DS Lab Notes of Sandeep

To see the other programs

Lab 1: Write a C program to perform the following operations on the given array

(i) Insert element in specific position in to array

(ii) Delete random element from array

(iii) Reverse the array elements

Lab 2: A) Write a C program to implement Single linked list

i) Insertion ii) Deletion iii) Display

B) Write a C program to implement Circular linked list

i) Insertion ii) Deletion. iii)Display

Lab 3: A) Write a C program to implement Doubly linked list

i) Insertion ii) Deletion. iii)Display

B) Write C programs to implement Stack ADT using

i) Array ii) Linked List

Lab 4:

A. Write a C program that uses stack operations to convert a given infix expression in to its postfix equivalent. (Display the role of stack).
B. Write a C program for Evaluation of postfix expression.

.

Lab 5: Write C programs to implement Queue ADT using

i) Array ii) Linked List

Lab 6: Write a C program to implement Binary search tree

i) Insertion ii) deletion iii) Traversals

Lab 7:

Write a C program to implement binary search tree Non - recursively traversals

i) Pre- Order ii) Post –Order iii) In-Order

Lab 8:

(A) Write a C Program to Check if a Given Binary Tree is an AVL Tree or Not

(B) Write a C program to find height of a Binary tree

(C) Write a C program to count the number of leaf nodes in a tree

Lab 9:

Write a C program for implementing Graph traversal
i) DFS ii) BFS

Lab 10:

A) Write a C program to implement different hash methods

B) Write a C program to implement the following collision resolving
i) Quadratic probing. ii) Linear Probing

Lab 11:

Write C programs for implementing the following Sorting methods and display the important steps.
i) Quick Sort ii) Heap sort

Lab 12:

Write a C program for implementing pattern matching algorithms

I. Knuth-Morris-Pratt ii) Brute Force

Additional

A. Implement the priority queue using Heap
B. Write a C Program to Implement Merge sort
C. Write a C program to implement AVL tree
   1. Creation ii) Deletion iii) Traversals
D. Write a function to reverse the nodes of a linked list
E. Write a C program to implement 2-3-4 tree operations
F. Write a C program to implement B tree operations
G. Write a C program to implement B+ tree operations

1.1. Insertion Of Element Int An Array

Date 01/04/2021
Thursday

Aim :

Write a program to inserting an element into a specific position of an array

Algorithm :

```
1. Start
2. Read length of an array ( say l )
3. Create an Array a of size l+1 and i,temp
4. for (i = 0; i < l; i++ )
   4.1. Read element and assign a[ I ]
5. Read  ie and ip
6. for ( i = ip; i < l+1 ; i++)
   6.1. temp = a[ i ]
   6.2. a[ i ] = ie
   6.3. ie = temp
7. print "the array of after replacing"
8. for ( i = 0; i < l+1; i++ )
   8.1. print a[ i ]
9. Stop
```

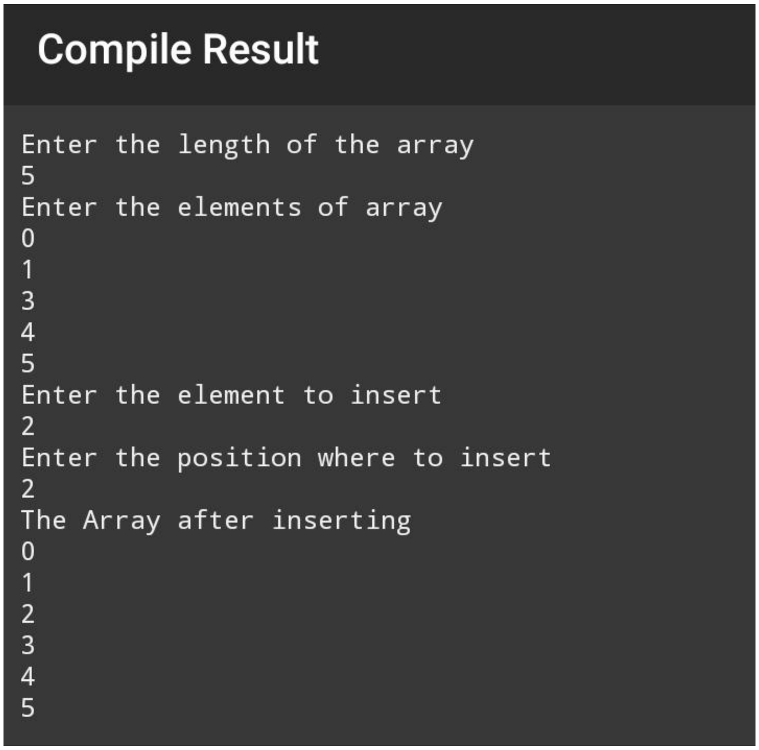
Program :

```
#include<stdio.h>
void main()
{
        int i, l, ie, ip, temp;

        printf("Enter the length of the array\n");
        scanf("%d",&l);
        int a[l+1];
        printf("Enter the elements of array\n");

        for(i=0; i<l; i++)
                scanf("%d",&a[i]);
        printf("Enter the element to insert\n");
        scanf ("%d",&ie);
        printf("Enter the position where to insert\n");
        scanf("%d",&ip);

        for(i = ip; i < l+1; i++)
        {
                temp = a[i];
                a[i] = ie;
                ie = temp;
        }

        printf("The Array after inserting\n");
        for(i=0; i<l+1; i++)
                printf("%d\n",a[i]);
    }
```

