Practical -4

## Aim:-

**Implement program for randomized version of quick sort**

# *Randomized Quick Sort:*

## CODE:

#include<stdio.h> #include<math.h>

void quicksort(int \*ar,int start,int end);

int divide(int \*ar,int start,int end,int pivot);

int main(){

int a[] = {33,24,90,39,16};

int n = sizeof(a) / sizeof(a[0]);

quicksort(a,0,n-1); for(int i=0;i<n;i++)

printf(" %d ",a[i]);

}

void quicksort(int \*ar,int start,int end){ if(start < end) {

int pivot = start;

pivot = rand() % (end - start) + start ; pivot = divide(ar,start,end,pivot); quicksort(ar,start,pivot-1);

quicksort (ar,pivot+1,end);

}

}

int divide(int \*ar,int start,int end,int pivot){ int temp,next,next1;

temp =ar[pivot]; ar[pivot]=ar[start]; ar[start]=temp; pivot =start;

next =start+1; while(next<=end){

if(ar[next]>ar[pivot]){ temp = ar[next];

next1 = next;

while(next1!=pivot+1){ ar[next1]=ar[next1-1]; next1--;

}

ar[next1]=ar[pivot]; ar[pivot]=temp; pivot++;

}

next++;

}

return pivot;

}

## OUTPUT:

|  |  |
| --- | --- |
| No | Randomized |
| 10 | 0.005559 |
| 20 | 0.006433 |
| 30 | 0.007000 |
| 40 | 0.007199 |
| 50 | 0.007601 |