

Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Frequência F11

Ordenação por inserção

```
// programa para ordenacao por insercao
#include <bits/stdc++.h>
using namespace std;

/* funcao pra ordenar por insercao */
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;

        /* Move elementos de arr[0..i-1], que sao
        maiores que key, para uma posicao a frente
        da sua atual */
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

// uma funcao para printar uma array de tamanho n
void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int main()
{
    int arr[] = { 12, 11, 13, 5, 6 };
```

```

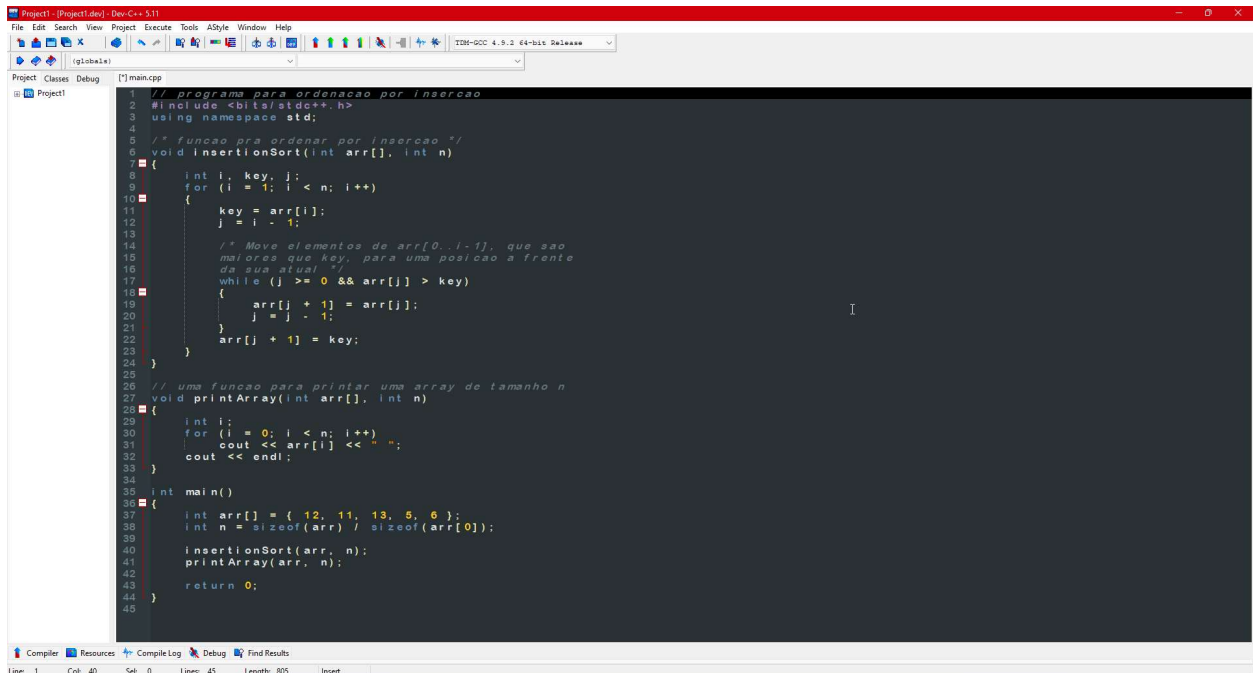
    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}

```

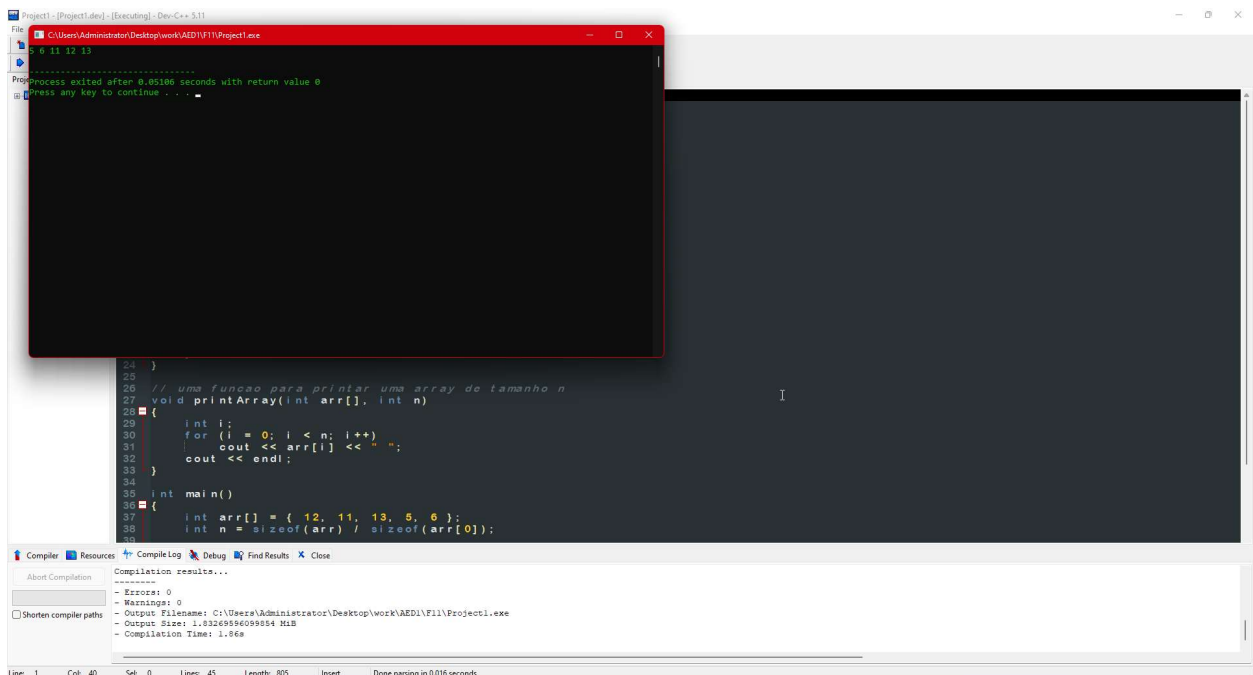
Prints da execução



```

1 // programa para ordenacao por insercao
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 // funcao pra ordenar por insercao
6 void insertionSort(int arr[], int n)
7 {
8     int i, key, j;
9     for (i = 1; i < n; i++)
10     {
11         key = arr[i];
12         j = i - 1;
13
14         // Move elementos de arr[0..i-1], que sao
15         // maiores que key, para uma posicao a frente
16         // da sua atual
17         while (j >= 0 && arr[j] > key)
18         {
19             arr[j + 1] = arr[j];
20             j = j - 1;
21         }
22         arr[j + 1] = key;
23     }
24 }
25
26 // uma funcao para imprimir uma array de tamanho n
27 void printArray(int arr[], int n)
28 {
29     int i;
30     for (i = 0; i < n; i++)
31         cout << arr[i] << " ";
32     cout << endl;
33 }
34
35 int main()
36 {
37     int arr[] = { 12, 11, 13, 5, 6 };
38     int n = sizeof(arr) / sizeof(arr[0]);
39
40     insertionSort(arr, n);
41     printArray(arr, n);
42
43     return 0;
44 }
45

```



```

5 6 11 12 13
-----
Process exited after 0.05106 seconds with return value 0
Press any key to continue . . .

```

```

24 }
25
26 // uma funcao para imprimir uma array de tamanho n
27 void printArray(int arr[], int n)
28 {
29     int i;
30     for (i = 0; i < n; i++)
31         cout << arr[i] << " ";
32     cout << endl;
33 }
34
35 int main()
36 {
37     int arr[] = { 12, 11, 13, 5, 6 };
38     int n = sizeof(arr) / sizeof(arr[0]);
39

```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED\TF11\Project1.exe
- Output Size: 1.3360960939514 KB
- Compilation Time: 1.06s

