Data structure & Algorithms

Assignment-4 (Multi-dimensional Arrays and Polynomials)

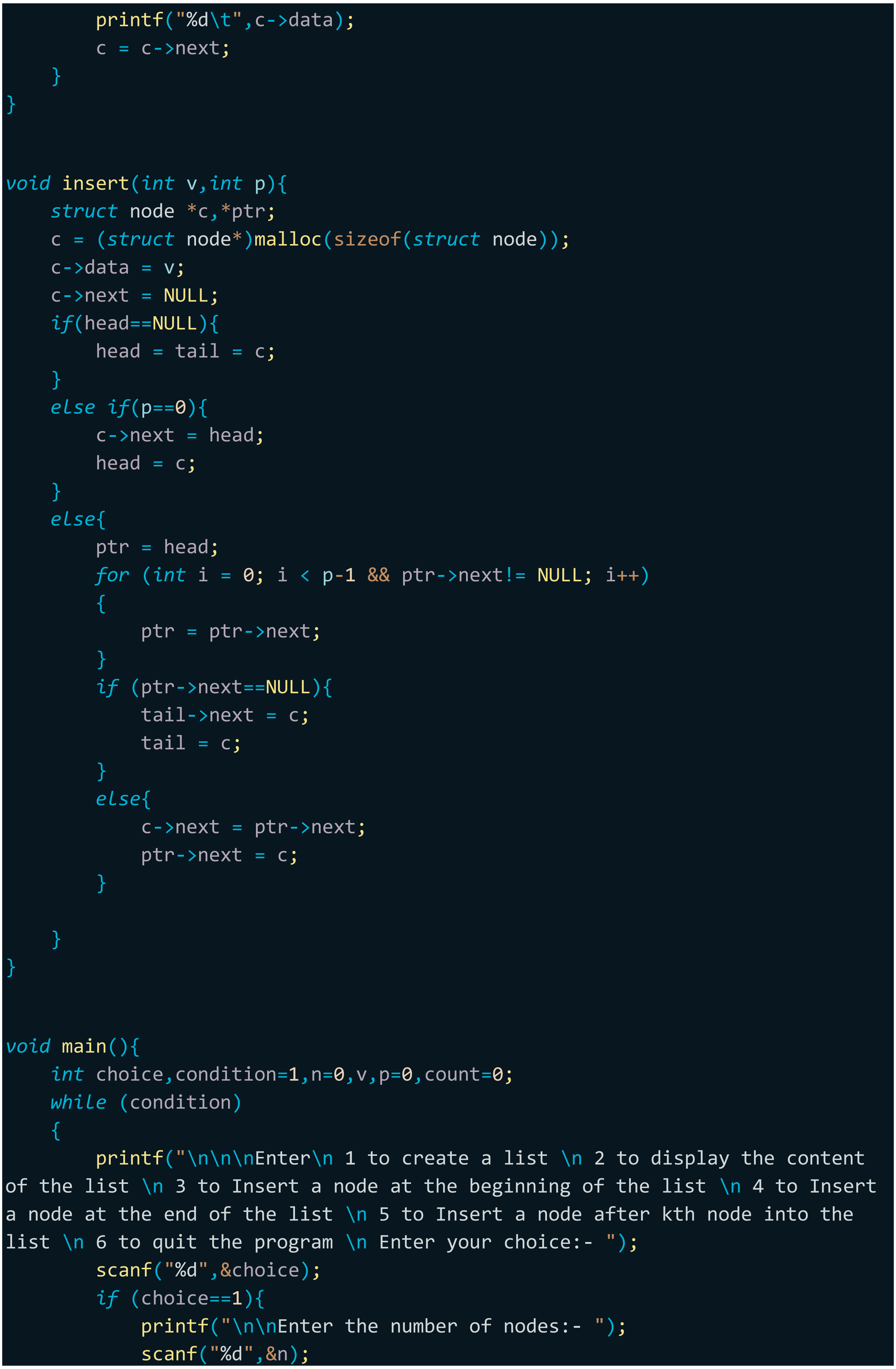
BTech **Name : Aditya Narayana Choudhury**
Date: 10/08/2022 **Roll No. : 2129011**

Lab Assignment:

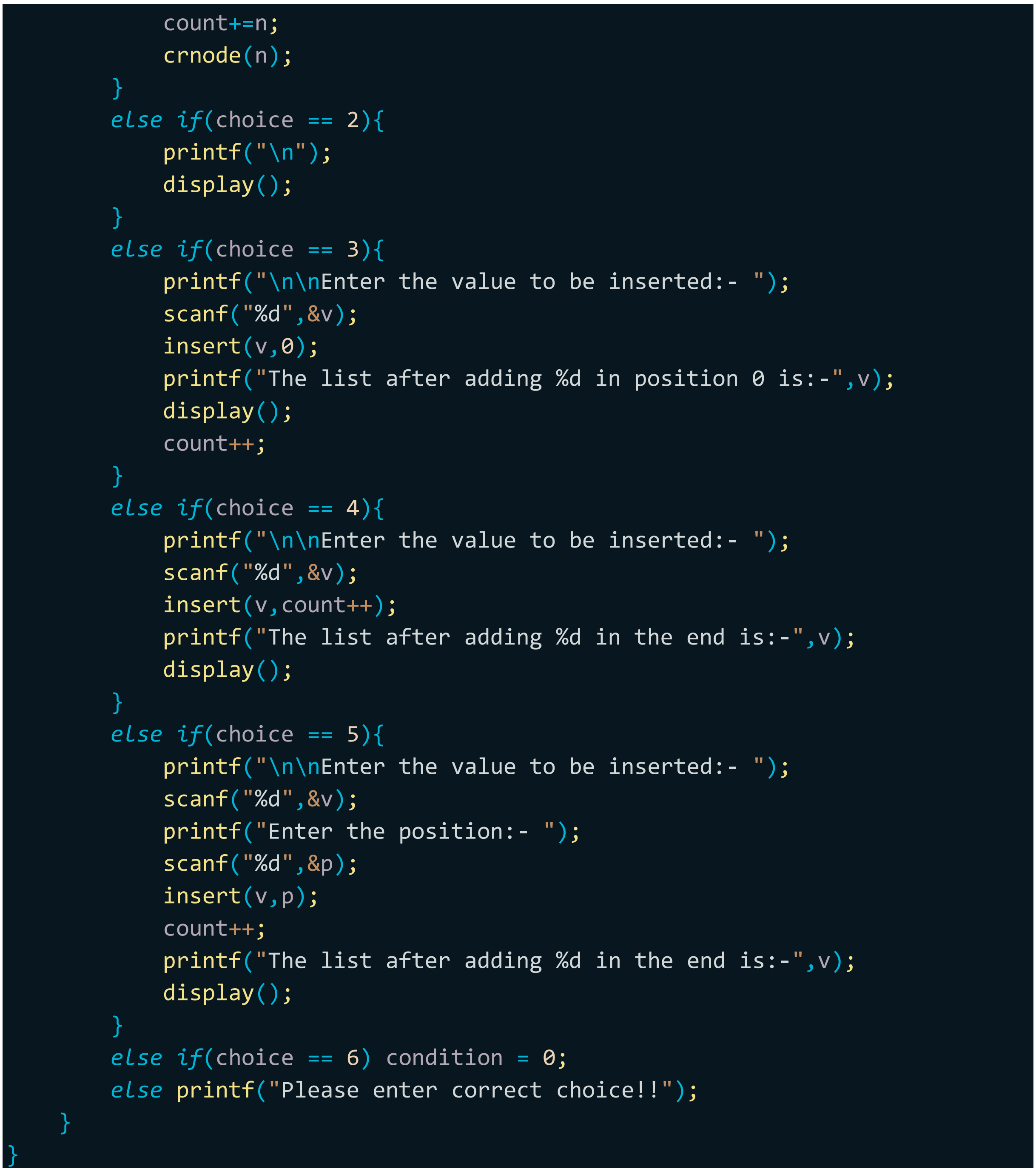
1. Write a Menu driven program in C to perform following in Single linked list.
    a. Create a list.
    b. Display the content of the list.
    c. Insert a node at the beginning of the list.
    d. Insert a node at the end of the list.
    e. Insert a node after k^{th} node in to the list.