Input :

```
    5
    0
    1
    3
    4
    5
    2
    2
```

Output :



```
Enter the length of the array
5
Enter the elements of array
0
1
3
4
5
Enter the element to insert
2
Enter the position where to insert
2
The Array after inserting
0
1
2
3
4
5
```

1.2. Deletion Of Specific Element From The Array

Date : 08/04/2021

Aim :

To write a C program to Delete an element of an Array

Algorithm :

```
1. Start
2. Read length of array as n
3. Read All Elements Of array using loop iterates from 0 to n - 1
4. Read the position of deleting element
5.
   6.1  If deleting position is greater the n-1
           6.1.1. print deletion is not possible
   6.2 else
           6.2.1. Left Shift the elements from index deleting position
6. Print the elements after deletion
7. Stop
```

Program :

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

void main()
{
   int n,dp;
   int *arr;

   printf("Enter the length of the array: ");
   scanf("%d",&n);

   arr = (int*) malloc (n* sizeof(int));

   printf("Enter the elements of the array\n");
   for(int i = 0; i < n; i++)
             scanf("%d",&arr[i]);

   printf("The elements of the array are \n");
   for(int i = 0; i < n; i++)
            printf("%d\n",arr[i]);
```

```
    printf("Enter the position of element want to delete\n");
    scanf ("%d",&dp);

    for(int i = dp; i < n-1; i++)
            arr[i] = arr[i+1];
    n--;
    arr = (int*) realloc(arr, n*sizeof(int));

    printf("After deletion The elements of the array are \n");
    for(int i = 0; i < n; i++)
            printf("%d\n",arr[i]);

}
```

Input :

```
6
1
2
3
13
4
5
2
```

Output :

```
Enter the length of the array: Enter the elements of the array
The elements of the array are
1
2
3
13
4
5
Enter the position of element want to delete
After deletion The elements of the array are
1
2
13
4
```

1.3. Reversing The Elements Of The Array

Aim :

To write a C program To reverse the elements Of The Array

Algorithm :

```
1. Start
2. Read size of array as n
3. Read the elements Of Array using loop iterates from 0 to n-1
4. Loop iterates from I = 0 to n/2
   4.1. Swap arr[I] and arr[n-i-1] to reverse
5. Print The Elements Of the  Array After Reversing
6. Stop
```

Program :

```
#include<stdio.h>
void main()
{
   int n,i;

   printf("Enter the length of the array\n");
   scanf("%d",&n);

   int a[n];
   printf("Enter the elements of array\n");
   for(i=0; i<n; i++)
     scanf("%d",&a[i]);

   for(int i = 0; i < n/2; i++)
   {
      int temp = a[i];
      a[i] = a[n-1 -i];
      a[n-i-1] = temp;
   }

   printf("Array of after Reverse\n");
   for(i = 0;i < n; i++)
      printf("%d\n",a[i]);
}
```

Input :

```
5
1
```

```
        2
        3
        4
        5
```

Output :

```
        Enter the length of the array
        Enter the elements of array
        Array of after Reverse
        5
        4
        3
        2
        1
```

Lab 1: Write a C program to perform the following operations on the given array
Insert element in specific position in to array
Delete random element from array
Reverse the array elements

Date 08/04/2021

Lo

Aim :

Algorithm :

