sparse

```
#include <stdio.h>

int main(void) {
  int arr[5][5],i,j,r,c,row[10],col[10],n,val[10],k=0;
  printf("Enter how many row in 2D:");
  scanf("%d",&r);
  printf("Enter how many columns in 2D:");
  scanf("%d",&c);
  printf("Enter %d x %d array elements.....\n",r,c);
  for(i=0;i<r;i++)
  {
    for(j=0;j<c;j++)
    {
      printf("Enter element[%d][%d]:",i+1,j+1);
      scanf("%d",&arr[i][j]);
    }
  }
  printf("Array .....\n");
  for(i=0;i<r;i++)
  {
    for(j=0;j<c;j++)
    {
      printf("%d\t",arr[i][j]);
    }
    printf("\n");
  }
  for(i=0;i<r;i++)
  {
    for(j=0;j<c;j++)
    {
      if(arr[i][j]!=0)
      {
        row[k]=i+1;
        col[k]=j+1;
        val[k]=arr[i][j];
        k++;
      }
    }
  }
  printf("Your matrix in sparse matrix form....\n");
  printf("Element No\tRow\tColumn\tValue\n");
  for(i=0;i<k;i++)
```

	0	1	2	3
0	0	0	1	0
1	0	3	0	0
2	0	0	0	0
3	0	0	0	2

	row	col
0	1	3
1	2	2
2	4	4

val
1
3
2

```
R=4
c=4
Array .....
0       0       1       0
0       3       0       0
0       0       0       0
0       0       0       2
i=0, 1, 2, 3
j=0, 1, 2
arr[1][2]
k=0 , 1, 2, 3
Your matrix in sparse matrix form....
Element No Row Column Value
1                   1      3       1
2                   2      2       3
3                   4      4       2
```

```
  {
   printf("%d\t%d\t%d\t%d\n",i+1,row[i],col[i],val[i]);
  }
 return 0;
}
```

output

```
Enter how many row in 2D:3

Enter how many columns in 2D:3

Enter 3 x 3 array elements.....

Enter element[1][1]:0

Enter element[1][2]:0

Enter element[1][3]:3

Enter element[2][1]:0

Enter element[2][2]:2

Enter element[2][3]:0

Enter element[3][1]:4

Enter element[3][2]:0

Enter element[3][3]:0型

Array .....

0   0   3

0   2   0

4   0   0

Your matrix in sparse matrix form....

Element No  Row Column  Value
```

1	1	3	3
2	2	2	2
3	3	1	4

Mattospar

```
#include <stdio.h>

int main(void) {
 int r,c,n,i,j,row[10],col[10],val[10],k;
 printf("Enter how many rows in your sparse matrix:");
 scanf("%d",&r);
 printf("Enter how many columns in your sparse matrix:");
 scanf("%d",&c);
 printf("Enter how many non - zero elements in your sparse matrix:");
 scanf("%d",&n);
 printf("Enter %d non zero element details(row number,column
 number, value) with order...\n",n);
 for(i=0;i<n;i++)
 {
   printf("Enter Element[%d] details.....\n",i+1);
   printf("Enter Row number:");
   scanf("%d",&row[i]);
   printf("Enter column number:");
   scanf("%d",&col[i]);
   printf("Enter value:");
   scanf("%d",&val[i]);
 }
 printf("You entered sparse matrix...\n");
 printf("Element number\trow number\tcolumn number\tvalue");
 for(i=0;i<n;i++)
 { printf("\n%d\t\t\t\t%d\t\t\t%d\t\t\t\t%d",i+1,row[i],col[i],val[i]);
 }
 printf("\nYour sparse matrix in matrix form......\n");
 for(i=0,k=0;i<r;i++)
 {
   for(j=0;j<c;j++)
   {
     if((i==row[k]-1)&&(j==col[k]-1))
     {
       printf("%d\t",val[k]);
       k++;
     }
     else
     {
       printf("0\t");
     }
   }
   printf("\n");
 }
  return 0;
}
```

	row
0	1
1	3

col
1
1

val
1
2

```
r=3
c=3
n=2
i=0, 1, 2
k=0, 1, 2
j=0, 1
i==row[k]-1 && j==col[k] – 1
2==3 – 1 && 0==1 -1
2==2 && 0==0
1         0         0
0         0         0
2
```