Code:-

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

struct node{
    int data;
    struct node* next;
}*head=NULL,*tail=NULL;

int crnode(int n){
    struct node* c;
    for (int i = 0; i < n; i++)
    {
        c = (struct node*)malloc(sizeof(struct node));
        printf("Enter the value of element %d:- ",i);
        scanf("%d",&c->data);
        c->next = NULL;
        if (head == NULL){
            head = tail = c;
        }
        else{
            tail->next = c;
            tail = c;
        }
    }

}

void display(){
    struct node* c;
    c = head;
    while(c!=NULL){
```

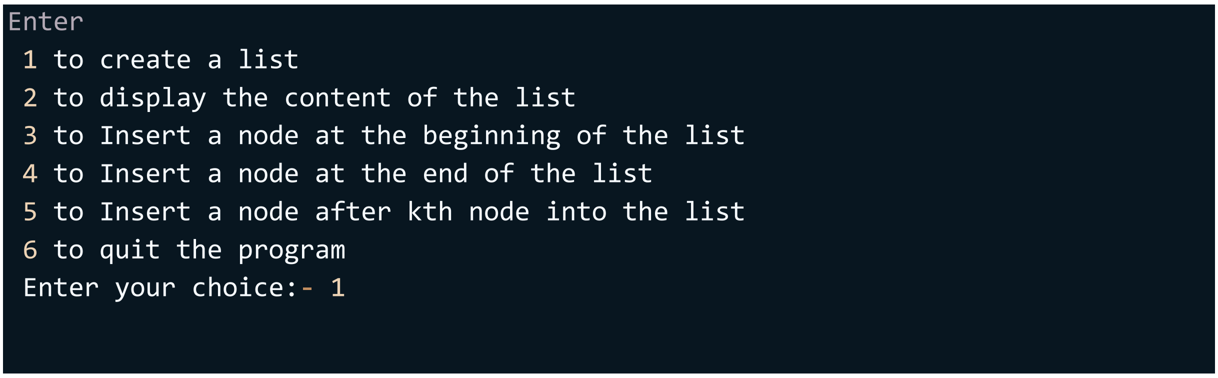


```c
        printf("%d\t",c->data);
        c = c->next;
    }
}


void insert(int v,int p){
    struct node *c,*ptr;
    c = (struct node*)malloc(sizeof(struct node));
    c->data = v;
    c->next = NULL;
    if(head==NULL){
        head = tail = c;
    }
    else if(p==0){
        c->next = head;
        head = c;
    }
    else{
        ptr = head;
        for (int i = 0; i < p-1 && ptr->next!= NULL; i++)
        {
            ptr = ptr->next;
        }
        if (ptr->next==NULL){
            tail->next = c;
            tail = c;
        }
        else{
            c->next = ptr->next;
            ptr->next = c;
        }

