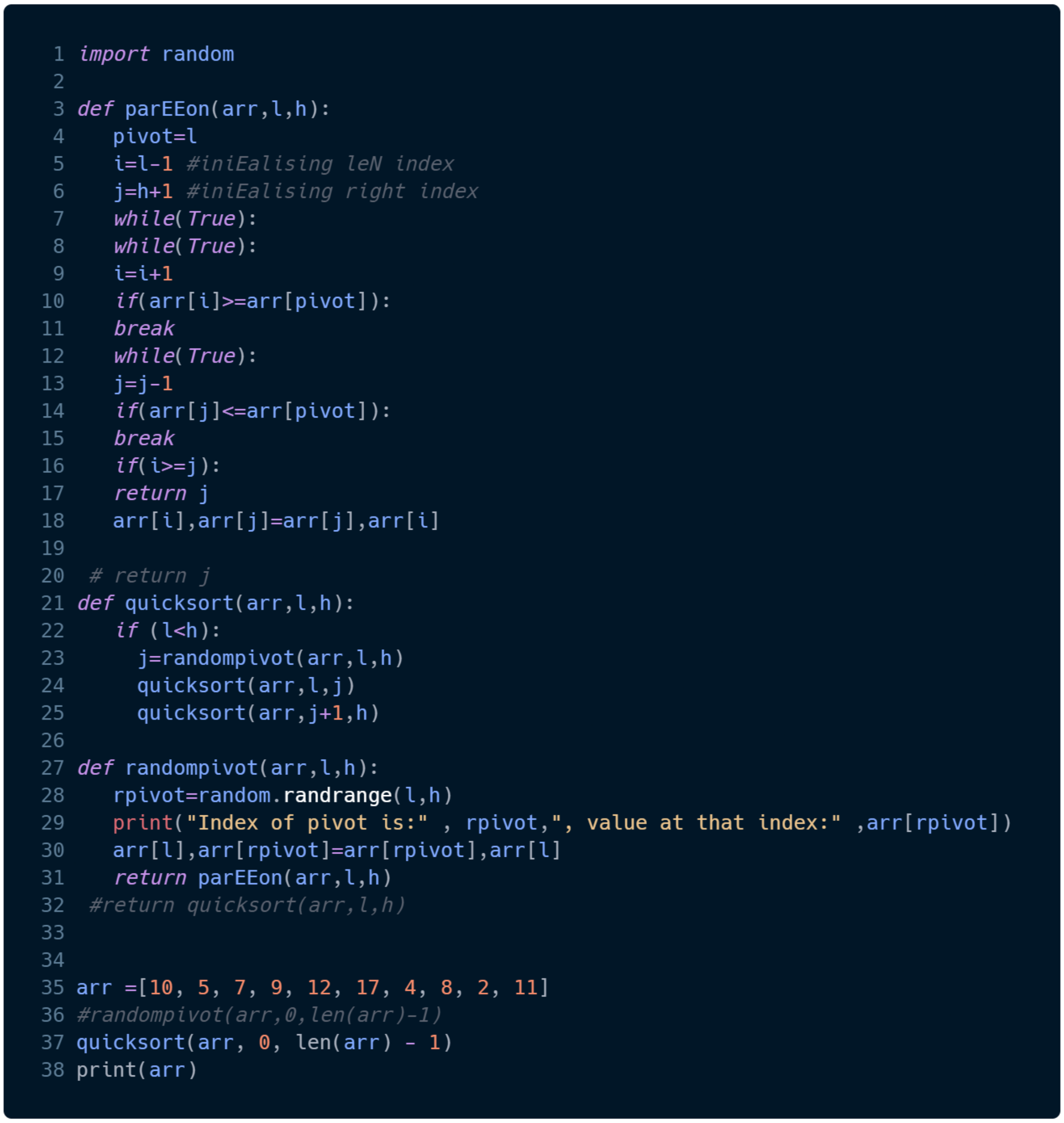
LAB 12

Randomized quick sort is an extension of quicksort in which the pivot element is chosen randomly.



```
import random

def parEEon(arr,l,h):
    pivot=l
    i=l-1 #iniEalising leN index
    j=h+1 #iniEalising right index
    while( True):
    while( True):
    i=i+1
    if(arr[i]>=arr[pivot]):
    break
    while( True):
    j=j-1
    if(arr[j]<=arr[pivot]):
    break
    if( i>=j):
    return j
    arr[i],arr[j]=arr[j],arr[i]

 # return j
def quicksort(arr,l,h):
    if (l<h):
      j=randompivot(arr,l,h)
      quicksort(arr,l,j)
      quicksort(arr,j+1,h)

def randompivot(arr,l,h):
    rpivot=random.randrange(l,h)
    print("Index of pivot is:" , rpivot,", value at that index:" ,arr[rpivot])
    arr[l],arr[rpivot]=arr[rpivot],arr[l]
    return parEEon(arr,l,h)
 #return quicksort(arr,l,h)


arr =[10, 5, 7, 9, 12, 17, 4, 8, 2, 11]
#randompivot(arr,0,len(arr)-1)
quicksort(arr, 0, len(arr) - 1)
print(arr)
```

Explanation:

In this case, the pivot has been selected by using the "random" function. First, the index position 6 with value 4 has been chosen as the index. So, the first element i.e. 10 is swapped with 4. So the new array now is [4, 5, 7, 9, 12, 17, 10, 8, 2, 11]. Now the value of arr[i] is 4 and arr[j] is 11. Now, as i moves ahead in one iteration, j also moves to 2. Now, 5 and 2 are swapped. Now, the process is continued until j crosses i. In quicksort, the new position of j is swapped with the pivot element. Now, the next pivot element is 7 and the same process is repeated.