190905514 C SECTION 4th SEMESTER MOHAMMAD TOFIK

LAB 8 :

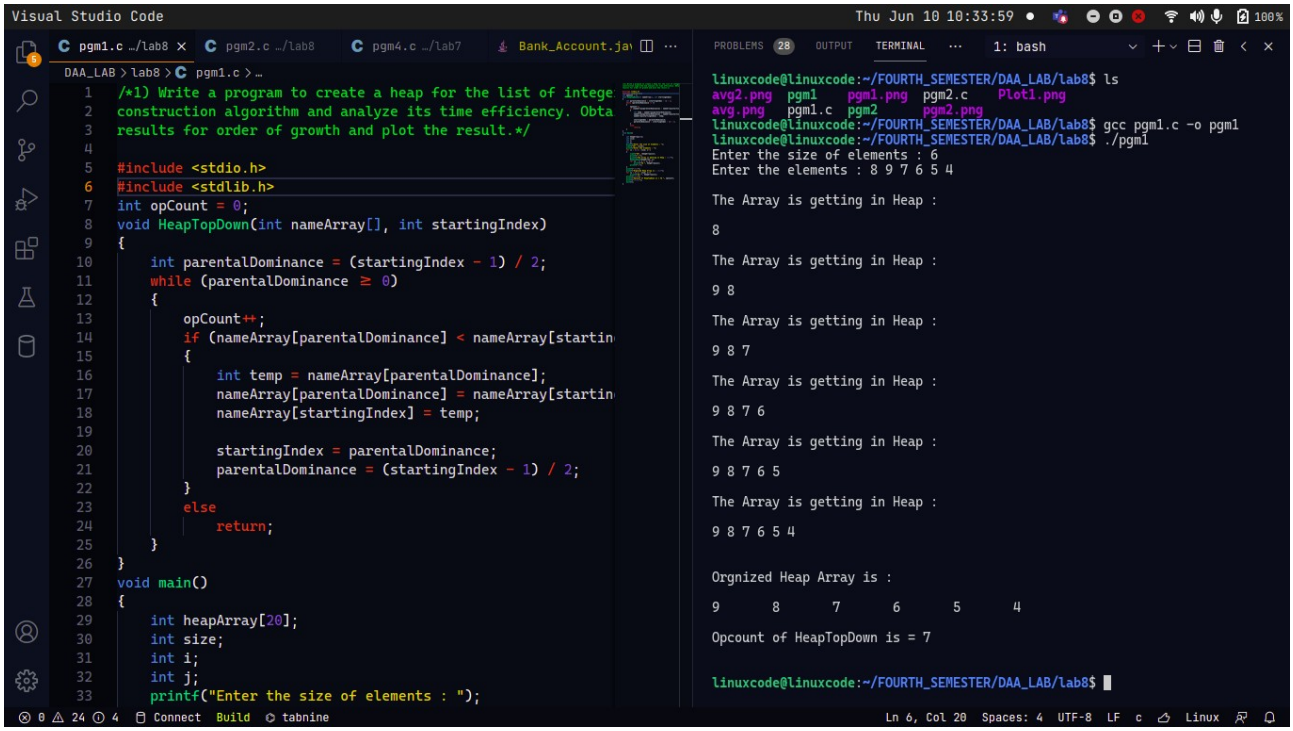
LAB EXERCISES :