```
1. Start
2. Read n (No. Of elements of the array)
3. Read the elements of the array using loop
4. Select choice
   4.1. insertion / deletion / reversing
5. Select 1 to Continue and 2 to stop
   5.1. If(selection == 1)
        5.1.1. Repeat from 4.
6. Stop
```

Program :

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

int *a,n,i,temp;
void choice();//to select options
void insert();
void deletion ();
void reversion ();
void display();

//Main function
void main()
{
   printf("Enter the length of the array\n");
   scanf("%d",&n);

   a = (int*) malloc(n*sizeof(int));
```

```
    printf("Enter the elements of array\n");
    for(i=0; i<n; i++)
      scanf("%d",a+i);

    choice();
}

//For selection which operation going to do
void choice()
{
    int op;
    printf("\n0. To Stop the process");
    printf("\n1. To insert an element");
    printf("\n2. To delete an element");
    printf("\n3. To Reverse array");
    printf("\nSelect one option\n");
    scanf("%d",&op);

    switch(op)
    {
        case 0: return;
                    break;
        case 1: insert();
                break;
        case 2: deletion();
                break;
        case 3: reversion();
                break;
        default: printf("\nEnter agian");
                     choice ();
    }
    display();
    printf("\nSelect \n\t1. Continue\n\t 2. Stop");
    scanf("%d",&op);
    if(op == 1)
        choice();
    else
        return;
}
```

```
//For insertion of an element
void insert()
{
   int ie,ip,i;
   printf("Enter the element to insert\n");
   scanf ("%d",&ie);
   printf("Enter the position where to insert\n");
   scanf("%d",&ip);
   n++;
   a = (int*)realloc(a,n*sizeof(int));
   for(i = ip; i < n+1; i++)
   {
      temp = *(a+i);
      *(a+i) = ie;
      ie = temp;
   }
}

//For deletion of element
void deletion()
{
   int dp;
   printf("Enter the position which to delete");
   scanf("%d",&dp);
   for(i = dp; i < n;i++)
      *(a+i) = *(a+i+1);

   n--;
   a = (int*)realloc(a,n*sizeof(int));
}

//To revese an array
void reversion()
{
   for(i = 0; i <= n/2;i++)
   {
      temp = a[i];
      a[i] = a[n-1-i];
      a[n-1-i] = temp;
```

```
    }
}

void display()
{
    printf("Array elements are\n");
    for(i = 0;i<n; i++)
        printf("%d\n",a[i]);
}
```

Input:

```
7
0
1
2
3
4
5
6
1
17
3
1
1
15
3
1
2
3
1
2
3
1
3
1
3
2
```

Output :

```
Enter the length of the array
Enter the elements of array

1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Enter the element to insert
Enter the position where to insert
Array elements are
0
1
2
17
3
4
5
6

Select
          1. Continue
           2. Stop
1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Enter the element to insert
Enter the position where to insert
Array elements are
0
1
2
15
17
3
4
5
6

Select
          1. Continue
```

```
              2. Stop
1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Enter the position which to deleteArray of after deletion
0
1
2
17
3
4
5
Array elements are
0
1
2
17
3
4
5
6

Select
             1. Continue
              2. Stop
1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Enter the position which to deleteArray of after deletion
0
1
2
3
4
5
Array elements are
0
1
```

```
2
3
4
5
6

Select
          1. Continue
           2. Stop
1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Array elements are
6
5
4
3
2
1
0

Select
          1. Continue
           2. Stop
1. To insert an element
2. To delete an element
3. To Reverse array
Select one option
Array elements are
0
1
2
3
4
5
6

Select
          1. Continue
```

2. Stop

Lab 2: A) Write a C program to implement Single linked list
i) Insertion ii) Deletion iii) Display

Aim :

To create a Single Linked List and do the Insertion, Deletion and Display operations on it.

https://ideone.com/VFbJ1L

Algorithm:

1.