    }
}


void main(){
    int choice,condition=1,n=0,v,p=0,count=0;
    while (condition)
    {
        printf("\n\n\nEnter\n 1 to create a list \n 2 to display the content
of the list \n 3 to Insert a node at the beginning of the list \n 4 to Insert
a node at the end of the list \n 5 to Insert a node after kth node into the
list \n 6 to quit the program \n Enter your choice:- ");
        scanf("%d",&choice);
        if (choice==1){
            printf("\n\nEnter the number of nodes:- ");
            scanf("%d",&n);
```

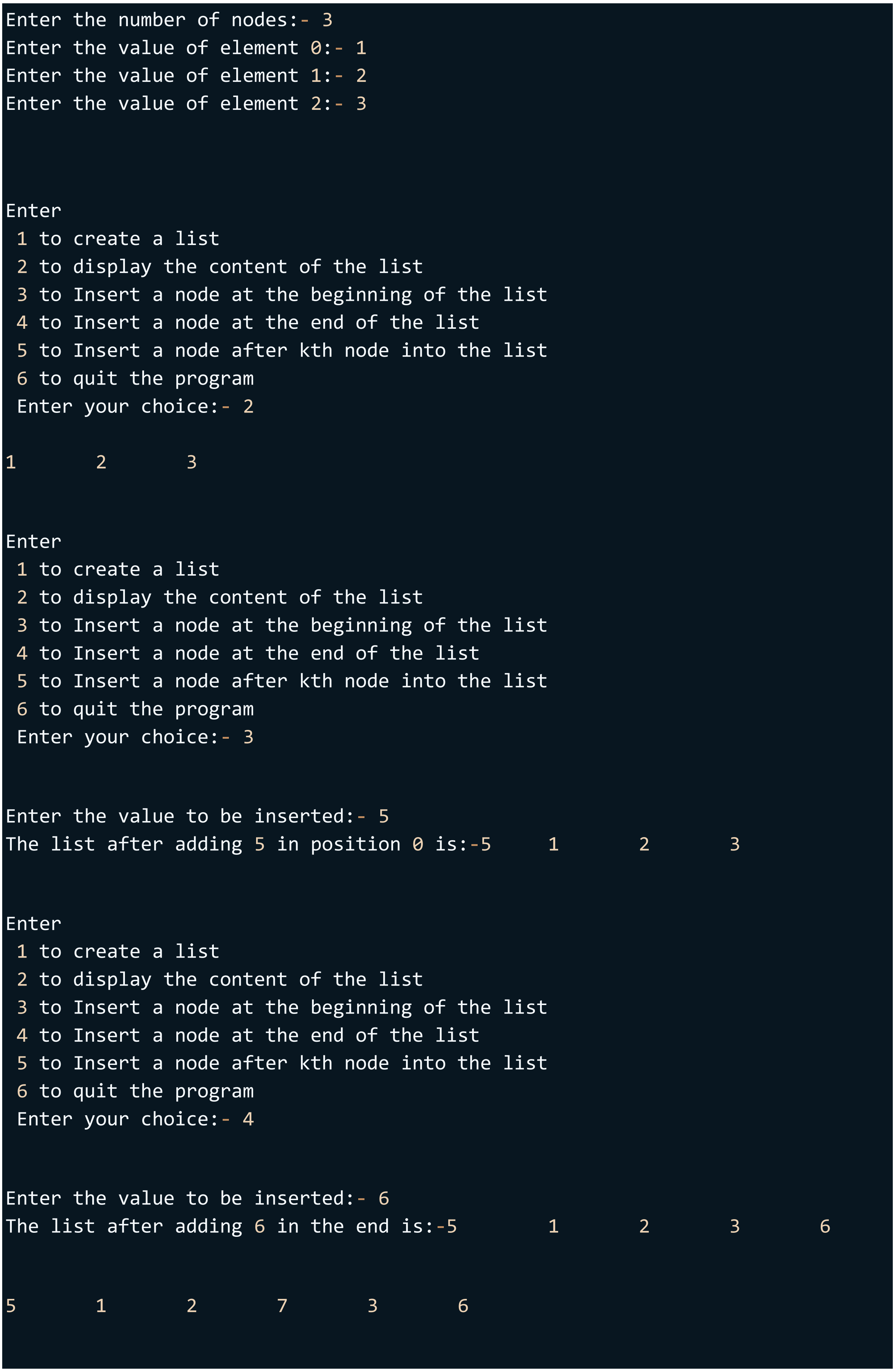


```c
            count+=n;
            crnode(n);
        }
        else if(choice == 2){
            printf("\n");
            display();
        }
        else if(choice == 3){
            printf("\n\nEnter the value to be inserted:- ");
            scanf("%d",&v);
            insert(v,0);
            printf("The list after adding %d in position 0 is:-",v);
            display();
            count++;
        }
        else if(choice == 4){
            printf("\n\nEnter the value to be inserted:- ");
            scanf("%d",&v);
            insert(v,count++);
            printf("The list after adding %d in the end is:-",v);
            display();
        }
        else if(choice == 5){
            printf("\n\nEnter the value to be inserted:- ");
            scanf("%d",&v);
            printf("Enter the position:- ");
            scanf("%d",&p);
            insert(v,p);
            count++;
            printf("The list after adding %d in the end is:-",v);
            display();
        }
        else if(choice == 6) condition = 0;
        else printf("Please enter correct choice!!");
    }
}
```

Output:-



```
Enter
 1 to create a list
 2 to display the content of the list
 3 to Insert a node at the beginning of the list
 4 to Insert a node at the end of the list
 5 to Insert a node after kth node into the list
 6 to quit the program
 Enter your choice:- 1
```



```
Enter the number of nodes:- 3
Enter the value of element 0:- 1
Enter the value of element 1:- 2
Enter the value of element 2:- 3