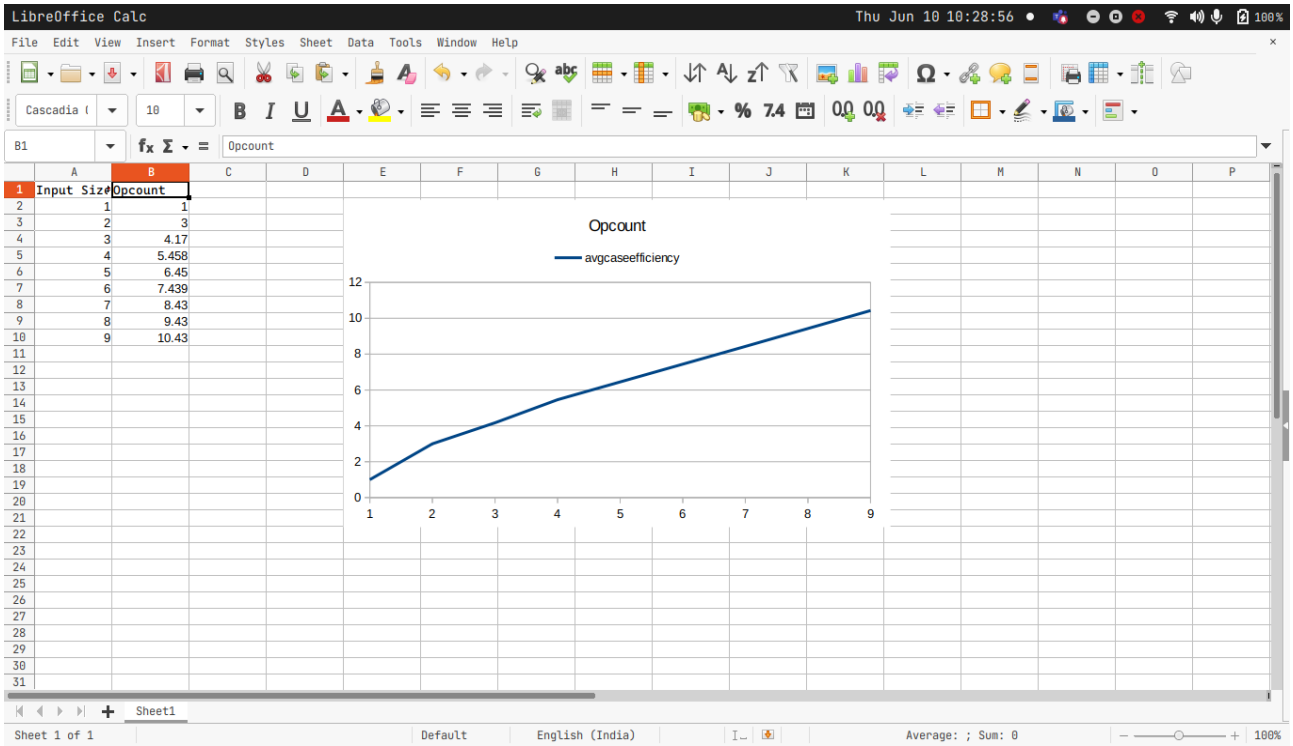
/*1) Write a program to create a heap for the list of integers using top-down heap construction algorithm and analyze its time efficiency. Obtain the experimental results for order of growth and plot the result.*/

```
#include <stdio.h>
#include <stdlib.h>
int opCount = 0;
void HeapTopDown(int nameArray[], int startingIndex)
{
int parentalDominance = (startingIndex - 1) / 2;
while (parentalDominance ≥ 0)
{
opCount++;
if (nameArray[parentalDominance] < nameArray[startingIndex])
{
int temp = nameArray[parentalDominance];
nameArray[parentalDominance] = nameArray[startingIndex];
nameArray[startingIndex] = temp;
startingIndex = parentalDominance;
parentalDominance = (startingIndex - 1) / 2;
}
else
return;}}
void main()
{
int heapArray[20];
int size;
int i;
int j;
printf("Enter the size of elements : ");
scanf("%d", &size);
printf("Enter the elements : ");
for (i = 0; i < size; i++)
{
scanf("%d", &heapArray[i]);
printf("\n");
printf("The Array is getting in Heap : \n\n");
HeapTopDown(heapArray, i);
for (j = 0; j ≤ i; j++)
printf("%d ", heapArray[j]);
printf("\n");}
```

```
printf("\n\n");
printf("Orgnized Heap Array is : \n\n");
for (i = 0; i < size; i++)
printf("%d\t", heapArray[i]);
printf("\n\n");
printf("Opcount of HeapTopDown is = %d ", opCount);
printf("\n\n\n");
exit(0);}
```

OUTPUT :