Program :

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

int len = 0;
typedef struct node {
   int data;
   struct node *next;
} Node;

void choice (Node*);
Node* create_assign(int ,Node*);
int delete_after_node(Node*);
void display (Node*);
int insertion(Node** ,int);
int deletion(Node**,int);

int SearchNode(Node*, int );

void main()
{
   Node *head;
   choice(head);
}

//For Creating A node and Assigning Values
Node* create_assign(int val,Node *link)
{
   len++;
```

```
    Node *temp;
    temp = (Node*)malloc(sizeof(Node));

    temp->next = link;
    temp->data = val;
    return temp;
}

//Choice Selecting Function
void choice (Node *head)
{
    int op,ie,ip,dp,ei;
    printf("\n***********Menu***************\n");
    printf("\n\nChoose one option from below\n");

    printf("\n\t1.Insert at Beginning");
    printf("\n\t2.Insert at Specific position");
    printf("\n\t3.Insert at End");
    printf("\n\t4.Delete at Beginning");
    printf("\n\t5.Delete at Specific Position");
    printf("\n\t6.Delete at End");
    printf("\n\t7.To Find a data from List");
    printf("\n\t8.display Current Linked List");
    printf("\n\t9.To stop the process");

    printf("\n\n\tEnter your choice: ");
    scanf("%d",&op);

    switch(op)
    {
        case 1: printf("\n\tNode of value %d is inserted at front",insertion(&head,1));
                break;
        case 2: printf("\n\tEnter the position where to insert:\n");
                scanf("%d",&ip);

                if(ip <= len+1 && ip >=0)
                    printf("\n\tNode of value %d is inserted",insertion(&head,ip));
                else
                    printf("\n\tThere is no link at That position");
                break;
```

```
case 3: printf("\n\tNode of value %d is inserted at end",insertion(&head,len+1));
        break;
case 4: if(head == NULL){
            printf("\nThe List is Empty Nothing To Delete");
            break;
        }
        printf("\n\tNode of value %d is Deleted at front",deletion(&head,1));
        break;
case 5: if(head == NULL){
            printf("\nThe List is Empty Nothing To Delete");
            break;
        }
        printf("\n\tEnter the position where to Delete:\n");
        scanf("%d",&dp);

        if(dp <= len && dp >0)
            printf("\n\tNode of value %d is Deleted",deletion(&head,dp));
        else
            printf("\n\tThere is no link at That position");
        break;
case 6: if(head == NULL)
        {
            printf("\nThe List is Empty Nothing To Delete");
            break;
        }
        else{
            printf("\n\tNode of value %d is Deleted at end",deletion(&head,len));
        break;
        }
case 7: printf("\nEnter element to find\n");
        scanf("%d",&op);
        int i = SearchNode(head,op);
        if(i == -1 )
            printf("\nThe Element is not Found");
        else
            printf("\nThe Element is Found at Node %d",i);
        break;
case 8: display(head);
        break;
case 9: printf("\nThe Process Finished..");
```

```
            return;
        default:printf("\nEnter Valid Addess\n");
            choice(head);
            break;
    }
    choice(head);
}

//Display Function
void display (Node *head)
{
    Node *temp;
    temp = head;
    printf("\nThe Node Values Of Single linked list\n\t");
    while(temp)
    {
        printf("%d -> ",temp->data);
        temp = temp->next;
    }
    printf("NULL ");
}


//Element Inserting Function
int insertion(Node** head,int pos)
{
    int val,i = 1;
    Node *temp;
    Node *new_node;

    temp = *head;
    printf("\n\tEnter the element to insert: ");
    scanf("%d",&val);

    if(pos == 1)
    {
        new_node = create_assign(val,temp);
        *head = new_node;
        return (*head)->data;
    }
    else
    {
        while(i < (pos - 1))
        {
            temp = temp->next;
            i++;
        }

        new_node = create_assign(val,temp->next);
        temp->next = new_node;
        return temp->next->data;
    }
}


int deletion(Node **head, int pos)
{
    Node *temp = *head;
    int d,i = 1;

    if(pos == 1)
    {
        *head = temp->next;
        len--;
        return temp->data;
    }

    while(i < (pos - 1))
    {
        temp = temp->next;
        i++;
    }


    len--;

    Node *deleting_node = temp->next;
    d = deleting_node->data;

    temp->next = deleting_node->next;
    free(deleting_node);
    return d;
}



int SearchNode(Node* head,int value)
{
    int i = 1;
    Node* temp = head;

    while((temp) && (temp->data != value))
    {
        temp = temp->next;
        i++;
    }

    if( i > len)
        return -1;
    else
        return i;

}
```

Input :

```
1
1
3
3
3
4
2
2
2
8
7
3
4
5
2
```

```
    8
    6
    6
    8
    9
```

Output :

```
    ************Menu****************


    Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Enter the element to insert:
        Node of value 1 is inserted at front
    ************Menu****************


    Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process
        Enter your choice:
        Enter the element to insert:
        Node of value 3 is inserted at end
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Enter the element to insert:
        Node of value 4 is inserted at end
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Enter the position where to insert:
        Enter the element to insert:
        Node of value 2 is inserted
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
The Node Values Of Single linked list
        1 -&gt; 2 -&gt; 3 -&gt; 4 -&gt; NULL
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
Enter element to find

The Element is Found at Node 3
************Menu****************
```

```
Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Node of value 1 is Deleted at front
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Enter the position where to Delete:

        Node of value 3 is Deleted
************Menu****************


Choose one option from below
```

```
        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
The Node Values Of Single linked list
        2 -&gt; 4 -&gt; NULL
***********Menu***************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Node of value 4 is Deleted at end
***********Menu***************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
        Node of value 2 is Deleted at end
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process

        Enter your choice:
The Node Values Of Single linked list
        NULL
************Menu****************


Choose one option from below

        1.Insert at Beginning
        2.Insert at Specific position
        3.Insert at End
        4.Delete at Beginning
        5.Delete at Specific Position
        6.Delete at End
        7.To Find a data from List
        8.display Current Linked List
        9.To stop the process
```

```
          Enter your choice:
  The Process Finished..
```

Lab 3: A) Write a C program to implement Doubly linked list
i) Insertion ii) Deletion. iii)Display

Program :

```
#include<stdio.h>
#include<stdlib.h>
//Double linked program of Sandeep
typedef struct node {
   struct node* prev;
   int data;
   struct node* next;
} Node;

Node* head;
Node* tail;
int len = 0;

Node* create_assign(Node*,int, Node* );
int isEmpty();
int insertion_beg(int );
int insertion_end(int );
int insertion_spe(int, int);
int deletion_beg();
int deletion_end();
int deletion_spe(int);
void display();
int SearchNode(int);
void choice();
void main() {
   choice();
}


Node* create_assign(Node* prev_link,int val, Node* next_link) {
   Node *new_node = (Node*)malloc(sizeof(Node));
   new_node->data = val;
   new_node->prev = prev_link;
   new_node->next = next_link;
   if(new_node == NULL) {
      printf("\n\tOverflow occurred \n\tinsertion not possible");
```

```
        return NULL;
    }
    len++;
    return new_node;
}
int isEmpty() {
    if(head == NULL) {
        printf("\n\tList is empty or Underflow occuerd\n\t");
        return 1;
    }
    return 0;
}
int insertion_beg(int val) {
    Node* new_node = create_assign(NULL, val, head);
    head = new_node;
    if(len == 1)
        tail = new_node;
    printf("\n\tNode of value %d inserted at front",head->data);
    return head->data;
}
int insertion_end(int val) {
    Node* new_node = create_assign(tail, val, NULL);
    tail->next = new_node;
    tail = new_node;
    printf("\n\tNode of value %d inserted at end",tail->data);
    return tail->data;
}
int insertion_spe(int val, int pos) {
    if((pos == 1) || (len == 0))
        return insertion_beg(val);
    else if(pos == (len+1))
        return insertion_end(val);

    Node *temp;
    Node *new_node;
    temp = head;

    int i = 1;
    while( i < (pos-1)){
        temp = temp->next;
    }


    /*for(int i = 1; i < pos; i++)
        temp = temp->next;

    new_node = create_assign(temp->prev, val, temp);
    temp->prev->next = new_node;
    temp->prev = new_node;*/

    new_node = create_assign(temp, val, temp->next);
    temp->next->prev = new_node;
    temp->next = new_node;
    printf("\n\tNode of value %d inserted at Position %d",temp->next->data,pos);
    return temp->next->data;
}
int deletion_beg() {
    if(isEmpty())
        return -1;
    int d = head->data;
    Node* deleting_node = head;
    head = head->next;
    head->prev = NULL;
    len--;
    printf("\n\tNode of value %d deleted at front",d);
    free(deleting_node);
    return d;
}
int deletion_end() {
    if(isEmpty())
        return -1;
    int d = tail->data;
    Node* deleting_node = tail;
    tail = tail->prev;
    tail->next = NULL;
    len--;
    printf("\n\tNode of value %d deleted at end",d);
    free(deleting_node);
    return d;
}
```