Enter
 1 to create a list
 2 to display the content of the list
 3 to Insert a node at the beginning of the list
 4 to Insert a node at the end of the list
 5 to Insert a node after kth node into the list
 6 to quit the program
 Enter your choice:- 2

1       2       3


Enter
 1 to create a list
 2 to display the content of the list
 3 to Insert a node at the beginning of the list
 4 to Insert a node at the end of the list
 5 to Insert a node after kth node into the list
 6 to quit the program
 Enter your choice:- 3


Enter the value to be inserted:- 5
The list after adding 5 in position 0 is:-5     1       2       3


Enter
 1 to create a list
 2 to display the content of the list
 3 to Insert a node at the beginning of the list
 4 to Insert a node at the end of the list
 5 to Insert a node after kth node into the list
 6 to quit the program
 Enter your choice:- 4


Enter the value to be inserted:- 6
The list after adding 6 in the end is:-5        1       2       3       6


5       1       2       7       3       6
```

```
Enter
 1 to create a list
 2 to display the content of the list
 3 to Insert a node at the beginning of the list
 4 to Insert a node at the end of the list
 5 to Insert a node after kth node into the list
 6 to quit the program
 Enter your choice:- 6
```

2. Write a program in C to determine whether the given matrix is a sparse matrix or not.

Code:-

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


void main(){
    int m,n,count=0;
    printf("Enter rows columns:- ");
    scanf("%d %d",&m,&n);
    int** arr = (int**)malloc(m*sizeof(int*));
    for(int i=0;i<m;i++){
        *(arr+i) = (int*)malloc(n*sizeof(int));
    }

    printf("Enter the elements:- ");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d",(*(arr+i)+j));
            if (*(*(arr+i)+j) == 0) count++;
        }
    }
    if (count>(m*n)/2) printf("It is a sparse matrix!!\n");
    else printf("It is not a sparse matrix!!\n");

}
```

Output:-

```
Enter rows columns:- 3 4
Enter the elements:-
1 2 0 0
0 0 0 0
3 4 0 0
It is a sparse matrix!!
```

3. Write a program in C to find out the transpose of a sparse matrix.

Sample Input:			**Sample Output:**		
Row	Col	Value	Row	Col	Value
6	6	8	6	6	8
0	0	15	0	0	15
0	3	22	0	4	91
0	5	-15	1	1	11
1	1	11	2	1	3
1	2	3	2	5	28
2	3	-6	3	0	22
4	0	91	3	2	-6
5	2	28	5	0	-15

Code:-

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

void transpose(int** arr,int** arr2,int n){
    for (int i = 0; i < n; ++i) for (int j = 0; j < n; ++j) arr2[j][i] =
arr[i][j];
}

void print(int** arr,int n){
    printf("Row\tColumn\tValue\n");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if(arr[i][j]!=0){
                printf("%d\t%d\t%d\n",i,j,arr[i][j]);
            }
        }
    }

}

void main(){
    int n,row,column,value;
    printf("Enter the number of rows:- ");
    scanf("%d",&n);
    int** arr = (int**)malloc(n*sizeof(int*));
    int** arr2 = (int**)malloc(n*sizeof(int*));
    for(int i=0;i<n;i++){
        *(arr+i) =(int*)calloc(n,sizeof(int));
        *(arr2+i) =(int*)calloc(n,sizeof(int));
    }
    int n1;
    printf("Enter the number of non-zero elements:- ");
    scanf("%d",&n1);
    printf("Enter rows columns values:-");
    for(int i=0;i<n1;i++){
        scanf("%d %d %d",&row,&column,&value);
        arr[row][column] = value;
    }
    printf("The matrix entered by the user is:- \n");
    print(arr,n);
    transpose(arr,arr2,n);
    printf("The matrix after the transpose is:- \n");
    print(arr2,n);
}
```

Output:-

```
Enter the number of rows:- 7
Enter the number of non-zero elements:- 9
Enter rows columns values:-
6       6       8
0       0       15
0       3       22
0       5       -15
1       1       11
1       2       3
2       3       -6
4       0       91
5       2       28
The matrix entered by the user is:-
Row     Column  Value
0       0       15
0       3       22
0       5       -15
1       1       11
1       2       3
2       3       -6
4       0       91
5       2       28
6       6       8
The matrix after the transpose is:-
Row     Column  Value
0       0       15
0       4       91
1       1       11
2       1       3
2       5       28
3       0       22
```

