

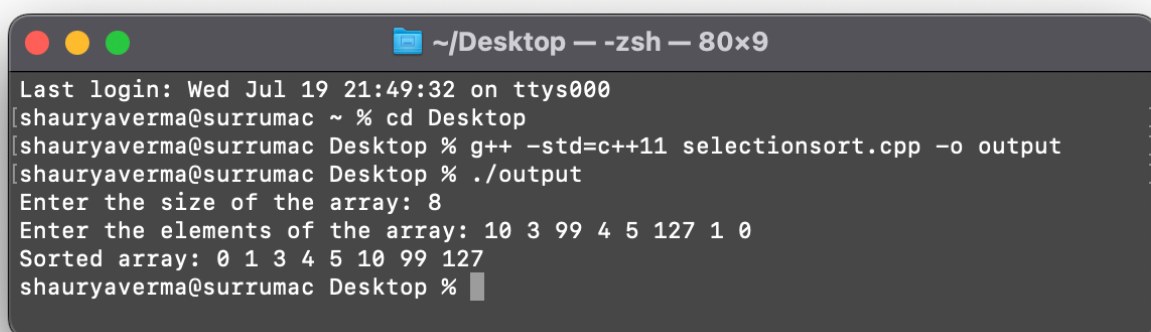
This document presents a C++ implementation of the Selection Sort algorithm, an efficient and straightforward sorting technique for arrays. The code aims to sort an array of integers in ascending order using the Selection Sort method. It allows users to input the size of the array and its elements, and then it applies the sorting algorithm to produce the sorted output.

The Selection Sort algorithm is based on the idea of repeatedly selecting the minimum (or maximum) element from an unsorted part of the array and moving it to the beginning (or end) of the sorted part. This process is iteratively performed until the entire array becomes sorted.

The `selectionSort()` function implements the Selection Sort algorithm. It takes a vector of integers as input and sorts the elements in ascending order.

- **Parameters:** The function takes a reference to a vector of integers named `arr` as input, which represents the array to be sorted.
- **Process:** The function iterates through the array using two nested loops. The outer loop (variable `i`) goes from 0 to `n-1`, where `n` is the size of the array. The inner loop (variable `j`) goes from `i+1` to `n`.
- **Finding Minimum:** Within the inner loop, the function compares the element at `arr[j]` with the current minimum element at `arr[minIndex]`. If `arr[j]` is found to be smaller than `arr[minIndex]`, the value of `minIndex` is updated to `j`.
- **Swapping:** After the inner loop completes, the function swaps the element at index `i` (current position) with the element at index `minIndex` (minimum element). This moves the minimum element to its correct position in the sorted part of the array.
- **Sorted Array:** After the outer loop finishes, the array will be sorted in ascending order.

OUTPUT –



```
~/Desktop — zsh — 80x9
Last login: Wed Jul 19 21:49:32 on ttys000
[shauryaverma@surrumac ~ % cd Desktop
[shauryaverma@surrumac Desktop % g++ -std=c++11 selectionsort.cpp -o output
[shauryaverma@surrumac Desktop % ./output
Enter the size of the array: 8
Enter the elements of the array: 10 3 99 4 5 127 1 0
Sorted array: 0 1 3 4 5 10 99 127
shauryaverma@surrumac Desktop %
```

The Selection Sort algorithm has an average time complexity of $O(n^2)$, making it less efficient than more advanced sorting algorithms like Quick Sort or Merge Sort. However, Selection Sort has the advantage of simplicity and can perform well for small arrays or in cases where memory usage is a concern.