

NAME : Badrul Zaman

ID : 8 13685

Submitted to : Sis Zulgarman Sb

Course Title : Data Structure and Algorithm

Q1.

Answer:

Insertion in an Element:

We need to follow the following steps in Insert an Element.

- 1- we will create a new node, with given value.
- 2- Then we will check whether list is empty ($head == null$)
- 3- After that if it is empty then, set $newNode \rightarrow next = null$ and $head = newNode$.
- 4- And at last if it is not empty then, set $newNode \rightarrow next = head$ and $head = newNode$.

Delete in an Element:

For delete an element we need to follow the following steps

- 1- first we will check whether list is Empty ($head == null$)
- 2- Then if it is Empty then, display "List is Empty!!! Deletion is not possible" and terminate the function
- 3- After that if it is not Empty then, define a node pointer 'Temp' and initialize with head.
- 4- Then check whether list is having only one node ($temp \rightarrow next == NULL$)
- 5- If it is TRUE then set $head = NULL$ and delete Temp (Setting Empty list conditions)
- 6- At last if it is FALSE then set $head = temp \rightarrow next$, and delete temp.

Q2:-

Ans:->

Program

```
#include <iostream>
```

```
void main()
```

```
{
```

```
int arr1[100], i, k, p, x;
```

```
printf("\n|k|Insert new value in the unsorted  
array:\n");
```

```
printf("-----\n");
```

```
scanf("%d elements in the
```

```
printf("Input the size of array:");
```

```
scanf("%d", &k);
```

```
/* unsorted values into the array*/
```

```
printf("Input %d elements in the array  
in ascending order:\n", n);
```

```
for (i=0; i<k; i++)
```

```
{
```

```
printf("element - %d: ", i);
```

```
scanf("%d", &arr1[i]);
```

```
}
```

```
printf("Input the value to be inserted:");
```

```
scanf("%d", &x);
```

```
printf("Input the value to be inserted:");  
scanf("%d", &(and)x);  
printf("Input the position, where the value  
to inserted:");  
scanf("%d", &P);
```

```
printf("The current list of the array:\n");  
for (i=0; i<k; i++)  
    printf("%5d", arr1[i]);  
/* Move all data at right side of the array */  
for (i=k; i>=P; i--)  
    arr1[i] = arr1[i-1];  
/* insert value at given position */  
arr1[P-1] = x;  
printf("\n\n After insert the element the  
new list is:\n");  
for (i=0; i<=k; i++)  
    printf("%5d", arr1[i]);  
printf("\n\n");  
}
```

Q3:

AnswerQuick Sort:

Quick sort is a sorting algorithm, which is commonly used in computer science. Quick sort is a divide and conquer algorithm. It creates two empty arrays to hold elements less than the pivot value and elements greater than the pivot value, and then recursively sort the sub arrays. There are two basic operations in the algorithm, swapping items in place and partitioning a section of the array.

- 1- Always pick first element as a pivot-
- 2- Always pick last element as a pivot.
- 3- Start a pointer at the first or last item array.
- 4- while the value at the right pointer in the array is less or greater than the pivot value.
while value is less (add 1)
while value is greater (subtract 1)

- 5) If the left pointer is less than equal to the right pointer then swap the values pointer these location in the array.
- 6) Move left pointer to the right by one and the right to the left.
- 7) If the left pointer and the right pointer don't meet go to step 1

Example:

```

var array = [9, 2, 5, 6, 4, 3, 7, 10, 11, 12, 8, 11];
function quick sort (array) {
  if (array.length == 0) return [];
  var left = [], right = [], Pivot = array[0];
  for (var i = 1; i < array.length; i++)
  {
    if (array[i] < Pivot)
      left.push(array[i])
    else
      right.push(array[i]);
  }
  return quick sort (left) concat (Pivot),
    quick sort (right);
}
console.log(quick sort (array, slice));
⇒ [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

```

Algorithm:

- 1- If the head node is null. Then insert the data in the head node.
- 2- else, if the Input data is Less than the start node. Then insert the node at the start node, Then insert the node at the start.
- 3- If the input data is greater than the start node. Till you get the right position to insert.
- 4- If the element lie between any two value, Then ~~connect~~ connect the node to the previous node and the next node. ie, $t \rightarrow \text{next} = \text{Temp} \rightarrow \text{next}$ and $\text{temp} \rightarrow \text{next} = t$.

Q4(a)

Ans:

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	1	0
C	0	1	0	1	1
D	0	1	1	0	0
E	1	0	1	0	0

Q4(b)

Ans:

