

Problem 6: Given a [Linked List](#) of size N, where every node represents a sub-linked-list and contains two pointers:

(i) a next pointer to the next node,

(ii) a bottom pointer to a linked list where this node is head.

class Node:

```
def __init__(self, data):
```

```
    self.data = data
```

```
    self.next = None
```

```
    self.bottom = None
```

```
def merge_lists(list1, list2):
```

```
    if not list1:
```

```
        return list2
```

```
    if not list2:
```

```
        return list1
```

```
    merged_list = None
```

```
    if list1.data < list2.data:
```

```
        merged_list = list1
```

```
        merged_list.bottom = merge_lists(list1.bottom, list2)
    else:
        merged_list = list2
        merged_list.bottom = merge_lists(list1, list2.bottom)

    merged_list.next = None
    return merged_list
```

```
def flatten_linked_list(head):
    if not head or not head.next:
        return head

    # Merge the first two lists
    head.next = merge_lists(head, head.next)

    # Flatten the remaining lists
    return flatten_linked_list(head.next)
```

```
def create_linked_list(arr, size_arr):
    head = None
    curr_node = None

    list_index = 0
    node_index = 0

    while list_index < len(size_arr):
        for _ in range(size_arr[list_index]):
            new_node = Node(arr[node_index])

            if not head:
                head = new_node
```

```
        curr_node = new_node
    else:
        curr_node.bottom = new_node
        curr_node = new_node
```

```
    node_index += 1
```

```
    list_index += 1
```

```
    return head
```

```
def print_linked_list(head):
```

```
    while head:
        print(head.data, end=" ")
        head = head.bottom
    print()
```

```
if __name__ == "__main__":
```

```
    arr = [5, 7, 8, 30, 10, 20, 19, 22, 50, 28, 35, 40, 45]
```

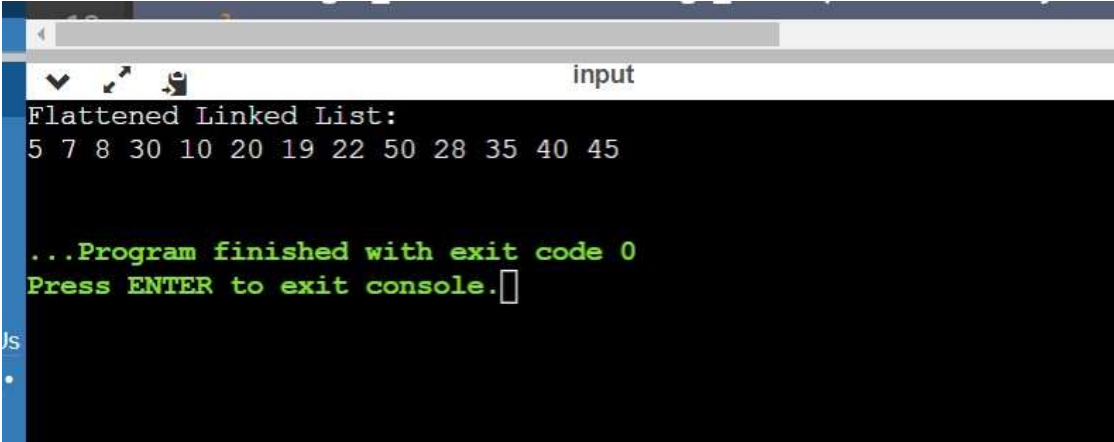
```
    size_arr = [4, 2, 3, 4]
```

```
    head = create_linked_list(arr, size_arr)
```

```
    print("Flattened Linked List:")
```

```
    flattened_head = flatten_linked_list(head)
```

```
print_linked_list(flattened_head)
```



```
input
Flattened Linked List:
5 7 8 30 10 20 19 22 50 28 35 40 45

...Program finished with exit code 0
Press ENTER to exit console.
Js
•
```