Lab 2 Stacks

Function	Big O

int linearSearch(int size,int x)
{
for(int i=0;i<size;i++)
{
  if(A[i]==x)
  {
  return i;
  }
}
return -1;
}
	
	n+1[For loop]
	
	2n[fetching value at A[i], comparison]
	
	1n[return]
	
	1[return]
	
	O(n)

int binarySearch(int size,int x)
{
int beg=0;
int end=size-1;
int mid=((beg+end)/2);
 	
while(beg<=end && A[mid]!=x)
{
  if(x<A[mid])
  {
  end = mid-1;
  }
  else
  {
  beg=mid+1;
  }
 	
  mid=((beg+end)/2);
}
if(beg>end)
{
  return -1;
}
else
{
  return mid;
}
}
	
	2[Declaration, Assignment]
	2[Declaration, Assignment]
	2[Declaration, Assignment]
	
	No of elements in array after every traversal
	n/2, n/4, n/8… $n/2^k$ (this goes for k times)
	Finally $2^k = n$
	$k = \log_2(n)$
	
	1[Comparison]
	
	1[return]
	
	1[return]
	$O(n)=\log_2(n)$