Sure, let's break down the provided C++ code line by line:

cpp
Copy code

This is a comment describing that this program implements the selection sort algorithm.

cpp
Copy code
include <bits/stdc++.h> using namespace

This includes necessary header files for the program. `<bits/stdc++.h>` is a header that includes all standard libraries. `using namespace std;` is a directive that allows identifiers in the `std` namespace to be used without qualification.

cpp
Copy code
void selectionSort int int int
for 0 1

This begins the definition of the `selectionSort` function. It takes an integer array `arr[]` and its size `n` as parameters. It initializes integer variables `i`, `j`, and `min_idx`.

The `for` loop iterates over the elements of the array `arr[]` up to the second-to-last element (`n - 1`). This loop controls the boundary of the unsorted subarray.

cpp
Copy code
for 1 if

Within the outer loop, another loop is used to find the index of the minimum element in the unsorted portion of the array. `min_idx` is updated if a smaller element is found.

cpp
Copy code
if swap

After finding the minimum element (`min_idx`), it swaps it with the element at index `i` if `min_idx` is different from `i`. This effectively moves the smallest element to the correct position in the sorted portion of the array.

cpp
Copy code

```
void printArray int int int for 0
" "
```

This defines a function named **`printArray`** which takes an integer array **`arr[]`** and its size **`size`** as parameters. It iterates over the array elements and prints them out, each followed by a newline.

cpp
Copy code

```
int main int 64 25 12 22 11 int sizeof sizeof 0
selectionSort "Sorted array: \n" printArray return 0
```

This is the **`main`** function where the program starts execution. It defines an integer array **`arr[]`** with some initial values. It calculates the size of the array using the **`sizeof`** operator and divides it by the size of its first element to determine the number of elements in the array (**`n`**).

It then calls the **`selectionSort`** function passing the array **`arr[]`** and its size **`n`**. After sorting, it prints the sorted array using the **`printArray`** function. Finally, it returns **`0`**, indicating successful execution of the program.