QUICK SORT

Q. Program-4: Using iteration & recursion concepts write the program for Quick Sort technique

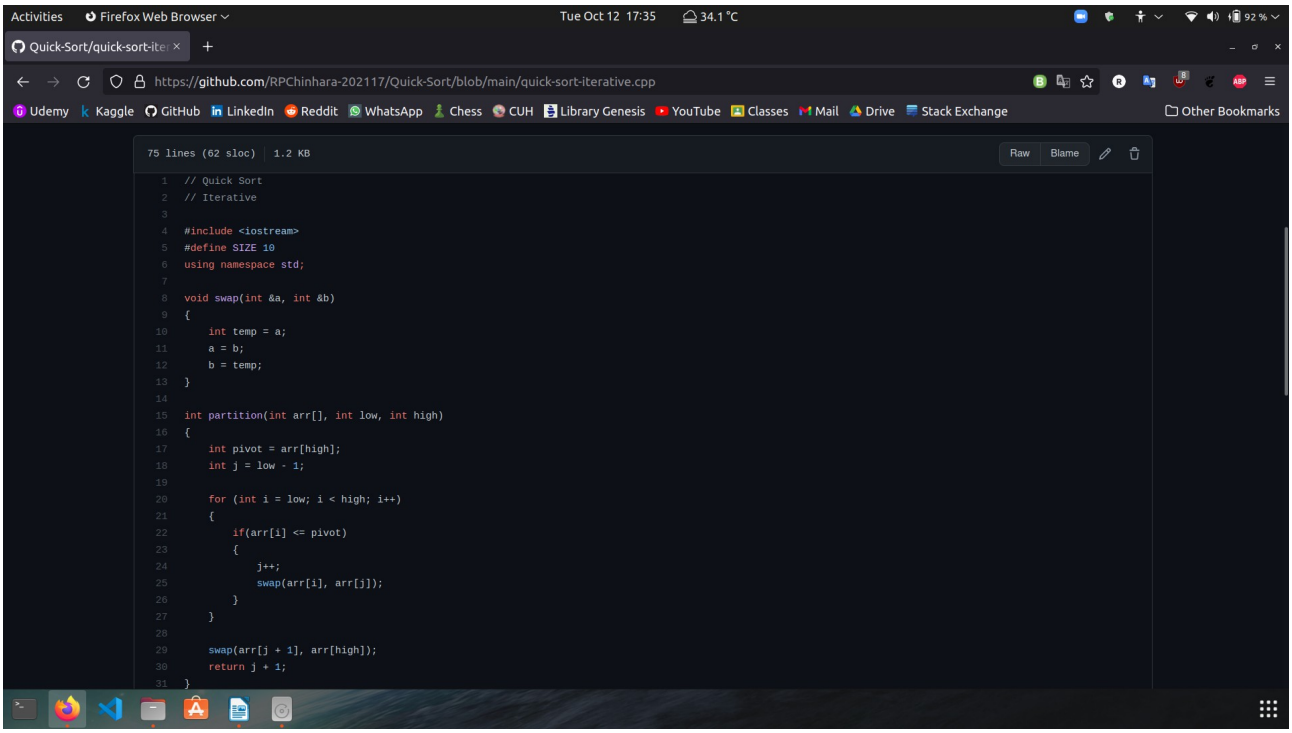
Made by:-

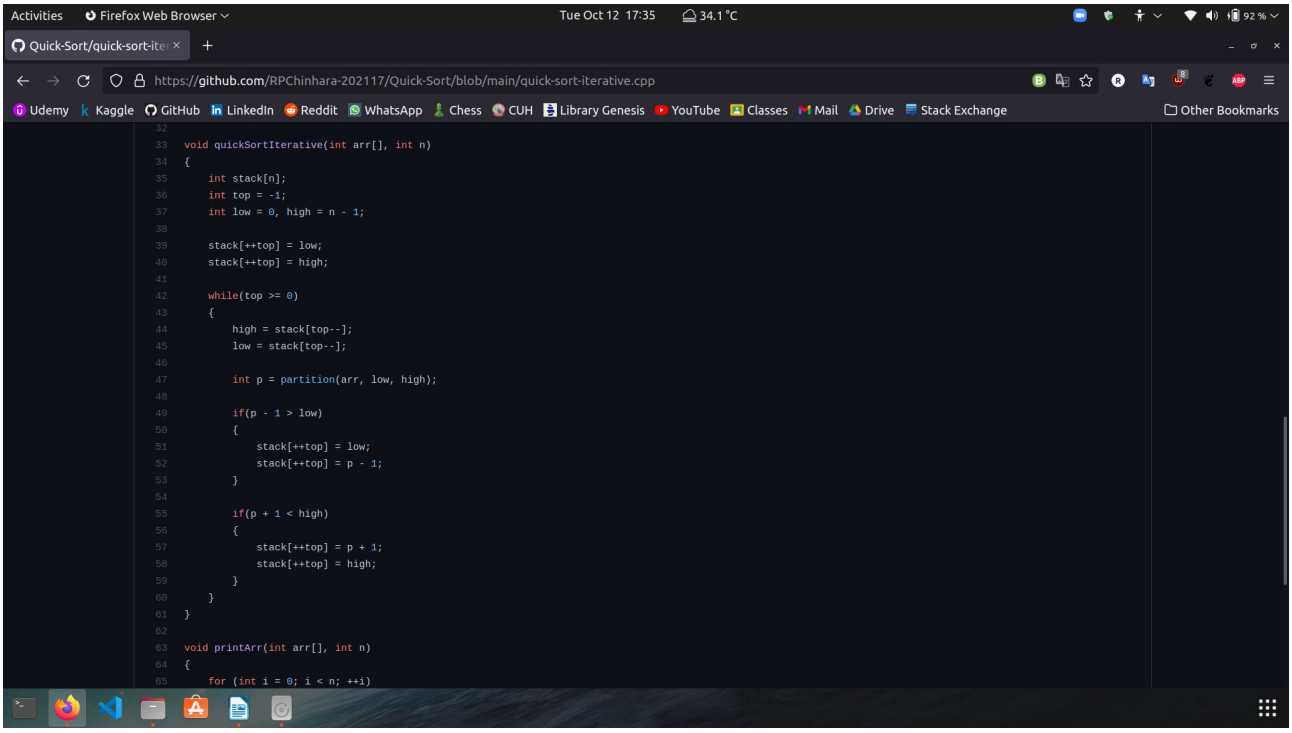
Rudra Chinhara

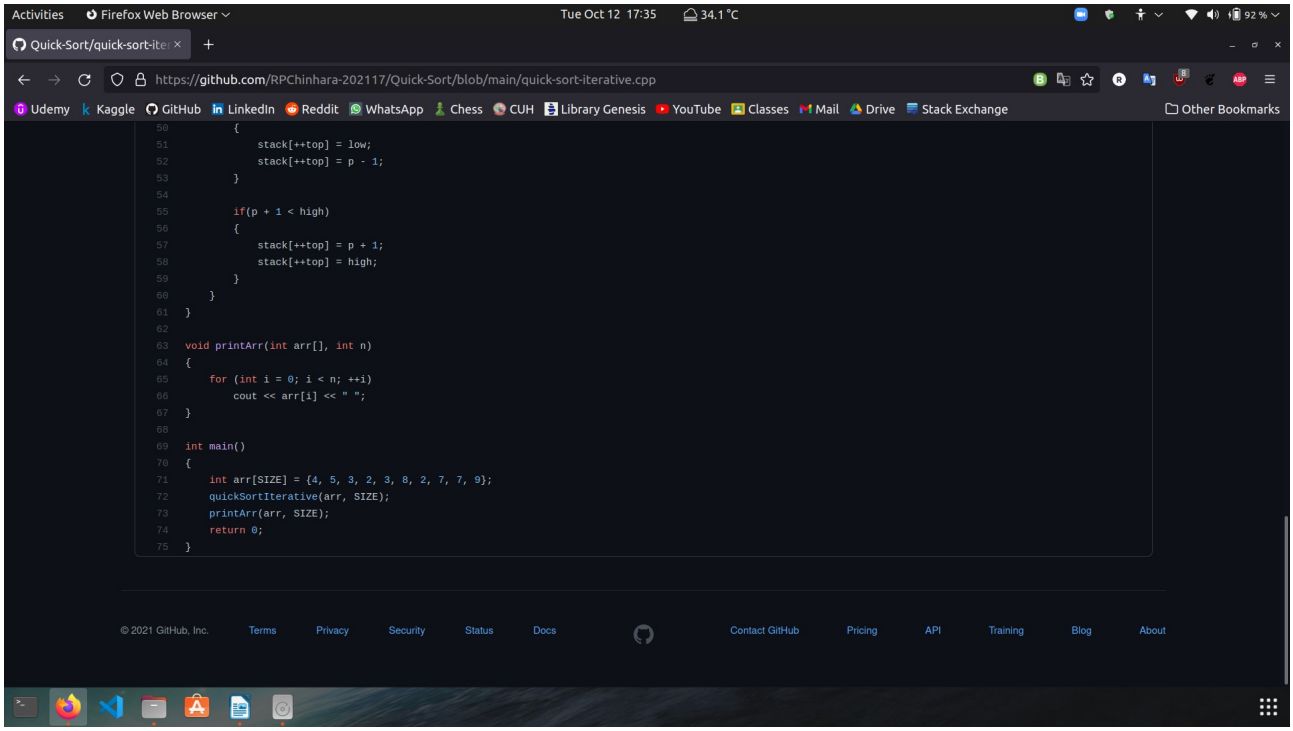
