

## exercicio1-1.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void selectionSort(int *vetor, int N){
5      int i, j, menor;
6      for (i=0; i < N-1; i++){
7          menor = i;
8          for (j=i+1; j < N; j++){
9              if (vetor[j] < vetor[menor])
10                 menor = j;
11          }
12
13          if (i != menor){
14              int aux = vetor[i];
15              vetor[i] = vetor[menor];
16              vetor[menor] = aux;
17          }
18      }
19  }
20
21 void insertionSort(int *vetor, int N){
22     int i, j, atual;
23     for (i=1; i<N; i++){
24         atual = vetor[i];
25         for (j=i; (j>0) && (atual < vetor[j-1]); j--){
26             vetor[j] = vetor[j-1];
27         }
28         vetor[j] = atual;
29     }
30 }
31
32 void bubbleSort(int *vetor, int N){
33     int i, j;
34     for (i=0; i < N-1; i++){
35         for(j=0; j < N-i-1; j++){
36             if (vetor[j] > vetor[j+1]){
37                 int aux = vetor[j];
38                 vetor[j] = vetor[j+1];
39                 vetor[j+1] = aux;
40             }
41         }
42     }
43 }
44
45 int main(){
46     int N;
47     printf ("Informe o tamanho do vetor: ");
48     scanf ("%d", &N);
49     int vetor[N];
50     for (int i = 0; i < N; i++){
51         printf ("vetor[%d] = ", i);
52         scanf ("%d", &vetor[i]);
53     }
54     printf ("Vetor Desordenado: ");
55     for (int i = 0; i < N; i++){
56         printf ("vetor[%d] = %d\n", i, vetor[i]);
57     }
```

```
58
59     printf ("Qual algoritmo deseja usar?\n");
60     printf ("1 - Selection Sort\n");
61     printf ("2 - Insertion Sort\n");
62     printf ("3 - Bubble Sort\n");
63     int opcao;
64     scanf ("%d", &opcao);
65
66     if (opcao == 1){
67         selectionSort(vetor, N);
68         printf ("Vetor Ordenado: \n");
69         for (int i = 0; i < N; i++){
70             printf ("vetor[%d] = %d\n", i, vetor[i]);
71         }
72     }
73     else if (opcao == 2){
74         insertionSort(vetor, N);
75         printf ("Vetor Ordenado: \n");
76         for (int i = 0; i < N; i++){
77             printf ("vetor[%d] = %d\n", i, vetor[i]);
78         }
79     }
80     else if (opcao == 3){
81         bubbleSort(vetor, N);
82         printf ("Vetor Ordenado: \n");
83         for (int i = 0; i < N; i++){
84             printf ("vetor[%d] = %d\n", i, vetor[i]);
85         }
86     }
87 }
```

