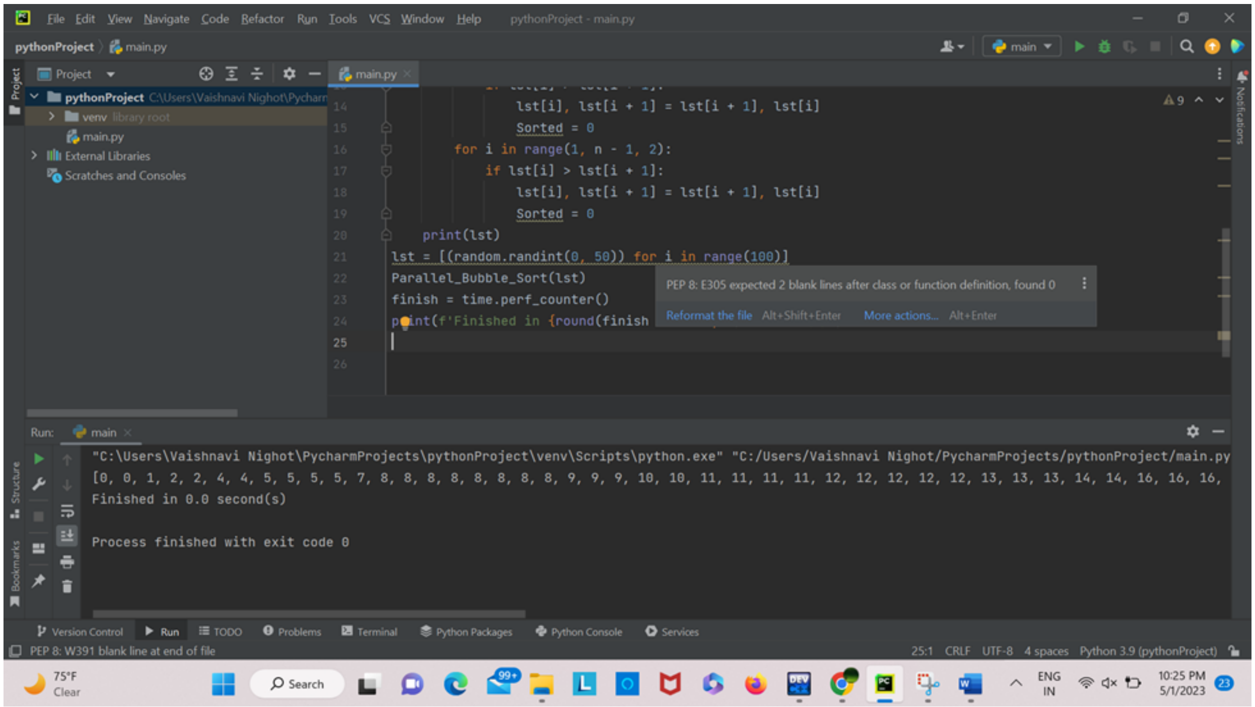
Assignment No. 2

Bubble Sort

Code -

```
import time
import random

#timer to keep track of performance
start = time.perf_counter()

# a function to implement bubble sort in parallel
def Parallel_bubble_sort(lst):
  # variable to keep track of swaps to end the while loop
      Sorted = 0

  # variable to get length of list
      n = len(lst)

  #loop to traverse all list elements in phases
      while Sorted == 0:

    # set to 1 initially to assume list is sorted
    # and no swaps occurred
            Sorted = 1

    # traverse all list elements in pair
    # start at index 0 for odd phase
    # start at index 1 for even phase
            for i in range(0, n-1, 2):
      # check if current element greater than next element
                  if lst[i] > lst[i+1]:
        # if so, swap the elements
                        lst[i], lst[i+1] = lst[i+1], lst[i]

        # set to 0 to imply a swap occurred
                        Sorted = 0
            for i in range(1, n-1, 2):
                  if lst[i] > lst[i+1]:
                        lst[i], lst[i+1] = lst[i+1], lst[i]
                        Sorted = 0
        # print final sorted list
        print(lst)

# an example list to test above program
lst = [(random.randint(0,100)) for i in range(100)]

Parallel_Bubble_Sort(lst)

finish = time.perf_counter()

print(f'Finished in {round(finish-start,2)} second(s)')
```

Output -



File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - main.py
pythonProject main.py
Project
pythonProject C:\Users\Vaishnavi Nighot\Pycharm
venv library root
main.py
External Libraries
Scratches and Consoles
main.py
main

```
                lst[i], lst[i + 1] = lst[i + 1], lst[i]
                Sorted = 0
        for i in range(1, n - 1, 2):
            if lst[i] > lst[i + 1]:
                lst[i], lst[i + 1] = lst[i + 1], lst[i]
                Sorted = 0
    print(lst)
lst = [(random.randint(0, 50)) for i in range(100)]
Parallel_Bubble_Sort(lst)
finish = time.perf_counter()
print(f'Finished in {round(finish
```

PEP 8: E305 expected 2 blank lines after class or function definition, found 0

Reformat the file Alt+Shift+Enter More actions... Alt+Enter
Run: main

```
"C:\Users\Vaishnavi Nighot\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:/Users/Vaishnavi Nighot/PycharmProjects/pythonProject/main.py
[0, 0, 1, 2, 2, 4, 4, 5, 5, 5, 5, 7, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 12, 13, 13, 13, 14, 14, 16, 16, 16,
Finished in 0.0 second(s)

Process finished with exit code 0
```

