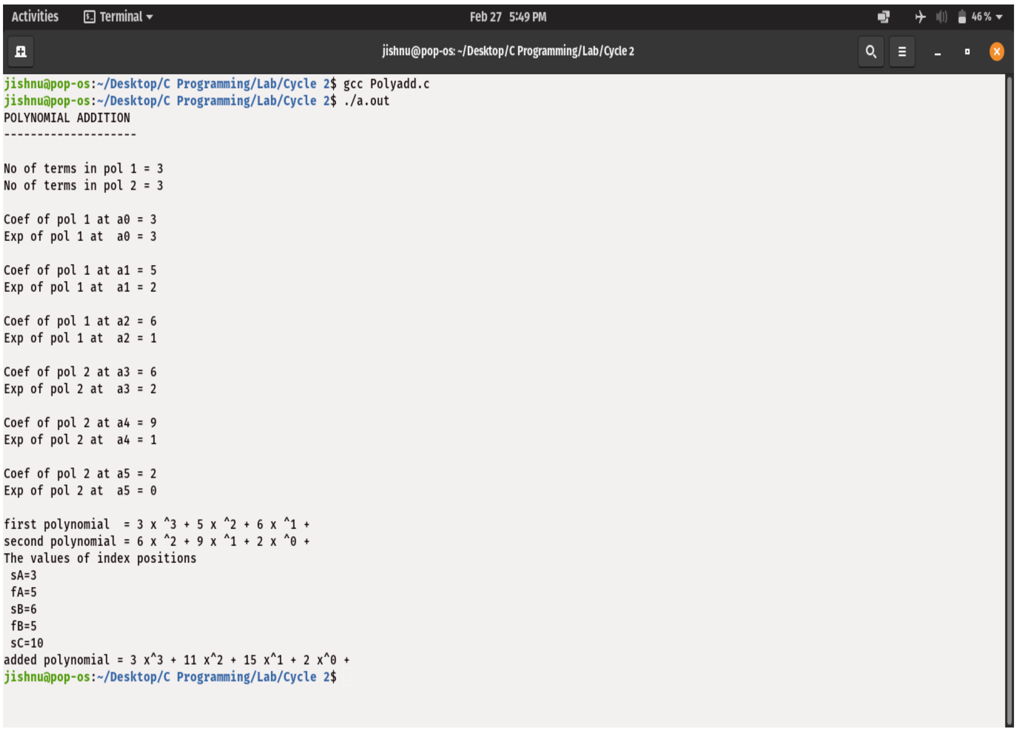
(1) POLYNOMIAL ADDITION

```
#include<stdio.h>
#define max 100
typedef struct pol{
	int coef;
	int exp;
}pol;
pol A[max];
void main(){
	int sA=0,sB,sC,fA,fB;
	int i,c,p,q;
	printf("POLYNOMIAL ADDITION\n");
	for(i=0;i<20;i++){
		printf("%c",'-');
	}printf("\n\n");
	printf("No of terms in pol 1 = ");
	scanf("%d",&p);
	printf("No of terms in pol 2 = ");
	scanf("%d",&q);
	printf("\n");
	sA=0;
	fA=p-1;
	sB=p;
	fB=p+q-1;
	sC=p+q;
	for(i=0;i<p;i++){
		printf("Coef of pol 1 at a%d = ",i);
		scanf("%d",&A[i].coef);
		printf("Exp of pol 1 at  a%d = ",i);
		scanf("%d",&A[i].exp);
		printf("\n");
	}
	for(i=p;i<p+q;i++){
		printf("Coef of pol 2 at a%d = ",i);
		scanf("%d",&A[i].coef);
		printf("Exp of pol 2 at  a%d = ",i);
		scanf("%d",&A[i].exp);
		printf("\n");
	}

	while(sA<=fA && sB<=fB){
		if(A[sA].exp > A[sB].exp){
			A[sC].exp = A[sA].exp;
			A[sC].coef = A[sA].coef;
			sA++;
			sC++;
		}
		else if(A[sA].exp < A[sB].exp){
			A[sC].exp = A[sB].exp;
			A[sC].coef = A[sB].coef;
			sB++;
			sC++;
		}else{
			c = A[sA].coef + A[sB].coef;
			if(c != 0){
				A[sC].exp = A[sB].exp;
				A[sC].coef = c;
				sC++;
			}
			sA++;
```

```
                        sB++;
                }
        }
        while(sA<=fA){
                A[sC].exp = A[sA].exp;
                A[sC].coef = A[sA].coef;
                sA++;
                sC++;
        }
        while(sB<=fB){
                A[sC].exp = A[sB].exp;
                A[sC].coef = A[sB].coef;