Output

```
Enter how many rows in your sparse matrix:3

Enter how many columns in your sparse matrix:3

Enter how many non - zero elements in your sparse matrix:2

Enter 2 non zero element details(row number,column number, value) with
order...

Enter Elemment[1] details.....

Enter Row number:1

Enter column number:1

Enter value:2

Enter Elemment[2] details.....

Enter Row number:3

Enter column number:2

Enter value:3

You entered sparse matrix...

Element number  row number  column number   value

1               1           1               2

2               3           2               3

Your sparse matrix in matrix form......

2   0   0

0   0   0

0   3   0
```

spatran

```c
#include <stdio.h>

int main(void) {
 int r,c,n,i,j,row[10],col[10],val[10];
 printf("Enter how many rows in your sparse matrix:");
 scanf("%d",&r);
 printf("Enter how many columns in your sparse matrix:");
 scanf("%d",&c);
 printf("Enter how many non - zero elements in your sparse matrix:");
 scanf("%d",&n);
 printf("Enter %d non zero element details(row number,column number, value) with order...\n",n);
 for(i=0;i<n;i++)
 {
  printf("Enter Elemment[%d] details.....\n",i+1);
  printf("Enter Row number:");
  scanf("%d",&row[i]);
  printf("Enter column number:");
  scanf("%d",&col[i]);
  printf("Enter value:");
  scanf("%d",&val[i]);
 }
 printf("You entered sparse matrix...\n");
 printf("Element number\trow number\tcolumn number\tvalue");
 for(i=0;i<n;i++)
 {
  printf("\n%d\t\t\t\t%d\t\t\t%d\t\t\t\t%d",i+1,row[i],col[i],val[i]);
 }
 printf("Transpose in sparse matrix form ......\n");
 printf("Element number\trow number\tcolumn number\tvalue");
 for(i=0;i<n;i++)
 {
  printf("\n%d\t\t\t\t%d\t\t\t%d\t\t\t\t%d",i+1,col[i],row[i],val[i]);
 }
  return 0;
}
```

	row	col	val
0	1	3	1
1	2	1	2

```
R=2
C=3
N=2
You entered sparse matrix...
Number row number column number value
          1         1           3                   1
          2         2           1                   2
Transpose in sparse matrix form ......
i=0, 1, 2
Element number row number column number  value
          1              3                 1        1
           2             1                  2       2
```

Output

```
Enter how many rows in your sparse matrix:3
Enter how many columns in your sparse matrix:3
Enter how many non - zero elements in your sparse matrix:2
Enter 2 non zero element details(row number,column number, value) with order...
Enter Elemment[1] details.....
Enter Row number:1
Enter column number:2
Enter value:3
Enter Elemment[2] details.....
Enter Row number:2
Enter column number:1
Enter value:5
You entered sparse matrix...
Element number  row number  column number   value
1         1      2         3
2         2      1         5
Transpose in sparse matrix form ......
Element number  row number  column number   value
1         2      1         3
2         1      2         5
```

Ssbs

```c
#include <stdio.h>

int main(void) {
 int r,c,n,i,j,row[10],col[10],val[10],sum=0,big=0,small=0;
 printf("Enter how many rows in your sparse matrix:");
 scanf("%d",&r);
 printf("Enter how many columns in your sparse matrix:");
 scanf("%d",&c);
 printf("Enter how many non - zero elements in your sparse matrix:");
 scanf("%d",&n);
 printf("Enter %d non zero element details(row number,column number, value) with order...\n",n);
 for(i=0;i<n;i++)
 {
  printf("Enter Elemment[%d] details.....\n",i+1);
  printf("Enter Row number:");
  scanf("%d",&row[i]);
  printf("Enter column number:");
  scanf("%d",&col[i]);
  printf("Enter value:");
  scanf("%d",&val[i]);
 }
 printf("You entered sparse matrix...\n");
 printf("Element number\trow number\tcolumn number\tvalue");
 for(i=0;i<n;i++)
 {
  printf("\n%d\t\t\t\t%d\t\t\t%d\t\t\t\t%d",i+1,row[i],col[i],val[i]);
 }
 sum=val[0];
 big=val[0];
 small=val[0];
 for(i=1;i<n;i++)
 {
  sum+=val[i];
  if(big<val[i])
  big=val[i];
  if(small>val[i])
  small=val[i];
 }
 printf("\nSum = %d",sum);
 printf("\nBig = %d",big);
 printf("\nSmall = %d",small);
  return 0;
}
```

Output

```
Enter how many rows in your sparse matrix:4

Enter how many columns in your sparse matrix:4
Enter how many non - zero elements in your sparse matrix:3

Enter 3 non zero element details(row number,column number, value) with order...

Enter Elemment[1] details.....

Enter Row number:1

Enter column number:3

Enter value:1

Enter Elemment[2] details.....

Enter Row number:2

Enter column number:2

Enter value:3

Enter Elemment[3] details.....

Enter Row number:4

Enter column number:4

Enter value:2

You entered sparse matrix...

Element number  row number  column number   value

1         1      3         1

2         2      2         3

3         4      4         2

Sum = 6

Big = 3

Small = 1
```