PEP 8: W391 blank line at end of file

Version Control Run TODO Problems Terminal Python Packages Python Console Services
25:1 CRLF UTF-8 4 spaces Python 3.9 (pythonProject)
75°F Clear
Search
ENG IN
10:25 PM 5/1/2023

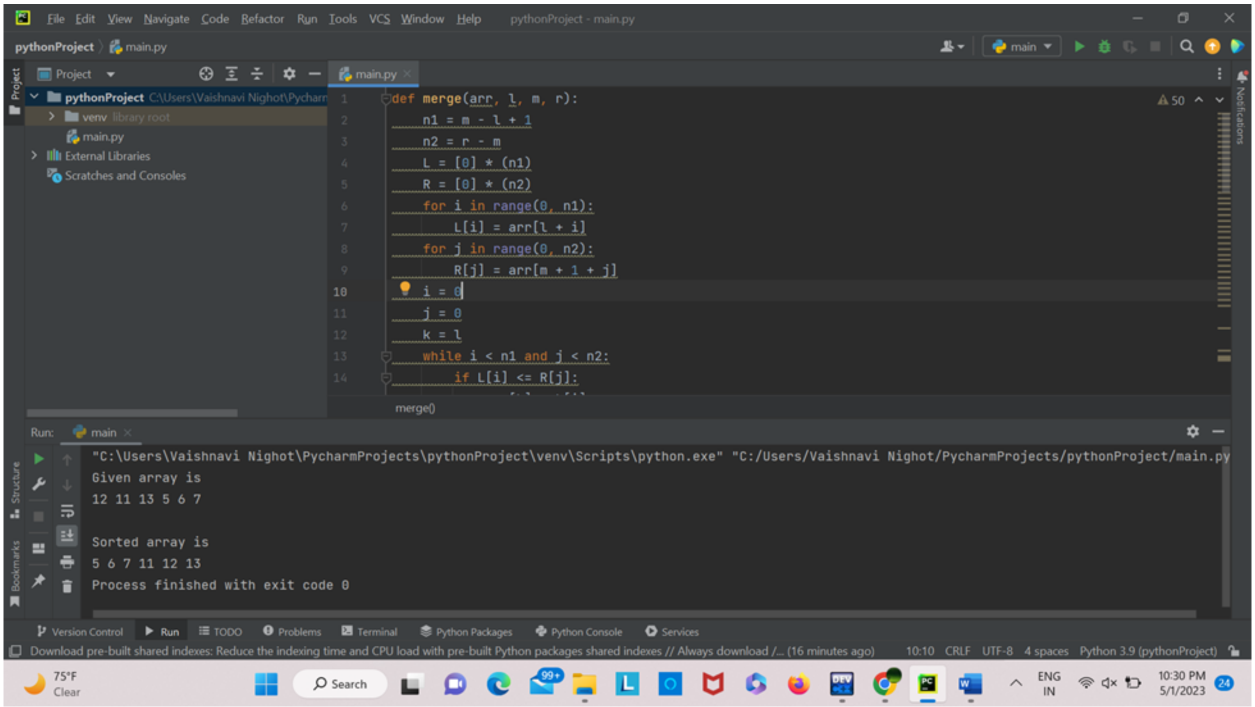
Merge Sort

Code -

```
def merge(arr, l, m, r):
    n1 = m - l + 1
    n2 = r - m

    # create temp arrays
    L = [0] * (n1)
    R = [0] * (n2)

    # Copy data to temp arrays L[] and R[]
    for i in range(0, n1):
        L[i] = arr[l + i]

    for j in range(0, n2):
        R[j] = arr[m + 1 + j]

    # Merge the temp arrays back into arr[l..r]
    i = 0     # Initial index of first subarray
    j = 0     # Initial index of second subarray
    k = l     # Initial index of merged subarray

    while i < n1 and j < n2:
        if L[i] <= R[j]:
            arr[k] = L[i]
            i += 1
        else:
            arr[k] = R[j]
            j += 1
        k += 1

    # Copy the remaining elements of L[], if there
    # are any
    while i < n1:
        arr[k] = L[i]
        i += 1
        k += 1

    # Copy the remaining elements of R[], if there
    # are any
    while j < n2:
        arr[k] = R[j]
```

```
                j += 1
                k += 1

# l is for left index and r is right index of the
# sub-array of arr to be sorted


def mergeSort(arr, l, r):
        if l < r:

                # Same as (l+r)//2, but avoids overflow for
                # large l and h
                m = l+(r-l)//2

                # Sort first and second halves
                mergeSort(arr, l, m)
                mergeSort(arr, m+1, r)
                merge(arr, l, m, r)


# Driver code to test above
arr = [12, 11, 13, 5, 6, 7]
n = len(arr)
print("Given array is")
for i in range(n):
        print("%d" % arr[i],end=" ")

mergeSort(arr, 0, n-1)
print("\n\nSorted array is")
for i in range(n):
        print("%d" % arr[i],end=" ")
```

Output -



```python
def merge(arr, l, m, r):
    n1 = m - l + 1
    n2 = r - m
    L = [0] * (n1)
    R = [0] * (n2)
    for i in range(0, n1):
        L[i] = arr[l + i]
    for j in range(0, n2):
        R[j] = arr[m + 1 + j]
    i = 0
    j = 0
    k = l
    while i < n1 and j < n2:
        if L[i] <= R[j]:
```

```
"C:\Users\Vaishnavi Nighot\PycharmProjects\pythonProject\venv\Scripts\python.exe" "C:/Users/Vaishnavi Nighot/PycharmProjects/pythonProject/main.py
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13
Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download /... (16 minutes ago)