```
int deletion_spe(int pos) {
    if(isEmpty())
        return -1;
    if(pos == 1 || len == 1)
        return deletion_beg();
    else if(pos == len)
        return deletion_end();

    int i = 1;
    Node* deleting_node;
    Node* temp; temp = head;
    while(i < pos) {
        i++;
        temp = temp->next;
    }
    deleting_node = temp;
    temp->prev->next = temp->next;
    temp->next->prev = temp->prev;
    int d = deleting_node->data;
    len--;
    printf("\n\tNode of value %d deleted at position %d",d,pos);
    free(deleting_node);
    return d;
}
void display() {
    Node* temp;
    temp = head;
    if(isEmpty()) {
        printf("\n\tNothing to Display");
        return;
    }
    printf("\n\tHead ");
    int i = 0;
    do {
        i++;
        printf("<-> %d ",temp->data);
        temp = temp->next;
    } while(i < len);
    printf("-> NULL");
```

```
}
int SearchNode(int val) {
   Node *temp;
   temp = head;
   int i = 0;
   while(temp->next ) {
      i++;
      if(temp->data == val) {
         printf("\n\tThe data %d is found at Node %d",val,i);
         return  i;
      }
      temp = temp->next;
   }
   printf("\n\tThe data is not found: ");
   return -1;
}
void choice() {
   int op,ie,ip,dp;
   printf("\n************Menu****************\n");
   printf("\n\nChoose one option from below\n");

   printf("\n\t1.Insert at Beginning");
   printf("\n\t2.Insert at Specific position");
   printf("\n\t3.Insert at End");
   printf("\n\t4.Delete at Beginning");
   printf("\n\t5.Delete at Specific Position");
   printf("\n\t6.Delete at End");
   printf("\n\t7.To Find a data from List");
   printf("\n\t8.To Delete specific data");
   printf("\n\t9.display Current Linked List");
   printf("\n\t10.To stop the process");

   printf("\n\n\tEnter your choice: ");
   scanf("%d",&op);

   if((op == 1) || (op == 2) || (op == 3)) {
      printf("\n\tEnter the value to insert: ");
      scanf("%d",&ie);
   }
   switch(op)
```

```
{
case 1:
   insertion_beg(ie);
   break;
case 2:
   printf("\n\tEnter the position where to insert:\n");
   scanf("%d",&ip);

   if((ip <= (len + 1)) && (ip >= 0))
      insertion_spe(ie,ip);
   else
      printf("\n\tThere is no link at That position");
   break;
case 3:
   insertion_end(ie);
   break;
case 4:
   if(isEmpty())
      break;

   deletion_beg();
   break;
case 5:
   if(isEmpty())
      break;
   printf("\n\tEnter the position where to Delete:\n");
   scanf("%d",&dp);

   if(dp <= len && dp >0)
      deletion_spe(dp);
   else
      printf("\n\tThere is no link at That position");
   break;
case 6:
   if(isEmpty())
      break;
   deletion_end();
   break;

case 7:
```

```
            if(isEmpty())
                break;
            printf("\nEnter element to find\n");
            scanf("%d",&op);
            SearchNode(op);
            break;
        case 8:
            if(isEmpty())
                break;
            printf("\n\tEnter data to delete\n");
            scanf("%d",&op);
            deletion_spe(SearchNode(op));
        case 9:
            display();
            break;
        case 10:
            printf("\nThe Process Finished..");
            return;
        default:
            printf("\nEnter Valid Addess\n");
            choice();
            break;
        }
        choice();
    }
```

Input :

```
1
1
1
0
3
2
3
3
3
4
3
5
2
```

```
        7
        2
        9
        5
        2
        4
        6
        9
        7
        3
        8
        2
        9
        10
```

Output :

```
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
                    Enter the value to insert:
                    Node of value 1 inserted at front
************Menu****************


Choose one option from below
```

```
            1.Insert at Beginning
            2.Insert at Specific position
            3.Insert at End
            4.Delete at Beginning
            5.Delete at Specific Position
            6.Delete at End
            7.To Find a data from List
            8.To Delete specific data
            9.display Current Linked List
            10.To stop the process

            Enter your choice:
            Enter the value to insert:
            Node of value 0 inserted at front
************Menu****************


Choose one option from below

            1.Insert at Beginning
            2.Insert at Specific position
            3.Insert at End
            4.Delete at Beginning
            5.Delete at Specific Position
            6.Delete at End
            7.To Find a data from List
            8.To Delete specific data
            9.display Current Linked List
            10.To stop the process