Ssearch

```
#include <stdio.h>

int main(void) {
 int r,c,n,i,j,row[10],col[10],val[10],num,flag=0;
 printf("Enter how many rows in your sparse matrix:");
 scanf("%d",&r);
 printf("Enter how many columns in your sparse matrix:");
 scanf("%d",&c);
 printf("Enter how many non - zero elements in your sparse matrix:");
 scanf("%d",&n);
 printf("Enter %d non zero element details(row number,column number, value) with order...\n",n);
 for(i=0;i<n;i++)
 {
   printf("Enter Elemment[%d] details.....\n",i+1);
   printf("Enter Row number:");
   scanf("%d",&row[i]);
   printf("Enter column number:");
   scanf("%d",&col[i]);
   printf("Enter value:");
   scanf("%d",&val[i]);
 }
 printf("Enter Number to Search:");
 scanf("%d",&num);
 printf("You entered sparse matrix...\n");
 printf("Element number\trow number\tcolumn number\tvalue");
 for(i=0;i<n;i++)
 {
   printf("\n%d\t\t\t\t%d\t\t\t%d\t\t\t\t%d",i+1,row[i],col[i],val[i]);
 }
 for(i=0;i<n;i++)
 {
   if(num==val[i])
    {
      flag=1;
      printf("\n%d is found at position [%d][%d]:",num,row[i],col[i]);
    }
 }
 if(flag==0)
 {
   printf("\n%d is not found",num);
 }
  return 0;
}
```

Output

```
Enter how many rows in your sparse matrix:3

Enter how many columns in your sparse matrix:3
Enter how many non - zero elements in your sparse matrix:1

Enter 1 non zero element details(row number,column number, value) with
order...

Enter Elemment[1] details.....

Enter Row number:2

Enter column number:3

Enter value:6

Enter Number to Search:3

You entered sparse matrix...

Element number  row number  column number   value

1               2           3               6

3 is not found
```

link

```c
#include <stdio.h>
#include <malloc.h>
struct node
{
  int data;
  struct node *next;
}*start=NULL;
int main(void) {
  struct node *first_node,*second_node,*third_node;
  first_node=(struct node *)malloc(sizeof(struct node));
  second_node=(struct node *)malloc(sizeof(struct node));
  third_node=(struct node *)malloc(sizeof(struct node));
  printf("Enter the node data:");
  scanf("%d",&first_node->data);
  first_node->next=NULL;
  start=first_node;
  printf("Enter second node data:");
  scanf("%d",&second_node->data);
  second_node->next=NULL;
  first_node->next=second_node;
  printf("Enter third node data:");
  scanf("%d",&third_node->data);
  third_node->next=NULL;
  second_node->next=third_node;
  return 0;
}
```

Output

```
Enter the node data:5

Enter second node data:3

Enter third node data:8
```

travlink

```
#include <stdio.h>
#include <malloc.h>
struct node
{
  int data;
  struct node *next;
}*start=NULL;
int main(void) {
  struct node *first_node,*second_node,*third_node,*fourth_node,*ptr;
  first_node=(struct node *)malloc(sizeof(struct node));
  second_node=(struct node *)malloc(sizeof(struct node));
  third_node=(struct node *)malloc(sizeof(struct node));
  fourth_node=(struct node *)malloc(sizeof(struct node));
  printf("Enter the node data:");
  scanf("%d",&first_node->data);
  first_node->next=NULL;
  start=first_node;
  printf("Enter second node data:");
  scanf("%d",&second_node->data);
  second_node->next=NULL;
  first_node->next=second_node;
  printf("Enter third node data:");
  scanf("%d",&third_node->data);
  third_node->next=NULL;
  second_node->next=third_node;
  printf("Enter fourth node data:");
  scanf("%d",&fourth_node->data);
  fourth_node->next=NULL;
  third_node->next=fourth_node;
  printf("Your linked list traversing...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf(" - >%d ",ptr->data);
    ptr=ptr->next;
  }
  return 0;
}
```

