Fri	Le Searches for performed inunsorted list
	frome & first element sequentially starting
	4 On Oln awerging 1120 and
*****	
	Linear (K. Size Key)  Repeat for i = 1 to size
	if K[i] = Key
	1 Prince Exement present
**********	Court = 1
	retion
2 M	ON if (court 1)  print element present
Z 1V1	elle print element present
***********	print not found
	lecursive Lineaur (K, first, last, key) if (first > a lost)
***********	return -1
***************************************	else if K[first] = key
************	else return first
*********	return Recursive-Linear (K, first+1, lost,
Notes	Rey)

Binary Search: > Corted data  It Applies d is a Strategy.  It Divides the away into Sub away based on rid  Value  Search is performed only on part of array  If an an average n/2 comparisons to search an  element.  Example: - search 1  Examp	Ringer Control dotter.
Divides the learny into sub array based on Rid  Value  Ly Search is performed only on part of array  Ly On an average 1/2 comparisons to search an  element  Example: - search 1  15 11 25 86 48 66 75  Bincory Search (K, size, key) Omid = Ott = 6 - 3:5 = 3  left = 0  mid + key Key K [mid]  right = Size - 1  If key = k[mid]  The key = k[	Le Applier du c Strategy 3 TUF
Search is performed only on part of array  Ly an an arrange 1/2 comparisons to search an element Example: Search 1  Bincary Search (K, size, key) Omid = 0+# = 6 - 3.5 = 2  1ett = 0 mid + key Key Key Key Key K Kimid J  right = Size - 1 might = mid - 1 = 2  while left L = right	Le Divides the prescry into sub prescry boused on will
Search is performed only on part of array  Ly on an anexage n/2 comparisons to search an  element Example: Search 1  Binary Search (K, Size, Key) Omid = 0+4 = 6 = 3.5 = 3  left = 0	Value
Recursive Binary (K feft, right, key) (D mid = 3  if left (= right) (2) mid = 36 4+7=1-5  mid = (left + right) /2  if key = K [mid]  return mid !  present  Clse if Key > K [mid]  Notes  Notes  Notes  Recursive Binary (k, mid+1, right, key)  else	Search is performed only on part of array  Ly on an average n/2 comparisons to search an  element Example: Search 1  Binary Search (K, size, key) Omid = 0+# = 6 - 3.5 = 3  left = 0 mid \( \frac{1}{2} \) Key
Recursive Binary (K, feft, right, key) (1) mid = 3  if left L= right (2) mid = 36 4+7=1-5  mid = [left] right) /2 (3) mid = 6+7=13=6  return mid    present (4) mid = 7+7=14=7  else if Key > K[mid]  return Recursive, Binary (k, mid+1, right, key)  else	ALIOT VOIL
if left L= right (K, feft, right, key) (D mid = 3  if left L= right (D) mid = 26 4+7=11-5  mid = (left + right) /2 (D) mid = 26 4+7=11-5  if key = K [mid] (D) mid = 6+7=13=6.  return mid 11 present (D) mid = 7+7=14=7.  else if key > K [mid]  Notes  Notes  else	
rebian ready (k, left, mud-11 key).	if left /= right (K, feft, right, key) (1) mid = 3  if left /= right (2) (2) mid = 36 4+7=1-5  mid = (left + right) /2 (3) mid = 6+7=1-5  return mid // present (1) mid = 7+7=14=7.  else if Key > K[mid]  return Recursive pin