            Enter your choice:
            Enter the value to insert:
            Node of value 2 inserted at end
************Menu****************


Choose one option from below

            1.Insert at Beginning
            2.Insert at Specific position
```

```
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
                    Enter the value to insert:
                    Node of value 3 inserted at end
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
                    Enter the value to insert:
                    Node of value 4 inserted at end
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
```

```
              5.Delete at Specific Position
              6.Delete at End
              7.To Find a data from List
              8.To Delete specific data
              9.display Current Linked List
              10.To stop the process

              Enter your choice:
              Enter the value to insert:
              Node of value 5 inserted at end
************Menu****************


Choose one option from below

              1.Insert at Beginning
              2.Insert at Specific position
              3.Insert at End
              4.Delete at Beginning
              5.Delete at Specific Position
              6.Delete at End
              7.To Find a data from List
              8.To Delete specific data
              9.display Current Linked List
              10.To stop the process

              Enter your choice:
              Enter the value to insert:
              Enter the position where to insert:

              Node of value 7 inserted at Position 2
************Menu****************


Choose one option from below

              1.Insert at Beginning
              2.Insert at Specific position
              3.Insert at End
              4.Delete at Beginning
```

```
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
                    Head &lt;-&gt; 0 &lt;-&gt; 7 &lt;-&gt; 1 &lt;-&gt; 2 &lt;-&gt; 3
&lt;-&gt; 4 &lt;-&gt; 5 -&gt; NULL
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
                    Enter the position where to Delete:

                    Node of value 7 deleted at position 2
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
```

```
                        6.Delete at End
                        7.To Find a data from List
                        8.To Delete specific data
                        9.display Current Linked List
                        10.To stop the process

                        Enter your choice:
                        Node of value 0 deleted at front
************Menu****************


Choose one option from below

                        1.Insert at Beginning
                        2.Insert at Specific position
                        3.Insert at End
                        4.Delete at Beginning
                        5.Delete at Specific Position
                        6.Delete at End
                        7.To Find a data from List
                        8.To Delete specific data
                        9.display Current Linked List
                        10.To stop the process

                        Enter your choice:
                        Node of value 5 deleted at end
************Menu****************


Choose one option from below

                        1.Insert at Beginning
                        2.Insert at Specific position
                        3.Insert at End
                        4.Delete at Beginning
                        5.Delete at Specific Position
                        6.Delete at End
                        7.To Find a data from List
                        8.To Delete specific data
                        9.display Current Linked List
```

```
                    10.To stop the process

                    Enter your choice:
                    Head &lt;-&gt; 1 &lt;-&gt; 2 &lt;-&gt; 3 &lt;-&gt; 4 -&gt; NULL
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process

                    Enter your choice:
Enter element to find

                    The data 3 is found at Node 3
************Menu****************


Choose one option from below

                    1.Insert at Beginning
                    2.Insert at Specific position
                    3.Insert at End
                    4.Delete at Beginning
                    5.Delete at Specific Position
                    6.Delete at End
                    7.To Find a data from List
                    8.To Delete specific data
                    9.display Current Linked List
                    10.To stop the process
```

```
                  Enter your choice:
                  Enter data to delete

                  The data 2 is found at Node 2
                  Node of value 2 deleted at position 2
                  Head &lt;-&gt; 1 &lt;-&gt; 3 &lt;-&gt; 4 -&gt; NULL
************Menu****************


Choose one option from below

                  1.Insert at Beginning
                  2.Insert at Specific position
                  3.Insert at End
                  4.Delete at Beginning
                  5.Delete at Specific Position
                  6.Delete at End
                  7.To Find a data from List
                  8.To Delete specific data
                  9.display Current Linked List
                  10.To stop the process

                  Enter your choice:
                  Head &lt;-&gt; 1 &lt;-&gt; 3 &lt;-&gt; 4 -&gt; NULL
************Menu****************


Choose one option from below

                  1.Insert at Beginning
                  2.Insert at Specific position
                  3.Insert at End
                  4.Delete at Beginning
                  5.Delete at Specific Position
                  6.Delete at End
                  7.To Find a data from List
                  8.To Delete specific data
                  9.display Current Linked List
                  10.To stop the process
```

```
                    Enter your choice:
The Process Finished..
```