```
3       2       -6
5       0       -15
6       6       8
```

4. Write a program in C to determine whether the given matrix is a lower triangular or upper triangular or tri-diagonal matrix.

Code:-

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

bool tri_diagonal(int** arr, int k){
    for(int i=0; i<k; i++){
        for(int j=0; j<k; j++){
            if (i == j && arr[i][j] != 0){
                if (arr[i][j+1]==0 && j<k){
                    return false;
                }
                if(arr[i][j-1]==0 && j>0){
                    return false;
                }
            }
        }
    }
return true;
}

bool upper_diagonal(int** arr, int n)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
        {
            if (j>=i && arr[i][j]==0) return false;
        }
    }
    return true;
}

bool lower_diagonal(int** arr, int n)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
        {
            if (i>=j && arr[i][j]==0) return false;
        }
    }
    return true;
}


int main(){
    int n=0;
    printf("Enter the size of array: ");
    scanf("%d",&n);

    int** arr= (int**)malloc(n*sizeof(int*));
    for(int i=0;i<n;i++){
        *(arr+i) = (int*)malloc(n*sizeof(int));
    }

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

    if(arr[0][n-1]==0 && lower_diagonal(arr,n)) printf("Lower Triangular!!");
    else if(arr[n-1][0]==0 && upper_diagonal(arr,n)) printf("Upper
triangular!!");
    else if (arr[n-1][0]==0 && arr[0][n-1]==0 && tri_diagonal(arr,n))
printf("Tri-Diagonal!!");
    else printf("Not a triangular matrix!!");
    return 0;
}
```

Output:-

```
Enter the size of array: 3
Enter the array elements:
1 2 3
0 4 5
0 0 6
Upper triangular!!
```

5. Write a program in C to add two polynomials with single variable. Take the polynomials from user and store them into suitable data structure.

Sample Input:

Polynomial 1 = $5x^7 - 3x^5 + x^2 + 9$

Polynomial 2 = $2x^2 + 1$

Sample Output:

New polynomial = $5x^7 - 3x^5 + 3x^2 + 10$

Code:-

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

int sum(int* arr1,int* arr2,int* arr3,int z, int q){
    int last = z,c=0;
    for (int i = 0; i < z; i=i+2)
    {
        arr3[i] = arr1[i];
        arr3[i+1] = arr1[i+1];
    }

    for (int i = 0; i < q; i=i+2)
    {
        c=0;
        for (int j = 0; j < z; j=j+2)
        {
            if(arr2[i]==arr3[j]){
                arr3[j+1] = arr3[j+1]+arr2[i+1];
                c=1;
                break;
            }
        }
        if(c==0){
            arr3 = realloc(arr3,z+(2*sizeof(int)));
            arr3[last++] = arr2[i];
            arr3[last++] = arr2[i+1];
        }

    }
    return last;
}


void main(){
    int m,n,i;

    printf("Enter the number of terms of 1st polynomial:- ");
    scanf("%d",&m);
    int z = m*2;
    int* arr1 = (int*)malloc(z*sizeof(int));
    printf("Enter power coefficient:- ");
    for (i = 0; i < z; i++)
    {
        scanf("%d",arr1+i);
    }

    printf("Enter the number of terms of 2nd polynomial:- ");
    scanf("%d",&n);
    int q = n*2;
    int* arr2 = (int*)malloc(q*sizeof(int));
    printf("Enter power coefficient:- ");
    for (i = 0; i < q; i++)
    {
        scanf("%d",arr2+i);
    }

    int h = (z+q)*sizeof(int);
    int* arr3 = (int*)malloc(z*sizeof(int));
    int last=sum(arr1,arr2,arr3,z,q);

    for (int i = 0; i < last; i=i+2)
    {
        if (i==last-2) printf("%dx^%d",arr3[i+1],arr3[i]);
        else printf("%dx^%d + ",arr3[i+1],arr3[i]);
    }

}
```

Output:-

```
Enter the number of terms of 1st polynomial:- 2
Enter power coefficient:- 6 4 3 2
Enter the number of terms of 2nd polynomial:- 4
Enter power coefficient:- 9 5 7 4 3 8 1 6
4x^6 + 10x^3 + 5x^9 + 4x^7 + 6x^1
```