Ordenação por inserção

```
// programa para ordenacao por selecao
#include <bits/stdc++.h>
using namespace std;

void troca(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selecao(int arr[], int n)
{
    int i, j, min_idx;

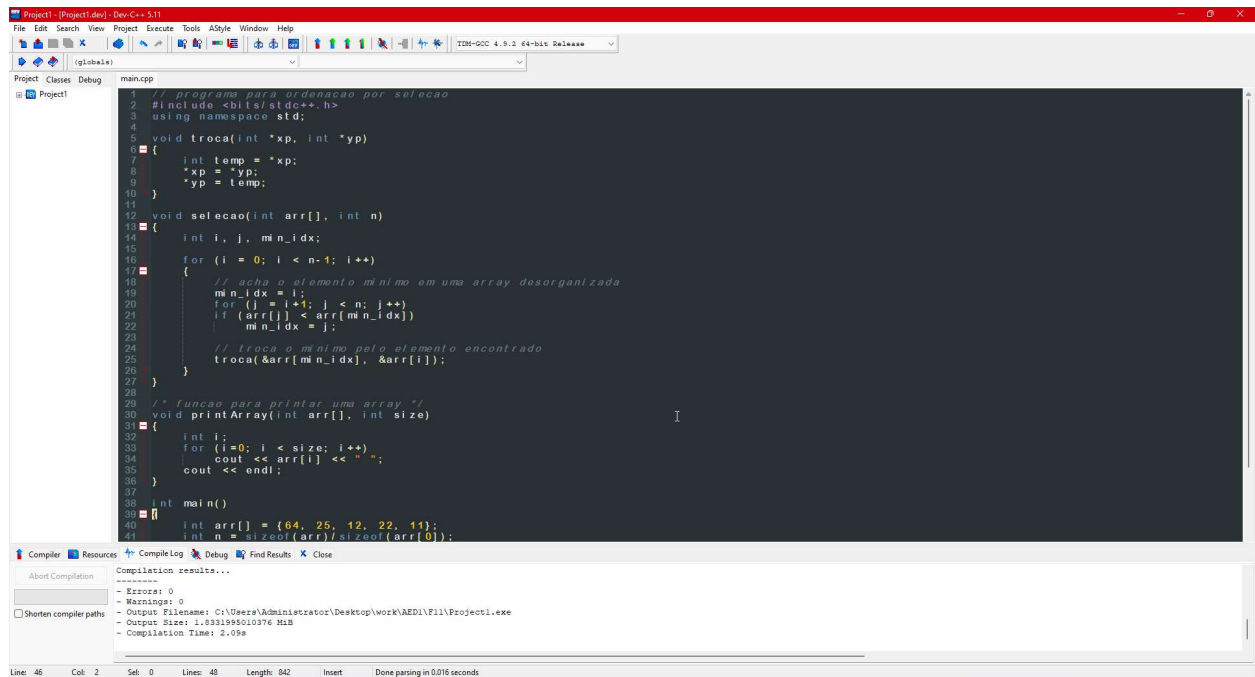
    for (i = 0; i < n-1; i++)
    {
        // acha o elemento minimo em uma array desorganizada
        min_idx = i;
        for (j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        // troca o minimo pelo elemento encontrado
        troca(&arr[min_idx], &arr[i]);
    }
}

/* funcao para printar uma array */
void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}

int main()
{
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr)/sizeof(arr[0]);
    selecao(arr, n);
    cout << "Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```

Prints da execução

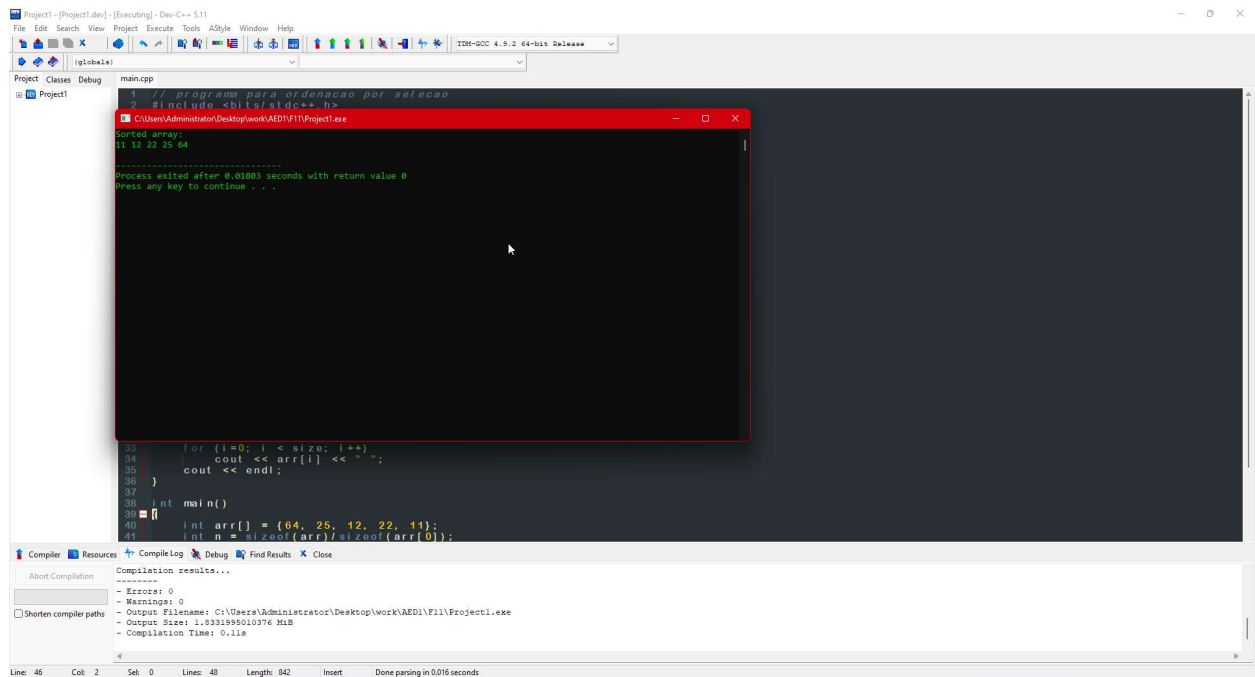