```
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> cd 'c:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output'
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> & .\'exercicio1-1.exe'
Informe o tamanho do vetor: 5
vetor[0] = 3
vetor[1] = 2
vetor[2] = 4
vetor[3] = 5
vetor[4] = 1
Vetor Desordenado: vetor[0] = 3
vetor[1] = 2
vetor[2] = 4
vetor[3] = 5
vetor[4] = 1
Qual algoritmo deseja usar?
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
1
Vetor Ordenado:
vetor[0] = 1
vetor[1] = 2
vetor[2] = 3
vetor[3] = 4
vetor[4] = 5
```

## exercicio1-2.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7  void selectionSort(int *vetor, int N){
8      int i, j, maior;
9      for (i=0; i < N-1; i++){
10         maior = i;
11         for (j=i+1; j < N; j++){
12             if (vetor[j] > vetor[maior])
13                 maior = j;
14         }
15
16         if (i != maior){
17             int aux = vetor[i];
18             vetor[i] = vetor[maior];
19             vetor[maior] = aux;
20         }
21     }
22 }
23
24 void insertionSort(int *vetor, int N){
25     int i, j, atual;
26     for (i=1; i<N; i++){
27         atual = vetor[i];
28         for (j=i; (j>0) && (atual > vetor[j-1]); j--){
29             vetor[j] = vetor[j-1];
30         }
31         vetor[j] = atual;
32     }
33 }
34
35 void bubbleSort(int *vetor, int N){
36     int i, j;
37     for (i=0; i < N-1; i++){
38         for(j=0; j < N-i-1; j++){
39             if (vetor[j] < vetor[j+1]){
40                 int aux = vetor[j];
41                 vetor[j] = vetor[j+1];
42                 vetor[j+1] = aux;
43             }
44         }
45     }
46 }
47
48 int main(){
49     int N;
50     printf ("Informe o tamanho do vetor: ");
51     scanf ("%d", &N);
52     int vetor[N];
53     for (int i = 0; i < N; i++){
54         printf ("vetor[%d] = ", i);
55         scanf ("%d", &vetor[i]);
56     }
57     printf ("Vetor Desordenado: ");
```

```
58     for (int i = 0; i < N; i++){
59         printf ("vetor[%d] = %d\n", i, vetor[i]);
60     }
61
62     printf ("Qual algoritmo deseja usar?\n");
63     printf ("1 - Selection Sort\n");
64     printf ("2 - Insertion Sort\n");
65     printf ("3 - Bubble Sort\n");
66     int opcao;
67     scanf ("%d", &opcao);
68
69     if (opcao == 1){
70         selectionSort(vetor, N);
71         printf ("Vetor Ordenado Decrescente: \n");
72         for (int i = 0; i < N; i++){
73             printf ("vetor[%d] = %d\n", i, vetor[i]);
74         }
75     }
76     else if (opcao == 2){
77         insertionSort(vetor, N);
78         printf ("Vetor Ordenado Decrescente: \n");
79         for (int i = 0; i < N; i++){
80             printf ("vetor[%d] = %d\n", i, vetor[i]);
81         }
82     }
83     else if (opcao == 3){
84         bubbleSort(vetor, N);
85         printf ("Vetor Ordenado Decrescente: \n");
86         for (int i = 0; i < N; i++){
87             printf ("vetor[%d] = %d\n", i, vetor[i]);
88         }
89     }
90 }
```

```
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> cd 'c:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output'
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> & .\'exercicio1-2.exe'
Informe o tamanho do vetor: 5
vetor[0] = 2
vetor[1] = 3
vetor[2] = 1
vetor[3] = 5
vetor[4] = 4
Vetor Desordenado: vetor[0] = 2
vetor[1] = 3
vetor[2] = 1
vetor[3] = 5
vetor[4] = 4
Qual algoritmo deseja usar?
1 - Selection Sort
2 - Insertion Sort
3 - Bubble Sort
2
Vetor Ordenado Decrescente:
vetor[0] = 5
vetor[1] = 4
vetor[2] = 3
vetor[3] = 2
vetor[4] = 1
```

## exercicio1-3.c

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include <windows.h>
5
6  #define TAMANHO_ARQUIVO 100
7
8  void selectionSort(int *vetor, int N){
9      int i, j, menor, mov, comp;
10     mov = comp = 0;
11     for (i=0; i < N-1; i++){
12         menor = i;
13         for (j=i+1; j < N; j++){
14             comp++;
15             if (vetor[j] < vetor[menor])
16                 menor = j;
17         }
18
19         if (i != menor){
20             int aux = vetor[i];
21             vetor[i] = vetor[menor];
22             vetor[menor] = aux;
23             mov++;
24         }
25     }
26     printf ("Comparacoes: %d\nMovimentacoes: %d\n", comp, mov);
27 }
28
29 void insertionSort(int *vetor, int N){
30     int i, j, atual, comp, mov;
31     mov = comp = 0;
32     for (i=1; i<N; i++){
33         atual = vetor[i];
34         comp++;
35         for (j=i; (j>0) && (atual < vetor[j-1]); j--){
36             vetor[j] = vetor[j-1];
37             comp++;
38             mov++;
39         }
40         vetor[j] = atual;
41     }
42     printf ("Comparacoes: %d\nMovimentacoes: %d\n", comp, mov);
43 }
44
45 void bubbleSort(int *vetor, int N){
46     int i, j, comp, mov;
47     mov = comp = 0;
48     for (i=0; i < N-1; i++){
49         for(j=0; j < N-i-1; j++){
50             comp++;
51             if (vetor[j] > vetor[j+1]){
52                 int aux = vetor[j];
53                 vetor[j] = vetor[j+1];
54                 vetor[j+1] = aux;
55                 mov++;
56             }
57         }
58     }
```