B. Tech CSE, 202117

Central University of Haryana

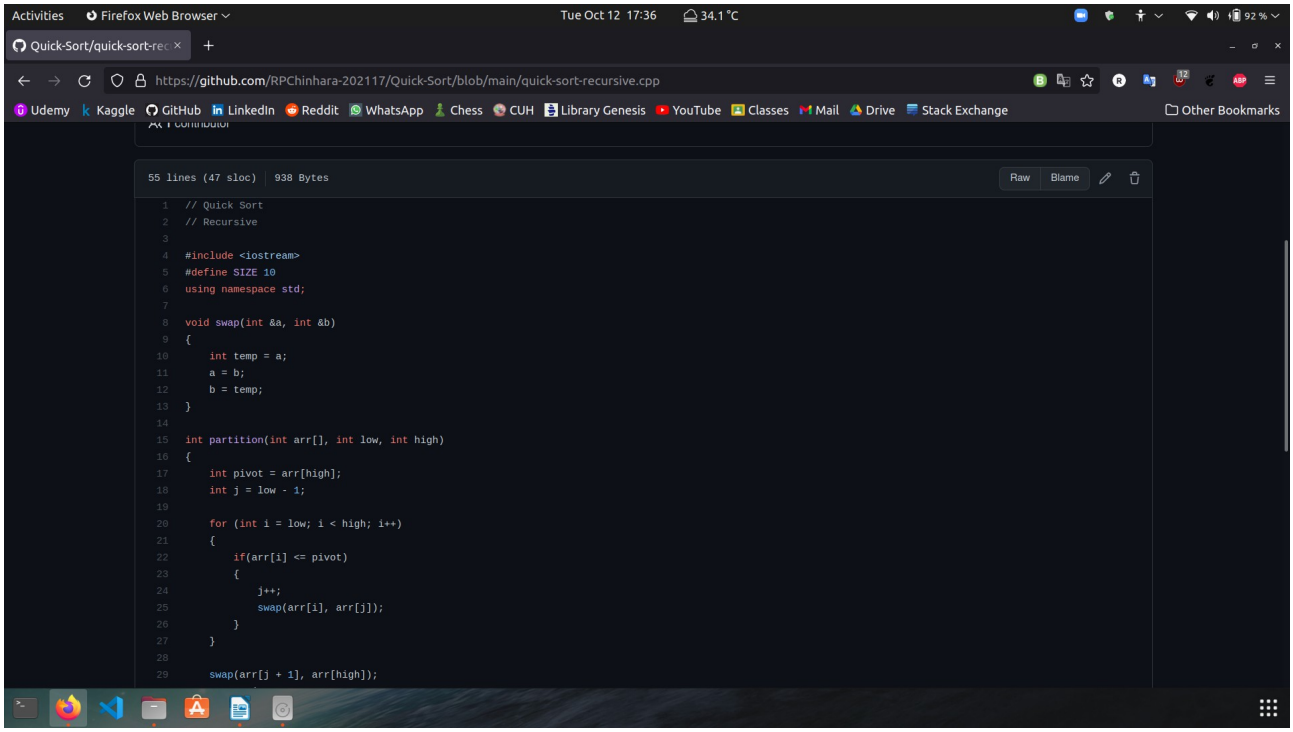
Iterative:

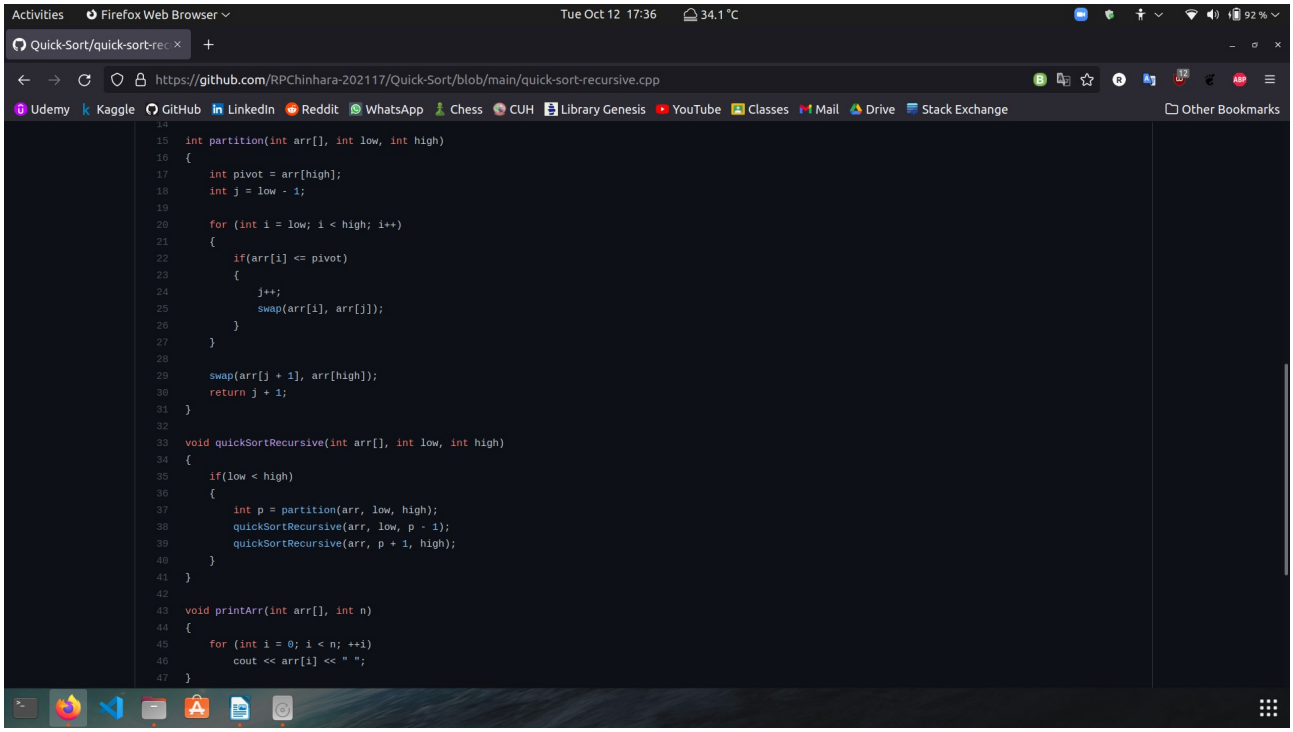

Activities
Firefox Web Browser
Tue Oct 12 17:35
34.1 °C
Quick-Sort/quick-sort-ite
https://github.com/RPChinhara-202117/Quick-Sort/blob/main/quick-sort-iterative.cpp
Udemy
Kaggle
GitHub
LinkedIn
Reddit
WhatsApp
Chess
CUH
Library Genesis
YouTube
Classes
Mail
Drive
Stack Exchange
Other Bookmarks
75 lines (62 sloc) 1.2 KB
Raw
Blame
// Quick Sort
// Iterative
#include <iostream>
#define SIZE 10
using namespace std;
void swap(int &a, int &b)
{
int temp = a;
a = b;
b = temp;
}
int partition(int arr[], int low, int high)
{
int pivot = arr[high];
int j = low - 1;
for (int i = low; i < high; i++)
{
if(arr[i] <= pivot)
{
j++;
swap(arr[i], arr[j]);
}
}
swap(arr[j + 1], arr[high]);
return j + 1;
}


