**Program 6)**

**Design a doubly linked list to represent sparse matrix. Each node in the list can have the row and column index of the matrix element and the value of the element. Print the complete matrix as the output.**

**Program :**

#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

struct node

{

int data;

int r, c;

struct node\* prev;

struct node\* next;

};

typedef struct node\* NODE;

NODE getnode()

{

NODE p;

p=(NODE)malloc(sizeof(struct node));

p->prev=NULL;

p->next=NULL;

return p;

}

NODE insertlast(NODE list,int n,int r,int c)

{

NODE p,q;

p=getnode();

p->data=n;

p->r=r;

p->c=c;

if(list==NULL)

{

return p;

}

q=list;

while(q->next!=NULL)

{

q=q->next;

}

p->prev=q;

q->next=p;

return list;

}

int main()

{

NODE p;

NODE q;

p=NULL;

int i,j,num,m,n;

printf("enter the number of rows of the matrix\n");

scanf("%d",&m);

printf("enter the number of columns of the matrix\n");

scanf("%d",&n);

printf("enter the elements of the matrix\n ");

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++)

{

scanf("%d",&num);

if(num!=0)

{

p=insertlast(p,num,i,j);

}

}

}

q=p;

printf("the matrix is\n ");

for(i=1;i<=m;i++)

{

for(j=1;j<=n;j++)

{

if(q!=NULL&&i==q->r&&j==q->c)

{

printf("%4d ",q->data);

q=q->next;

}

else

{

printf("%4d ",0);

}

}

printf("\n");

printf("\n");

}

}

**OUTPUTS )**

**Output1:**

enter the number of rows of the matrix

5

enter the number of columns of the matrix

5

enter the elements of the matrix

enter the number of rows of the matrix

5

enter the number of columns of the matrix

5

enter the elements of the matrix

0 0 0 0 0 0 25 31 0 0 45 8 9 0 0 0 0 1 0 63 0 0 0 0 0

the matrix is

0 0 0 0 0

0 25 31 0 0

45 8 9 0 0

0 0 1 0 63

0 0 0 0 0

**Output2 :**

enter the number of rows of the matrix

6

enter the number of columns of the matrix

6

enter the elements of the matrix

0 0 0 0 0 0 45 1 23 0 0 0 0 0 0 12 0 0 0 0 49 0 0 0 0 0 0 0 35 0 78 0 0 0 0 1

the matrix is

0 0 0 0 0 0

45 1 23 0 0 0

0 0 0 12 0 0

0 0 49 0 0 0

0 0 0 0 35 0

78 0 0 0 0 1

**Output 3:**

enter the number of rows of the matrix

6

enter the number of columns of the matrix

7

enter the elements of the matrix

0 0 0 0 0 0 4 0 3 1 5 1 0 0 0 0 0 0 0 0 12 78 0 0 0 0 0 6 0 0 0 7 89 0 0 0 0 0 45 0 3 0

the matrix is

0 0 0 0 0 0 4

0 3 1 5 1 0 0

0 0 0 0 0 0 12

78 0 0 0 0 0 6

0 0 0 7 89 0 0

0 0 0 45 0 3 0