```
1 // programa para ordenacao por selecao
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 void troca(int *xp, int *yp)
6 {
7     int temp = *xp;
8     *xp = *yp;
9     *yp = temp;
10 }
11
12 void selecao(int arr[], int n)
13 {
14     int i, j, min_idx;
15     for (i = 0; i < n-1; i++)
16     {
17         // acha o elemento minimo em uma array desorganizada
18         min_idx = i;
19         for (j = i+1; j < n; j++)
20             if (arr[j] < arr[min_idx])
21                 min_idx = j;
22         // troca o minimo pelo elemento encontrado
23         troca(&arr[min_idx], &arr[i]);
24     }
25 }
26
27 // funcao para imprimir uma array
28 void printArray(int arr[], int size)
29 {
30     int i;
31     for (i=0; i < size; i++)
32         cout << arr[i] << " ";
33     cout << endl;
34 }
35
36 int main()
37 {
38     int arr[] = {64, 25, 12, 22, 11};
39     int n = sizeof(arr)/sizeof(arr[0]);
40     selecao(arr, n);
41     printArray(arr, n);
42 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED1\F11\Project1.exe
- Output Size: 1.8331995010376 MiB
- Compilation Time: 2.09s

Line: 46 Col: 2 Sel: 0 Lines: 46 Length: 842 Insert Done parsing in 0.016 seconds



```
1 // programa para ordenacao por selecao
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 void troca(int *xp, int *yp)
6 {
7     int temp = *xp;
8     *xp = *yp;
9     *yp = temp;
10 }
11
12 void selecao(int arr[], int n)
13 {
14     int i, j, min_idx;
15     for (i = 0; i < n-1; i++)
16     {
17         // acha o elemento minimo em uma array desorganizada
18         min_idx = i;
19         for (j = i+1; j < n; j++)
20             if (arr[j] < arr[min_idx])
21                 min_idx = j;
22         // troca o minimo pelo elemento encontrado
23         troca(&arr[min_idx], &arr[i]);
24     }
25 }
26
27 // funcao para imprimir uma array
28 void printArray(int arr[], int size)
29 {
30     int i;
31     for (i=0; i < size; i++)
32         cout << arr[i] << " ";
33     cout << endl;
34 }
35
36 int main()
37 {
38     int arr[] = {64, 25, 12, 22, 11};
39     int n = sizeof(arr)/sizeof(arr[0]);
40     selecao(arr, n);
41     printArray(arr, n);
42 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Administrator\Desktop\work\AED1\F11\Project1.exe
- Output Size: 1.8331995010376 MiB
- Compilation Time: 0.11s

Line: 46 Col: 2 Sel: 0 Lines: 46 Length: 842 Insert Done parsing in 0.016 seconds