Activities
Firefox Web Browser
Tue Oct 12 17:35
34.1 °C
Quick-Sort/quick-sort-ite
https://github.com/RPChinhara-202117/Quick-Sort/blob/main/quick-sort-iterative.cpp
Udemy
Kaggle
GitHub
LinkedIn
Reddit
WhatsApp
Chess
CUH
Library Genesis
YouTube
Classes
Mail
Drive
Stack Exchange
Other Bookmarks
void quickSortIterative(int arr[], int n)
{
int stack[n];
int top = -1;
int low = 0, high = n - 1;
stack[++top] = low;
stack[++top] = high;
while(top >= 0)
{
high = stack[top--];
low = stack[top--];
int p = partition(arr, low, high);
if(p - 1 > low)
{
stack[++top] = low;
stack[++top] = p - 1;
}
if(p + 1 < high)
{
stack[++top] = p + 1;
stack[++top] = high;
}
}
}
void printArr(int arr[], int n)
{
for (int i = 0; i < n; ++i)

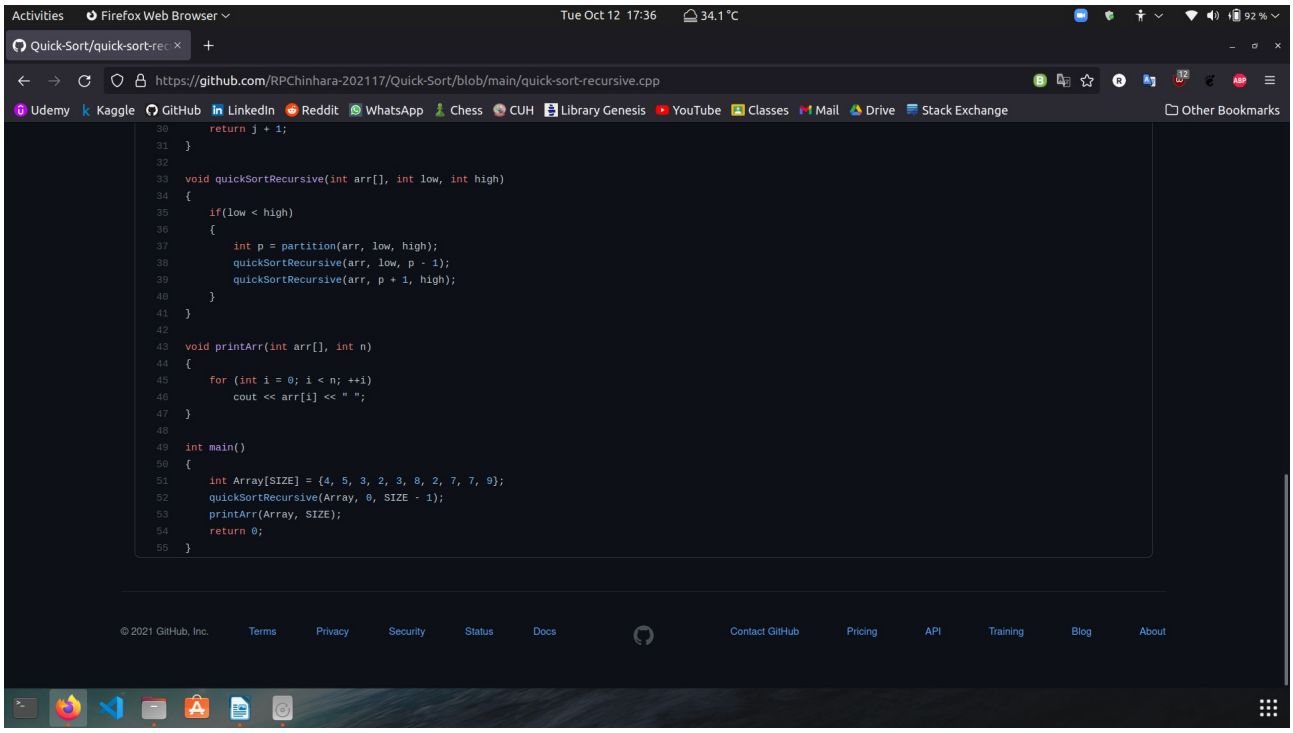

Activities
Firefox Web Browser
Tue Oct 12 17:35
34.1 °C
Quick-Sort/quick-sort-iter
https://github.com/RPChinhara-202117/Quick-Sort/blob/main/quick-sort-iterative.cpp
Udemy
Kaggle
GitHub
LinkedIn
Reddit
WhatsApp
Chess
CUH
Library Genesis
YouTube
Classes
Mail
Drive
Stack Exchange
Other Bookmarks
{
stack[++top] = low;
stack[++top] = p - 1;
}
if(p + 1 < high)
{
stack[++top] = p + 1;
stack[++top] = high;
}
}
}
void printArr(int arr[], int n)
{
for (int i = 0; i < n; ++i)
cout << arr[i] << " ";
}
int main()
{
int arr[SIZE] = {4, 5, 3, 2, 3, 8, 2, 7, 7, 9};
quickSortIterative(arr, SIZE);
printArr(arr, SIZE);
return 0;
}
© 2021 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

Recursive:


Activities
Firefox Web Browser
Tue Oct 12 17:36
34.1 °C
Quick-Sort/quick-sort-rec
https://github.com/RPChinhara-202117/Quick-Sort/blob/main/quick-sort-recursive.cpp
Udemy
Kaggle
GitHub
LinkedIn
Reddit
WhatsApp
Chess
CUH
Library Genesis
YouTube
Classes
Mail
Drive
Stack Exchange
Other Bookmarks
55 lines (47 sloc)
938 Bytes
Raw
Blame
// Quick Sort
// Recursive
#include <iostream>
#define SIZE 10
using namespace std;
void swap(int &a, int &b)
{
int temp = a;
a = b;
b = temp;
}
int partition(int arr[], int low, int high)
{
int pivot = arr[high];
int j = low - 1;
for (int i = low; i < high; i++)
{
if(arr[i] <= pivot)
{
j++;
swap(arr[i], arr[j]);
}
}
swap(arr[j + 1], arr[high]);

