Date

Expt. No. 3 Page No. 06

```
printf("\n Enter the value to be inserted : ");
scanf("%d", &num);
printf("\n Enter the position at which value is
        inserted : ");
scanf("%d", &pos);

for (i=n-1 ; i>=pos ; i--){
        arr[i+1] = arr[i];
    }
arr[pos] = num;
arr{ n += 1;
printf("\n The Array after insertion : \n");
for (i=0 ; i<n ; i++)
{
    printf("Array[%d] = %d ", i, arr[i]);
}
return 0;
}
```

Input : Enter the value to be inserted : 15
Enter the position at which value is inserted : 4

Output : Original Array's elements are :
Array[0] = 2
Array[1] = 5
Array[2] = 7
Array[3] = 4
Array[4] = 1

Teacher's Signature :

```
printf("\nEnter the position from which the
        number has to be deleted : ");
scanf("%d", &pos);

for(i=pos; i<n-1; i++)
{
    arr[i] = arr[i+1];
}
n = n-1;
printf("\n The array after deletion is : \n");

for(i=0; i<n; i++)
{
    printf("\n Array[%d] = %d", i, arr[i]);
}

return 0;
}
```

Input: Enter the size of the array : 5
Array [0] = 23
Array [1] = 46
Array [2] = 73
Array [3] = 99
Array [4] = 56

Enter the position from which the number has to be deleted : 3

Teacher's Signature : 02/01/24

Expt. No. 9 Date

Page No. 20

EXPERIMENT-09

Aim: Write a program in C to insert at Specific location.

Algorithm:

Program:

```
#include <stdio.h>
#include <stdlib.h>

Struct Node {
    int data;
    Struct Node * pnext;
};

Struct Node * InsertAtNode ( Struct Node * head,
                             int data, Struct Node *
                             prevNode )

{
    Struct Node * ptr = (Struct Node*) malloc
                        (sizeof (Struct Node));
```

Teacher's Signature :

```
ptr → data = data;
ptr → next = prevNode → next;
prevNode → next = ptr;

return head;

}

void traverselist (struct Node * head)
{
    while (head != NULL) {

    printf("Element is : %d \n", head → data);
    head = head → next;
    }
}

int main() {
        struct Node * head;
        struct Node * second;
        struct Node * third;
        struct Node * forth;

head = (struct Node *) malloc (sizeof (struct Node));
second = (struct Node*) malloc (size of (struct Node));
third = (struct Node*) malloc (size of (struct Node));
forth = (struct Node* ) malloc (size of (struct Node));
```

Teacher's Signature :

Expt. No. 10 Page No. 24 Date

EXPERIMENT-10

Aim: Write a program in C to reverse each item iteratively.

Program:

```
#include<stdio.h>
#include<stdlib.h>

struct Node {
        int data;
        struct Node * next;
};

Struct Node* reverseList(struct Node * head){

    Struct Node* prev = NULL;
    Struct Node* current = head;
    Struct Node* next = NULL;

    while (current != NULL){
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
```

Teacher's Signature :

```
    head = prev;
    return head;
}

void printList(struct Node* node){
    while (node != NULL){
        printf("%d", node->data);
        node = node->next;
    }
}

int main(){
    struct Node *head;
    struct Node * second;
    struct Node * third;

head = (struct Node*) malloc (sizeof (struct Node));
second = (struct Node*) malloc (sizeof (struct Node));
third = (struct Node*) malloc (sizeof (struct Node));
forth = (struct Node*) malloc (sizeof (struct Node));

    head -> data = 15;
    head -> next = second;
    second -> data = 35;
    second -> next = third;
    third -> data = 25;
    third -> next = forth;
```

Teacher's Signature :