Enter the number of elements : 6

Enter ~~for~~ elements of an array :

10 7 6 12 2 1

Elements of an array before sorting :

10 7 6 12 2 1

Elements of an array after sorting :

1 2 6 7 10 12

POORNIMA

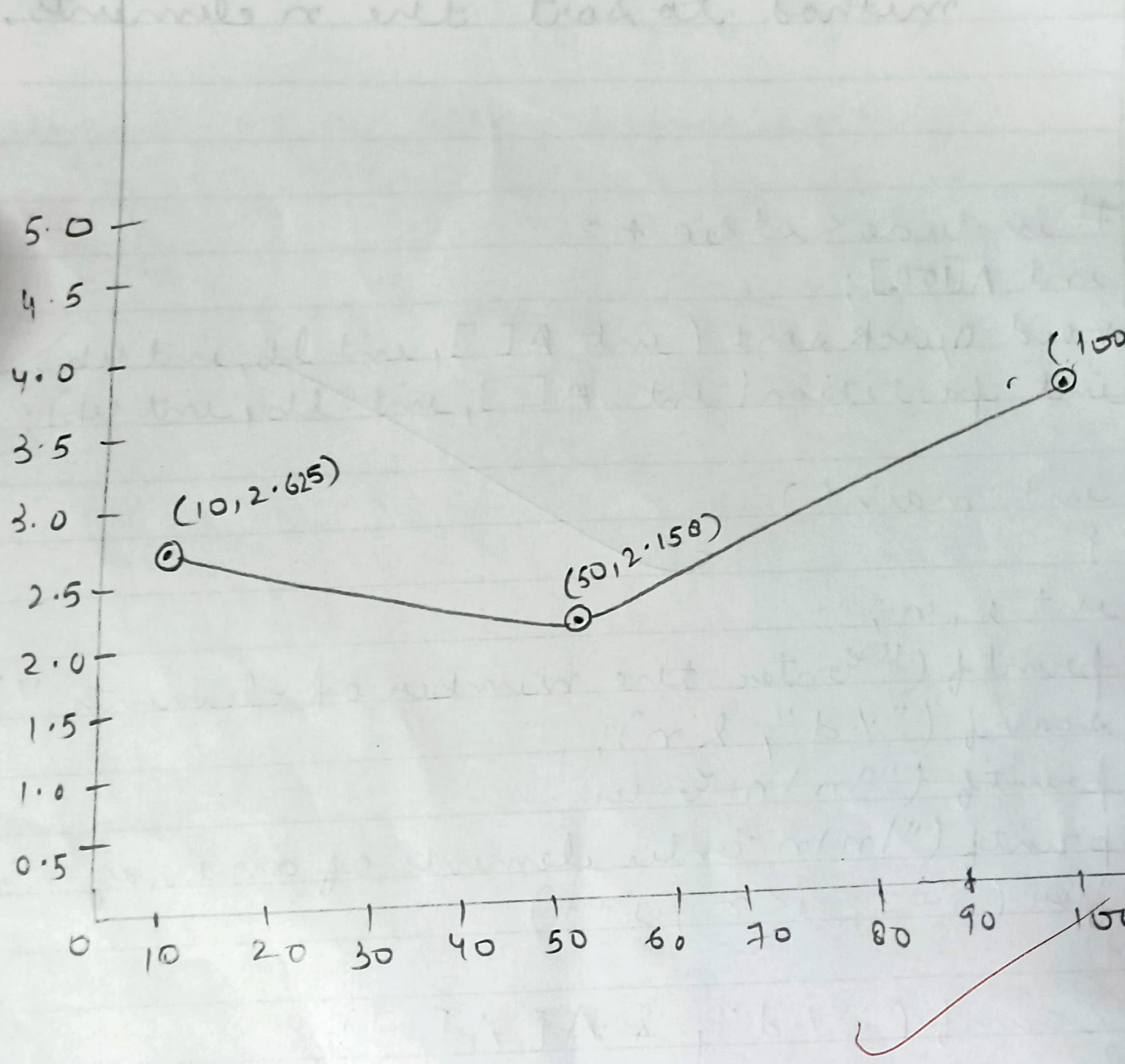
Experiment-1

Objective - WAP to perform Quick sort method to sort the n elements.

```
#include<stdio.h>
int A[100];
void quicksort(int A[], int lb, int ub);
int partition(int A[], int lb, int ub);

int main()
{
int i, n;
printf("Enter the number of elements:");
scanf("%d", &n);
printf("\n\n Enter elements of an array:");
for(i=0; i<n; i++)
{
scanf("%d", &A[i]);
}

printf("\n\n elements of an array before sorting:");
```


5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
10
20
30
40
50
60
70
80
90
(10, 2.625)
(50, 2.150)

```
for (i=0; i<n; i++)
 {
  printf ("%d\t", A[i]);
}
int partition(int A[], int lb, int ub)
{
int pivot, start, end, temp;
pivot = A[lb];
start = lb;
end = ub;
while ( start < end)
 {
while ( A[start] <= pivot)
 {
  start++;
}

while ( A[end] > pivot)
 {
 end--;
}
if ( start < end)
{
 temp = A[start];
A[start] = A[end];
A[end] = temp;
```

```
}
}
temp = A[lb];
A[lb] = A[end];
A[end] = temp;
return end; }

quicksort ( A, 0, n-1);
printf ("\n\n Elements of an array after
sorting :");
for (i=0; i<n; i++)
{
printf ("%d \t", A[i]);
}
void quicksort (int A[], int lb, int ub)
{
int loc;
if ( lb < ub)
{
loc. = partition (A, lb, ub)
quicksort (A, lb, loc-1)
quicksort (A, loc++, ub);
}
}
void quicksort (int A[], int lb, int ub)
{
int loc;
if ( lb < ub)
```

POORNIMA

```
{
  loc = partition (A, lb, ub);
quicksort (A, lb, loc-1);
quicksort (A, loc+1, ub);
}
}
```