```
58     }
59     printf ("Comparacoes: %d\nMovimentacoes: %d\n", comp, mov);
60 }
61
62 int main(){
63     FILE *arquivo;
64     char nomeArquivo[TAMANHO_ARQUIVO];
65
66     printf("Digite o nome do arquivo .txt que esta na pasta 'entradas': ");
67     scanf("%s", nomeArquivo);
68
69     char caminhoArquivo[TAMANHO_ARQUIVO];
70     GetModuleFileName(NULL, caminhoArquivo, TAMANHO_ARQUIVO);
71     char *ultimoBarra = strrchr(caminhoArquivo, '\\');
72     if (ultimoBarra != NULL) {
73         *ultimoBarra = '\\0';
74     }
75
76     snprintf(caminhoArquivo, TAMANHO_ARQUIVO, "%s\\entradas\\%s.txt", caminhoArquivo,
77 nomeArquivo);
78
79     arquivo = fopen(caminhoArquivo, "r");
80     printf("Caminho do arquivo: %s\n", caminhoArquivo);
81     if (arquivo == NULL) {
82         printf("Erro ao abrir o arquivo.\n");
83         return 1;
84     }
85
86     int n;
87     fscanf(arquivo, "%d", &n);
88     int vet[n], vetor_BubbleSort[n], vetor_SelectionSort[n], vetor_InsertionSort[n];
89     int i = 0;
90
91     while(!feof(arquivo)){
92         fscanf(arquivo, "%d", &vet[i]);
93         vetor_BubbleSort[i] = vet[i];
94         vetor_SelectionSort[i] = vet[i];
95         vetor_InsertionSort[i] = vet[i];
96         i++;
97     }
98
99     clock_t t1;
100     t1 = clock();
101     bubbleSort(vetor_BubbleSort, n);
102     t1 = clock() - t1;
103     printf ("%f seconds.\n", ((float)t1) / CLOCKS_PER_SEC);
104     printf("\n");
105
106     clock_t t2;
107     t2 = clock();
108     selectionSort(vetor_SelectionSort, n);
109     t2 = clock() - t2;
110     printf ("%f seconds.\n", ((float)t2) / CLOCKS_PER_SEC);
111     printf("\n");
112
113     clock_t t3;
114     t3 = clock();
115     insertionSort(vetor_InsertionSort, n);
116     t3 = clock() - t3;
117     printf ("%f seconds.\n", ((float)t3) / CLOCKS_PER_SEC);
```



