27};
41     bsr(v, 9);
42     exibe(v, 9);
43     return 0;
44 }
```

```
C:\Users\cloud\OneDrive - Fat  X  +  v
{19,27,38,46,51,69,75,82,99}

-----
Process exited after 0.05909 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 4

```
Ex4.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void troca(int v[], int i, int j) {
5      int temp = v[i];
6      v[i] = v[j];
7      v[j] = temp;
8  }
9
10 void intercala(int v[], int p, int m, int u) {
11     int *w = malloc((u-p+1)*sizeof(int));
12     int i=p, j=m+1, k=0;
13     while( i<=m && j<=u )
14         w[k++] = (v[i]<v[j]) ? v[i++] : v[j++];
15     while( i<=m ) w[k++] = v[i++];
16     while( j<=u ) w[k++] = v[j++];
17     for(k=0; k<=u-p; k++) v[p+k] = w[k];
18     free(w);
19 }
20
21 void exibe(int v[], int n) {
22     printf("{");
23     for (int i = 0; i < n; i++) {
24         printf("%d", v[i]);
25         if (i < n - 1) printf(",");
26     }
27     printf("}\n");
28 }
29
30 int main(void) {
31     int v[8] = {31,48,60,80,19,27,52,75};
32     intercala(v,0,3,7);
33     exibe(v,8);
34     int w[9] = {10,82,27,38,41,53,60,75,99};
35     intercala(w,0,1,8);
36     exibe(w,9);
37     return 0;
38 }
```

```
C:\Users\cloud\OneDrive - Fat  X  +  v
{19,27,31,48,52,60,75,80}
{10,27,38,41,53,60,75,82,99}

-----
Process exited after 0.08877 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 5

```
Ex4.c Ex5.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void troca(int v[], int i, int j) {
5      int temp = v[i];
6      v[i] = v[j];
7      v[j] = temp;
8  }
9
10 void intercala(int v[], int p, int m, int u) {
11     int *w = malloc((u-p+1)*sizeof(int));
12     int i=p, j=m+1, k=0;
13     while( i<=m && j<=u )
14         w[k++] = (v[i]<v[j]) ? v[i++] : v[j++];
15     while( i<=m ) w[k++] = v[i++];
16     while( j<=u ) w[k++] = v[j++];
17     for(k=0; k<=u-p; k++) v[p+k] = w[k];
18     free(w);
19 }
20
21 void ms(int v[], int p, int u) {
22     if( p==u ) return;
23     int m = (p+u)/2;
24     ms(v,p,m);
25     ms(v,m+1,u);
26     intercala(v,p,m,u);
27 }
28
29 void msort(int v[], int n) {
30     ms(v,0,n-1);
31 }
32
33 void exibe(int v[], int n) {
34     printf("{");
35     for (int i = 0; i < n; i++) {
36         printf("%d", v[i]);
37         if (i < n - 1) printf(",");
38     }
39 }
```

```
C:\Users\cloud\OneDrive - Fat  X + v
{20,25,27,31,46,80,83,91,96,96}

Process exited after 0.07379 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 6

```
Ex4.c Ex5.c Ex6.c
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void troca(int v[], int i, int j) {
5      int temp = v[i];
6      v[i] = v[j];
7      v[j] = temp;
8  }
9
10 void intercala(int v[], int p, int m, int u) {
11     int *w = malloc((u - p + 1) * sizeof(int));
12     int i = p, j = m + 1, k = 0;
13     while (i <= m && j <= u)
14         w[k++] = (v[i] < v[j]) ? v[i++] : v[j++];
15     while (i <= m)
16         w[k++] = v[i++];
17     while (j <= u)
18         w[k++] = v[j++];
19     for (k = 0; k <= u - p; k++)
20         v[p + k] = w[k];
21     free(w);
22 }
23
24 void ms(int v[], int p, int u) {
25     if (p == u)
26         return;
27     int m = (p + u) / 2;
28     ms(v, p, m);
29     ms(v, m + 1, u);
30     intercala(v, p, m, u);
31 }
32
33 void msort(int v[], int n) {
34     ms(v, 0, n - 1);
35 }
36
37 void exibe(int v[], int n) {
38     printf("{");
39 }
```

```
C:\Users\cloud\OneDrive - Fat  X + v
Vetor preenchido aleatoriamente:
{175,400,869,56,83,879,16,644,809,769}
Vetor ordenado:
{16,56,83,175,400,644,769,809,869,879}

Process exited after 0.062 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 7