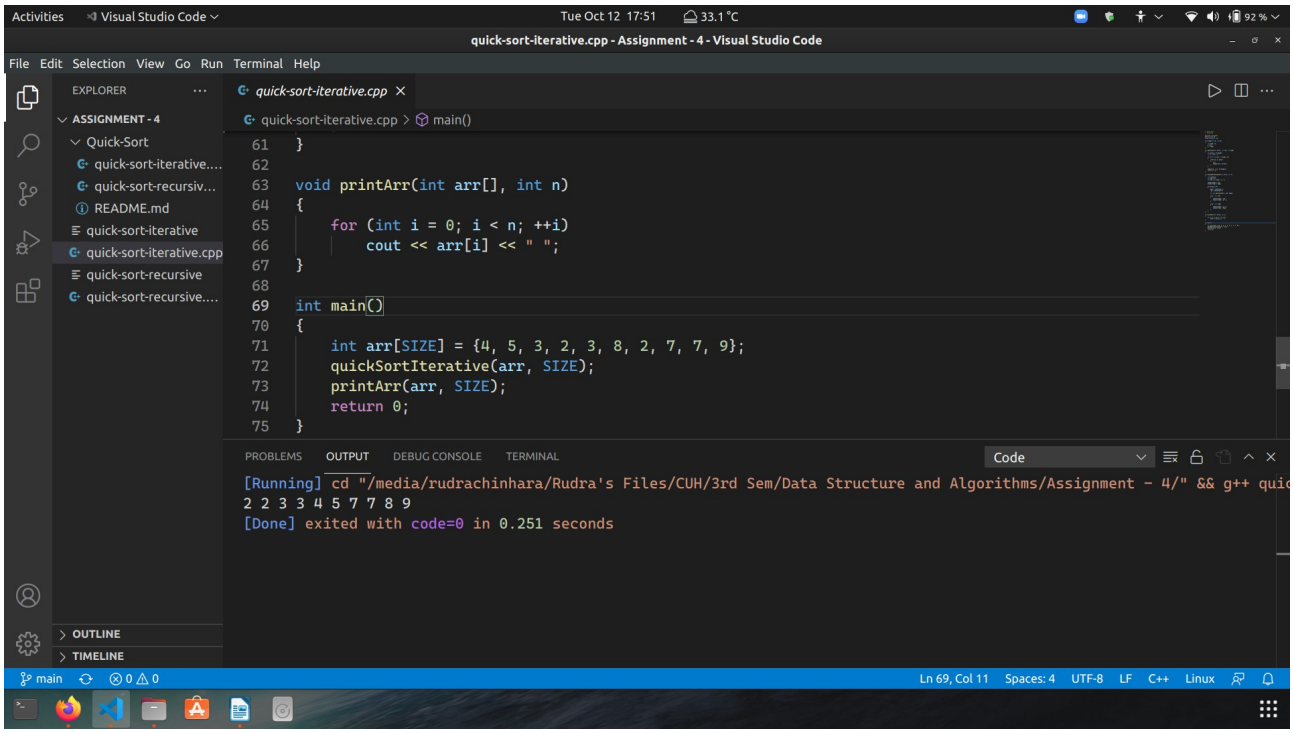


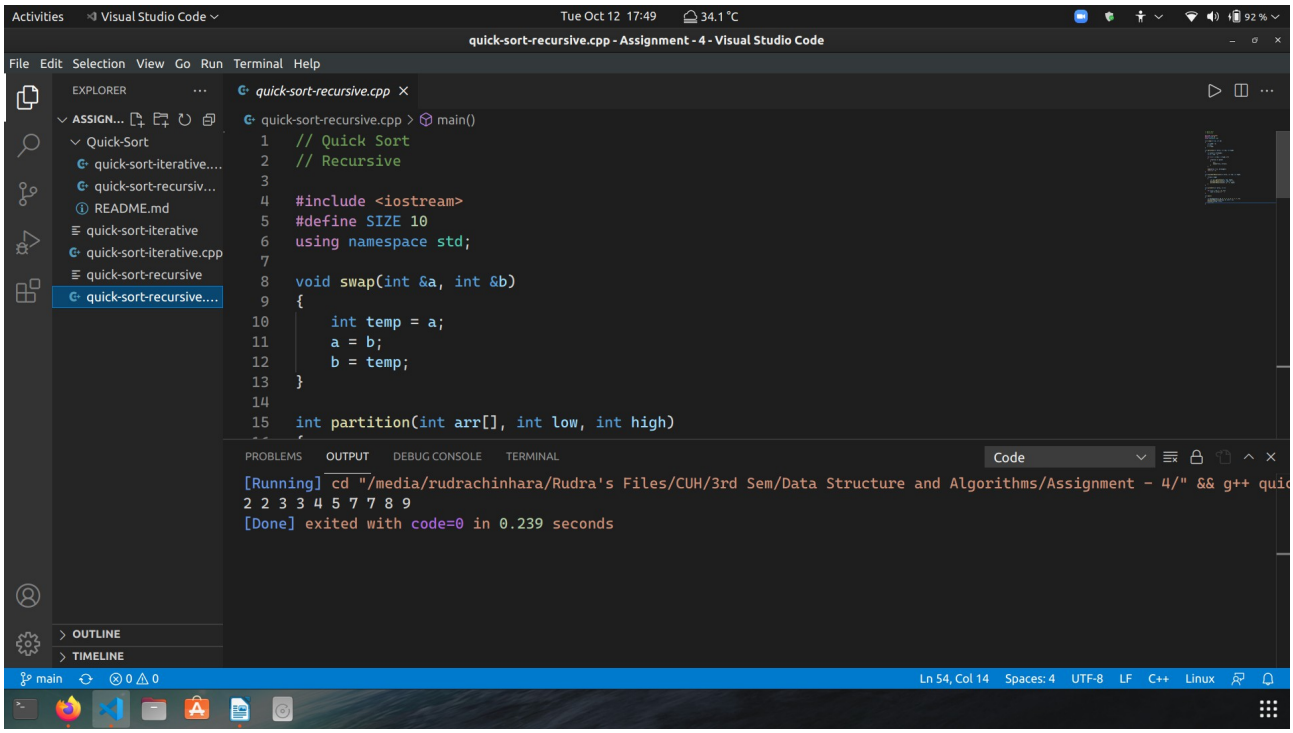
```cpp
int partition(int arr[], int low, int high)
{
    int pivot = arr[high];
    int j = low - 1;

    for (int i = low; i < high; i++)
    {
        if(arr[i] <= pivot)
        {
            j++;
            swap(arr[i], arr[j]);
        }
    }

    swap(arr[j + 1], arr[high]);
    return j + 1;
}

void quickSortRecursive(int arr[], int low, int high)
{
    if(low < high)
    {
        int p = partition(arr, low, high);
        quickSortRecursive(arr, low, p - 1);
        quickSortRecursive(arr, p + 1, high);
    }
}

void printArr(int arr[], int n)
{
    for (int i = 0; i < n; ++i)
        cout << arr[i] << " ";
}
```



```cpp
    return j + 1;
}

void quickSortRecursive(int arr[], int low, int high)
{
    if(low < high)
    {
        int p = partition(arr, low, high);
        quickSortRecursive(arr, low, p - 1);
        quickSortRecursive(arr, p + 1, high);
    }
}

void printArr(int arr[], int n)
{
    for (int i = 0; i < n; ++i)
        cout << arr[i] << " ";
}

int main()
{
    int Array[SIZE] = {4, 5, 3, 2, 3, 8, 2, 7, 7, 9};
    quickSortRecursive(Array, 0, SIZE - 1);
    printArr(Array, SIZE);
    return 0;
}
```