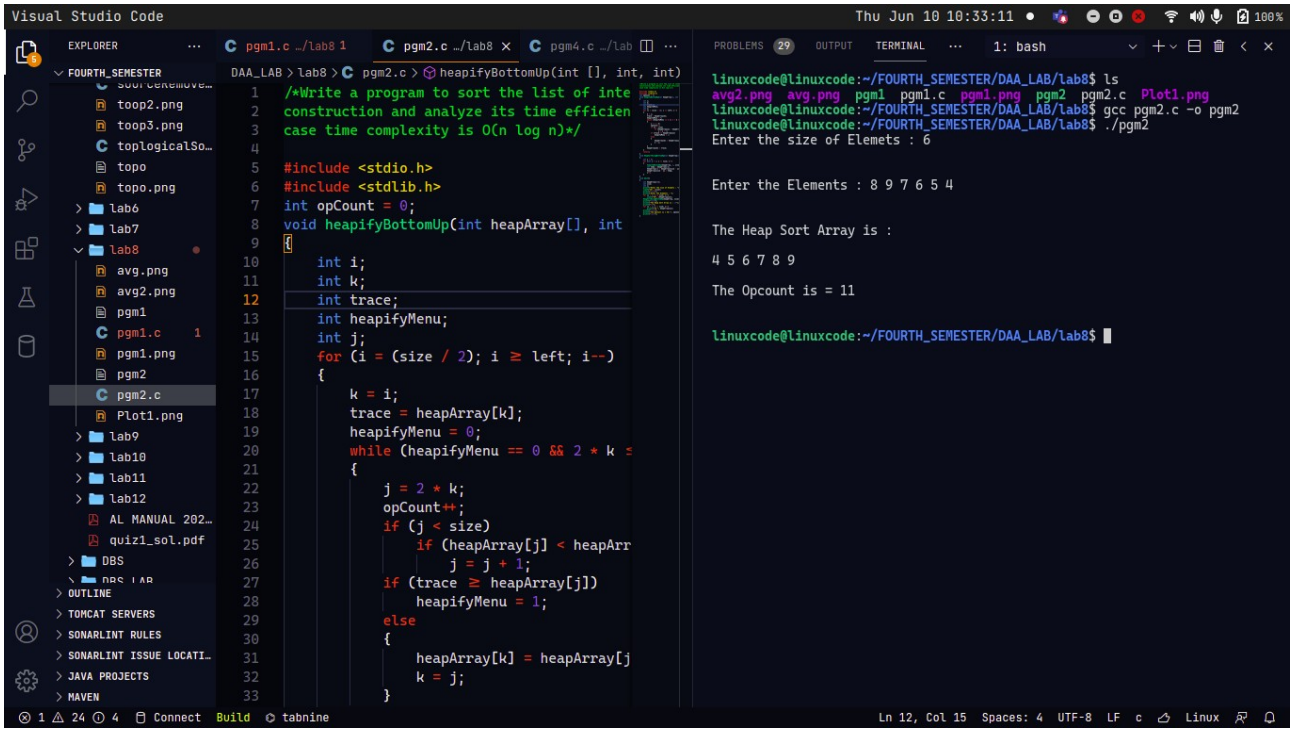


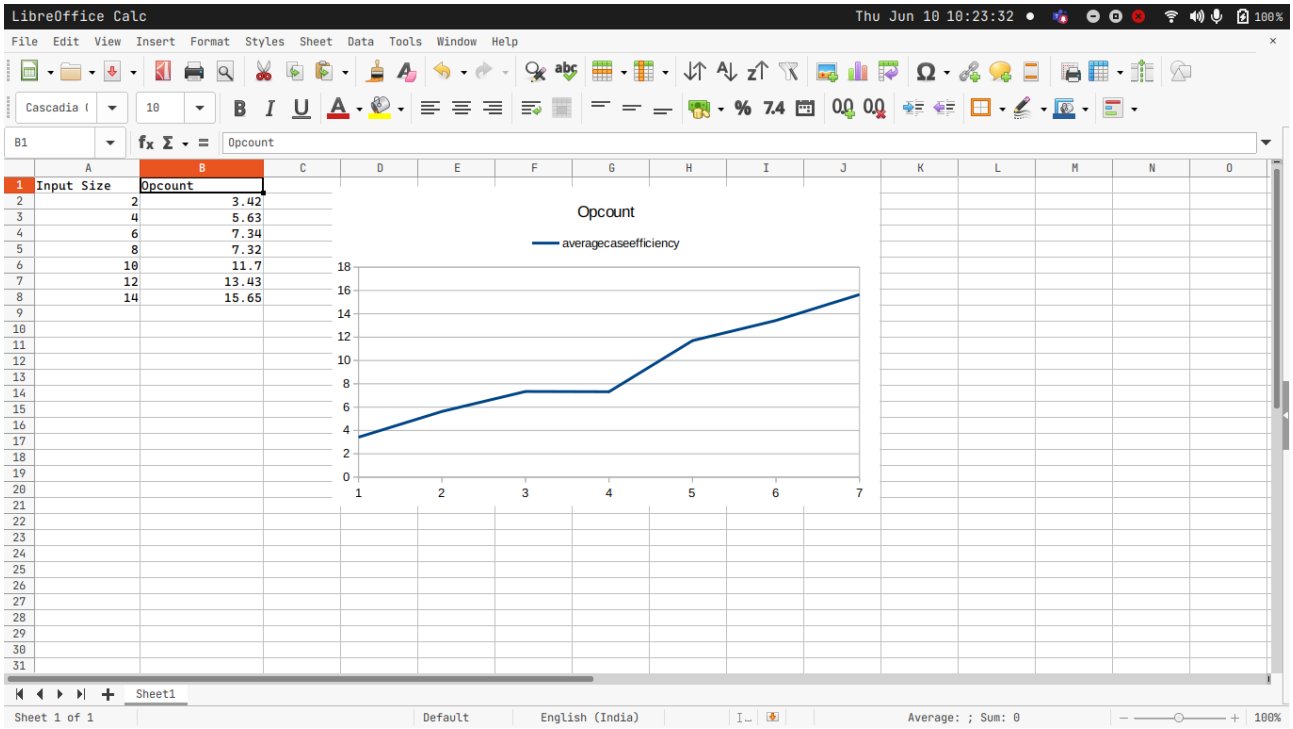
```c
/*Write a program to sort the list of integers using heap sort with
bottom up max heap construction and analyze its time efficiency. Prove
experimentally that the worst case time complexity is O(n log n)*/

#include <stdio.h>
#include <stdlib.h>
int opCount = 0;
void heapifyBottomUp(int heapArray[], int left, int size)
{
int i;
int k;
int trace;
int heapifyMenu;
int j;
for (i = (size / 2); i ≥ left; i--)
{
k = i;
trace = heapArray[k];
heapifyMenu = 0;
while (heapifyMenu == 0 && 2 * k ≤ size)
{
j = 2 * k;
opCount++;
if (j < size)
if (heapArray[j] < heapArray[j + 1])
j = j + 1;
if (trace ≥ heapArray[j])
heapifyMenu = 1;
else
{
heapArray[k] = heapArray[j];
k = j;
}
}
heapArray[k] = trace;
}
return;
}
void HeapSortUsingBottomUp(int heapArray[], int size)
{
int j = 0;
for (int i = 1; i ≤ size; i++)
{
heapifyBottomUp(heapArray, 1, size - j);
int temp = heapArray[1];
```

```
heapArray[1] = heapArray[size - j];
heapArray[size - j] = temp;
j++;
}
}
void main()
{
int heapArray[20];
int size;
int i;
printf("Enter the size of Elemets : ");
scanf("%d", &size);
printf("\n\n");
printf("Enter the Elements : ");
for (i = 1; i ≤ size; i++)
scanf("%d", &heapArray[i]);
HeapSortUsingBottomUp(heapArray, size);
printf("\n\n");
printf("The Heap Sort Array is : \n");
printf("\n");
for (i = 1; i ≤ size; i++)
printf("%d ", heapArray[i]);
printf("\n\n");
printf("The Opcount is = %d\n", opCount);
printf("\n\n");
}
```

OUTPUT :




LibreOffice Calc
Thu Jun 10 10:23:32
File Edit View Insert Format Styles Sheet Data Tools Window Help
Cascadia (
10
B1
Opcount
Input Size
Opcount
2 3.42
4 5.63
6 7.34
8 7.32
10 11.7
12 13.43
14 15.65
Opcount
averagecaseefficiency
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
Sheet1
Sheet 1 of 1
Default
English (India)
Average: ; Sum: 0
100%