```
Ex4.c Ex5.c Ex6.c Ex7.c
37     for(k=0; k<=u-p; k++) v[p+k] = w[k];
38     free(w);
39 }
40
41 void exibe(int v[], int n) {
42     printf("{");
43     for (int i = 0; i < n; i++) {
44         printf("%d", v[i]);
45         if (i < n - 1) printf(",");
46     }
47     printf("}\n");
48 }
49
50 void preenche(int v[], int n, int s) {
51     srand(s);
52     for (int i = 0; i < n; i++)
53         v[i] = rand() % 1000;
54 }
55
56 int main(void) {
57     int v[100000];
58     double t, b, m;
59     puts(" n bsort msort");
60     for(int n=10000; n<=100000; n+=10000) {
61         preenche(v,n,1);
62         t = clock(); // definida em time.h
63         bsort(v,n);
64         b = (clock()-t)/CLOCKS_PER_SEC; // tempo do bsort
65         preenche(v,n,1);
66         t = clock();
67         msort(v,n);
68         m = (clock()-t)/CLOCKS_PER_SEC; // tempo do msort
69         printf("%d %5.1f %5.1f\n",n,b,m);
70     }
71     return 0;
72 }
73 }
```

```
C:\Users\cloud\OneDrive - Fat  X + v
n bsort msort
10000 0.3 0.0
20000 1.3 0.0
30000 4.0 0.0
40000 7.1 0.0
50000 12.7 0.0
60000 17.0 0.0
70000 21.1 0.0
80000 28.3 0.0
90000 37.0 0.0
```

Exercício 8

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <math.h>
5
6 void troca(int v[], int i, int j) {
7     int temp = v[i];
8     v[i] = v[j];
9     v[j] = temp;
10 }
11
12 void bsort(int v[], int n) {
13     for(int i=1; i<n; i++)
14         for(int j=0; j<n-i; j++)
15             if( v[j]>v[j+1] )
16                 troca(v,j,j+1);
17 }
18
19 void ms(int v[], int p, int u) {
20     if( p==u ) return;
21     int m = (p+u)/2;
22     ms(v,p,m);
23     ms(v,m+1,u);
24     intercala(v,p,m,u);
25 }
26 void msort(int v[], int n) {
27     ms(v,0,n-1);
28 }
29
30 void intercala(int v[], int p, int m, int u) {
31     int *w = malloc((u-p+1)*sizeof(int));
32     int i=p, j=m+1, k=0;
33     while( i<=m && j<=u )
34         w[k++] = (v[i]<v[j]) ? v[i++] : v[j++];
35     while( i<=m ) w[k++] = v[i++];
36     while( j<=u ) w[k++] = v[j++];
37     for(k=0; k<=u-p; k++) v[p+k] = w[k];
38     free(w);
39 }
```

```
C:\Users\cloud\OneDrive - Fat  X + v
n msort
10000000 3.6
20000000 6.1
30000000 9.8
40000000 12.7
50000000 16.6
60000000 19.9
```

Exercício 10

```
1 #include <stdio.h>
2
3
4 int bsearch(int x, int v[], int n) {
5     int p = 0;
6     int u = n-1;
7     while( p<=u ) {
8         int m = (p+u)/2;
9         if( x==v[m] ) return 1;
10        if( x<v[m] ) u = m-1;
11        else p = m+1;
12    }
13    return 0;
14 }
15
16 int main(void) {
17     int v[8] = {19,27,31,48,52,66,75,80};
18     printf("27: %d\n", bsearch(27,v,8));
19     printf("51: %d\n", bsearch(51,v,8));
20     return 0;
21 }
22
```

```
C:\Users\cloud\OneDrive - Fat  X + v
27: 1
51: 0

-----
Process exited after 0.04988 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 11

```
1 #include <stdio.h>
2
3 int rlsearch(int x, int v[], int n, int i) {
4     if( i == n )
5         return 0;
6     if( v[i] == x )
7         return 1;
8     return rlsearch(x, v, n, i + 1);
9 }
10
11 int main(void) {
12     int v[8] = {66,80,31,48,27,75,19,52};
13     printf("27: %d\n", rlsearch(27,v,8,0));
14     printf("51: %d\n", rlsearch(51,v,8,0));
15     return 0;
16 }
17
18
19
20
21
22
23
```

```
C:\Users\cloud\OneDrive - Fat  X + v
27: 1
51: 0

-----
Process exited after 0.04611 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

Exercício 12

```
Ex4.c  Ex5.c  Ex6.c  Ex7.c  Ex8.c  Ex9.c  Ex10.c  Ex11.c  [*] Ex12.c
1      #include <stdio.h>
2
3      int rbsearch(int x, int v[], int p, int u) {
4
5          if (p > u) {
6              return 0;
7          }
8
9          int meio = (p + u) / 2;
10
11         if (v[meio] == x) {
12             return 1;
13         }
14
15         else if (x < v[meio]) {
16             return rbsearch(x, v, p, meio - 1);
17         }
18
19         else {
20             return rbsearch(x, v, meio + 1, u);
21         }
22     }
23
24     int main(void) {
25         int v[8] = {19,27,31,48,52,66,75,88};
26         printf("27: %d\n", rbsearch(27,v,0,7));
27         printf("51: %d\n", rbsearch(51,v,0,7));
28         return 0;
29     }
30
```

```
C:\Users\cloud\OneDrive - Fat  X + -
27: 1
51: 0

-----
Process exited after 0.07649 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```