Terminal:


Activities
Visual Studio Code
Tue Oct 12 17:51
33.1 °C
quick-sort-iterative.cpp - Assignment - 4 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ASSIGNMENT - 4
Quick-Sort
quick-sort-iterative....
quick-sort-recursiv...
README.md
quick-sort-iterative
quick-sort-iterative.cpp
quick-sort-recursive
quick-sort-recursive....
quick-sort-iterative.cpp
quick-sort-iterative.cpp > main()
61 }
62
63 void printArr(int arr[], int n)
64 {
65 for (int i = 0; i < n; ++i)
66 cout << arr[i] << " ";
67 }
68
69 int main()
70 {
71 int arr[SIZE] = {4, 5, 3, 2, 3, 8, 2, 7, 7, 9};
72 quickSortIterative(arr, SIZE);
73 printArr(arr, SIZE);
74 return 0;
75 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
[Running] cd "/media/rudrachinhara/Rudra's Files/CUH/3rd Sem/Data Structure and Algorithms/Assignment - 4/" && g++ quic
2 2 3 3 4 5 7 7 8 9
[Done] exited with code=0 in 0.251 seconds
OUTLINE
TIMELINE
main
Ln 69, Col 11 Spaces: 4 UTF-8 LF C++ Linux


Activities
Visual Studio Code
Tue Oct 12 17:49
34.1 °C
quick-sort-recursive.cpp - Assignment - 4 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
Quick-Sort
quick-sort-iterative....
quick-sort-recursiv...
README.md
quick-sort-iterative
quick-sort-iterative.cpp
quick-sort-recursive
quick-sort-recursive....
quick-sort-recursive.cpp
quick-sort-recursive.cpp > main()
1 // Quick Sort
2 // Recursive
3
4 #include <iostream>
5 #define SIZE 10
6 using namespace std;
7
8 void swap(int &a, int &b)
9 {
10 int temp = a;
11 a = b;
12 b = temp;
13 }
14
15 int partition(int arr[], int low, int high)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
[Running] cd "/media/rudrachinhara/Rudra's Files/CUH/3rd Sem/Data Structure and Algorithms/Assignment - 4/" && g++ quic
2 2 3 3 4 5 7 7 8 9
[Done] exited with code=0 in 0.239 seconds
OUTLINE
TIMELINE
main
Ln 54, Col 14 Spaces: 4 UTF-8 LF C++ Linux