Output

```
Enter the node data:7

Enter second node data:3

Enter third node data:4

Enter fourth node data:8

Your linked list traversing...

-> 7-> 3-> 4-> 8
```

linkinsert

```
#include <stdio.h>
#include <malloc.h>
struct node
{
  int data;
  struct node *next;
}*start=NULL;
int main(void) {
  struct node *first_node,*second_node,*third_node,*fourth_node,*ptr,*new_node;
  first_node=(struct node *)malloc(sizeof(struct node));
  second_node=(struct node *)malloc(sizeof(struct node));
  third_node=(struct node *)malloc(sizeof(struct node));
  fourth_node=(struct node *)malloc(sizeof(struct node));
  new_node=(struct node *)malloc(sizeof(struct node));
  printf("Enter the node data:");
  scanf("%d",&first_node->data);
  first_node->next=NULL;
  start=first_node;
  printf("Enter second node data:");
  scanf("%d",&second_node->data);
  second_node->next=NULL;
  first_node->next=second_node;
  printf("Enter third node data:");
  scanf("%d",&third_node->data);
  third_node->next=NULL;
  second_node->next=third_node;
  printf("Enter fourth node data:");
  scanf("%d",&fourth_node->data);
  fourth_node->next=NULL;
  third_node->next=fourth_node;
  printf("Your linked list traversing before insert...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  printf("\nEnter new  node data to insert at beginning:");
  scanf("%d",&new_node->data);
  new_node->next=start;
  start=new_node;
  printf("Your linked list traversing after insert...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  return 0;
}
```

Output

```
Enter the node data:5
Enter second node data:3
Enter third node data:7
Enter fourth node data:4
Your linked list traversing before insert...
-> 5-> 3-> 7-> 4
Enter new  node data to insert at beginning:8
Your linked list traversing after insert...
-> 8-> 5-> 3-> 7-> 4
```

linkinsertbeg

```
#include <stdio.h>
#include <malloc.h>
struct node
{
  int data;
  struct node *next;
}*start=NULL;
int main(void) {
  struct node *new_node,*ptr,*k;
  int flag=1;
  new_node=(struct node *)malloc(sizeof(struct node));
  printf("Enter the node data:");
  scanf("%d",&new_node->data);
  new_node->next=NULL;
  if(start==NULL)
   {
    start=new_node;
   }
  printf("Your linked list traversing...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  while(flag)
  {
   printf("\nEnter data:");
   k=(struct node *)malloc(sizeof(struct node));
   scanf("%d",&k->data);
   k->next=start;
   start=k;
   printf("Your linked list traversing...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  printf("\nDo you want to continue if yes enter 1 otherwise 0:");
  scanf("%d",&flag);
  }
  return 0;
}
```

Output

```
nter the node data:5
Your linked list traversing...
-> 5
Enter data:3
Your linked list traversing...
-> 3-> 5
Do you want to continue if yes enter 1 otherwise 0:1

Enter data:7
Your linked list traversing...
-> 7-> 3-> 5
Do you want to continue if yes enter 1 otherwise 0:1

Enter data:8
Your linked list traversing...
-> 8-> 7-> 3-> 5
Do you want to continue if yes enter 1 otherwise 0:0
```

linkinsertend

```
#include <stdio.h>
#include <malloc.h>
struct node
{
  int data;
  struct node *next;
}*start=NULL;
int main(void) {
  struct node *new_node,*ptr,*k,*cur;
  int flag=1;
  new_node=(struct node *)malloc(sizeof(struct node));
  printf("Enter the node data:");
  scanf("%d",&new_node->data);
  new_node->next=NULL;
  if(start==NULL)
   {
    start=new_node;
   }
  printf("Your linked list traversing...\n");
  ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  while(flag)
  {
   printf("\nEnter data:");
   k=(struct node *)malloc(sizeof(struct node));
   scanf("%d",&k->data);
   k->next=NULL;
   cur=start;
   while(cur->next!=NULL)
   {
     cur=cur->next;
   }
   cur->next=k;
   printf("Your linked list traversing...\n");
   ptr=start;
  while(ptr!=NULL)
  {
    printf("-> %d",ptr->data);
    ptr=ptr->next;
  }
  printf("\nDo you want to continue if yes enter 1 otherwise 0:");
  scanf("%d",&flag);
  }
  return 0;
}
```

Output

```
Enter the node data:6
Your linked list traversing...
-> 6
Enter data:3
Your linked list traversing...
-> 6-> 3
Do you want to continue if yes enter 1 otherwise 0:1

Enter data:8
Your linked list traversing...
-> 6-> 3-> 8
Do you want to continue if yes enter 1 otherwise 0:1

Enter data:2
Your linked list traversing...
-> 6-> 3-> 8-> 2
Do you want to continue if yes enter 1 otherwise 0:0
```