```
117     printf("\n");
118
119     fclose(arquivo);
120     return 0;
121 }
122
```

PS C:\Users\USER\OneDrive\Documentos\Lista10\output> cd 'c:\Users\USER\OneDrive\Documentos\Lista10\output'  
PS C:\Users\USER\OneDrive\Documentos\Lista10\output> & .\'exercicio1-3.exe'  
Digite o nome do arquivo .txt que esta na pasta 'entradas': 100-misturado  
Caminho do arquivo: C:\Users\USER\OneDrive\Documentos\Lista10\output\entradas\100-misturado.txt  
Comparacoes: 4950  
Movimentacoes: 2339  
0.000000 seconds.

Comparacoes: 4950  
Movimentacoes: 94  
0.000000 seconds.

Comparacoes: 2525  
Movimentacoes: 2426  
0.000000 seconds.

Digite o nome do arquivo .txt que esta na pasta 'entradas': 1000-misturado  
Caminho do arquivo: C:\Users\USER\OneDrive\Documentos\Lista10\output\entradas\1000-misturado.txt  
Comparacoes: 499500  
Movimentacoes: 232469  
0.001000 seconds.

Comparacoes: 499500  
Movimentacoes: 988  
0.001000 seconds.

Comparacoes: 233673  
Movimentacoes: 232674  
0.001000 seconds.

Digite o nome do arquivo .txt que esta na pasta 'entradas': 10000-misturado  
Caminho do arquivo: C:\Users\USER\OneDrive\Documentos\Lista10\output\entradas\10000-misturado.txt  
Comparacoes: 49995000  
Movimentacoes: 24774110  
0.216000 seconds.

Comparacoes: 49995000  
Movimentacoes: 9986  
0.101000 seconds.

Comparacoes: 24789169  
Movimentacoes: 24779170  
0.063000 seconds.

Digite o nome do arquivo .txt que esta na pasta 'entradas': 100000-misturado  
Caminho do arquivo: C:\Users\USER\OneDrive\Documentos\Lista10\output\entradas\100000-misturado.txt  
Comparacoes: 704982704  
Movimentacoes: 1800360435  
25.350000 seconds.

Comparacoes: 704982704  
Movimentacoes: 99988  
9.776000 seconds.

Comparacoes: 1800173298  
Movimentacoes: 1800273297  
6.085000 seconds.

## exercicio1-4.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct Pessoa{
6      char nome[50];
7      int idade;
8  } Pessoa;
9
10 void selectionSortCrescente(Pessoa *vetor, int N){
11     int i, j, menor;
12     for (i=0; i<N-1; i++){
13         menor = i;
14         for (j=i+1; j<N; j++){
15             if (strcmp(vetor[j].nome, vetor[menor].nome) < 0 || strcmp(vetor[j].nome,
vetor[menor].nome) == 0 && vetor[j].idade < vetor[menor].idade)
16                 menor = j;
17         }
18
19         if (i != menor){
20             Pessoa aux = vetor[i];
21             vetor[i] = vetor[menor];
22             vetor[menor] = aux;
23         }
24     }
25 }
26
27 void selectionSortDecrescente(Pessoa *vetor, int N){
28     int i, j, menor;
29     for (i=0; i<N-1; i++){
30         menor = i;
31         for (j=i+1; j<N; j++){
32             if (strcmp(vetor[j].nome, vetor[menor].nome) > 0 || strcmp(vetor[j].nome,
vetor[menor].nome) == 0 && vetor[j].idade > vetor[menor].idade)
33                 menor = j;
34         }
35
36         if (i != menor){
37             Pessoa aux = vetor[i];
38             vetor[i] = vetor[menor];
39             vetor[menor] = aux;
40         }
41     }
42 }
43
44 void insertionSortCrescente(Pessoa *vetor, int N){
45     int i, j;
46     Pessoa atual;
47     for (i=1; i<N; i++){
48         atual = vetor[i];
49         j = i;
50         while ((j > 0) && (strcmp(vetor[j-1].nome, atual.nome) < 0 || strcmp(vetor[j-1]
.nome, atual.nome) == 0 && vetor[j-1].idade < atual.idade)){
51             vetor[j] = vetor[j-1];
52             j--;
53         }
54         vetor[j] = atual;
55     }
```

```
56 }
57
58 void insertionSortDecrescente(Pessoa *vetor, int N){
59     int i, j;
60     Pessoa atual;
61     for (i=1; i<N; i++){
62         atual = vetor[i];
63         j = i;
64         while ((j > 0) && (strcmp(vetor[j-1].nome, atual.nome) > 0 || strcmp(vetor[j-1]
        .nome, atual.nome) == 0 && vetor[j-1].idade > atual.idade)){
65             vetor[j] = vetor[j-1];
66             j--;
67         }
68         vetor[j] = atual;
69     }
70 }
71
72 int main(){
73     int N;
74     printf ("Quantas pessoas serão informadas? ");
75     scanf ("%d", &N);
76     Pessoa vetor[N];
77     for (int i = 0; i < N; i++){
78         printf ("Nome: ");
79         scanf (" %s", &vetor[i].nome);
80         printf ("Idade: ");
81         scanf ("%d", &vetor[i].idade);
82     }
83
84     printf ("Qual algoritmo deseja usar?\n");
85     printf ("1 - Selection Sort\n");
86     printf ("2 - Insertion Sort\n");
87     int opcao;
88     scanf ("%d", &opcao);
89
90     if (opcao == 1){
91         selectionSortCrescente(vetor, N);
92         for (int i = 0; i < N; i++){
93             printf ("Nome: %s, Idade: %d\n", vetor[i].nome, vetor[i].idade);
94         }
95
96         selectionSortDecrescente(vetor, N);
97         printf ("Vetor Ordenado: \n");
98         for (int i = 0; i < N; i++){
99             printf ("Nome: %s, Idade: %d\n", vetor[i].nome, vetor[i].idade);
100         }
101     }
102     else if (opcao == 2){
103         insertionSortCrescente(vetor, N);
104         for (int i = 0; i < N; i++){
105             printf ("Nome: %s, Idade: %d\n", vetor[i].nome, vetor[i].idade);
106         }
107
108         insertionSortDecrescente(vetor, N);
109         printf ("Vetor Ordenado: \n");
110         for (int i = 0; i < N; i++){
111             printf ("Nome: %s, Idade: %d\n", vetor[i].nome, vetor[i].idade);
112         }
113     }
114 }
```

```
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> cd 'c:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output'
PS C:\Users\USER\OneDrive\Área de Trabalho\LabProg2\Lista 10\output> & .\'exercicio1-4.exe'
Quantas pessoas serão informadas? 3
Nome: Gabriel
Idade: 20
Nome: Pedro
Idade: 19
Nome: Pedro
Idade: 21
Qual algoritmo deseja usar?
1 - Selection Sort
2 - Insertion Sort
1
Nome: Gabriel, Idade: 20
Nome: Pedro, Idade: 19
Nome: Pedro, Idade: 21
Vetor Ordenado:
Nome: Pedro, Idade: 21
Nome: Pedro, Idade: 19
Nome: Gabriel, Idade: 20
```