                sB++;
                sC++;
        }

        printf("first polynomial  = ");
        for(i=0;i<p;i++){
                printf("%d x ^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
        printf("second polynomial = ");
        for(i=p;i<p+q;i++){
                printf("%d x ^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
        printf("The values of index positions\n sA=%d\n fA=%d\n sB=%d\n fB=%d\n sC=%d\
n",sA,fB,sB,fB,sC);
        printf("added polynomial = ");
        for(i=p+q;i<sC;i++){
                printf("%d x^%d + ",A[i].coef, A[i].exp);
        }
        printf("\n");
}
```

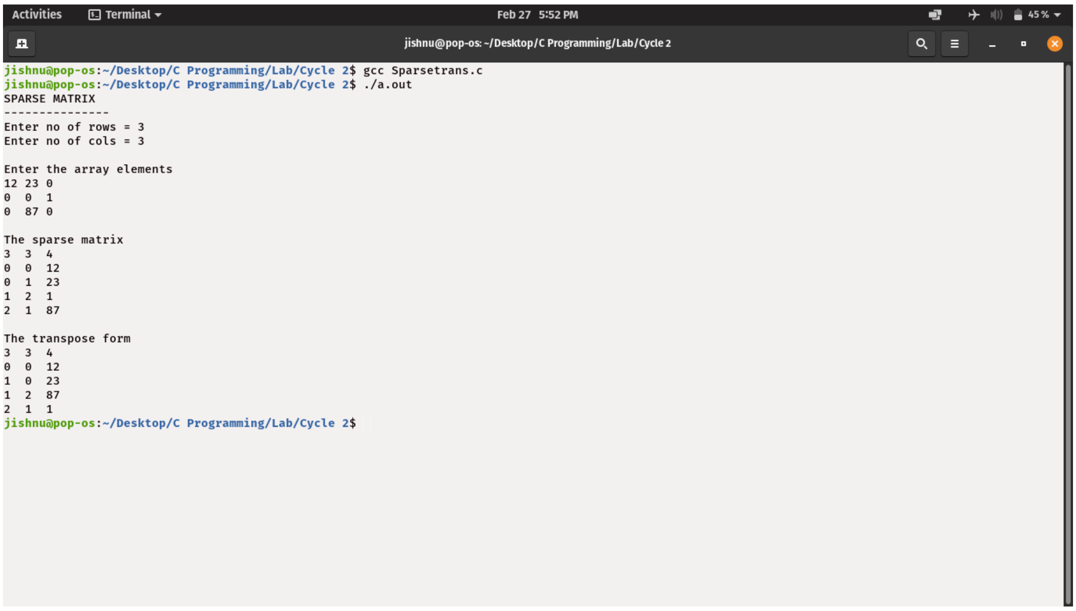


```
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$ gcc Polyadd.c
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$ ./a.out
POLYNOMIAL ADDITION
--------------------

No of terms in pol 1 = 3
No of terms in pol 2 = 3

Coef of pol 1 at a0 = 3
Exp of pol 1 at  a0 = 3

Coef of pol 1 at a1 = 5
Exp of pol 1 at  a1 = 2

Coef of pol 1 at a2 = 6
Exp of pol 1 at  a2 = 1

Coef of pol 2 at a3 = 6
Exp of pol 2 at  a3 = 2

Coef of pol 2 at a4 = 9
Exp of pol 2 at  a4 = 1

Coef of pol 2 at a5 = 2
Exp of pol 2 at  a5 = 0

first polynomial  = 3 x ^3 + 5 x ^2 + 6 x ^1 +
second polynomial = 6 x ^2 + 9 x ^1 + 2 x ^0 +
The values of index positions
 sA=3
 fA=5
 sB=6
 fB=5
 sC=10
added polynomial = 3 x^3 + 11 x^2 + 15 x^1 + 2 x^0 +
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$
```

(2)SPARSE TRANSPOSE

```
#include<stdio.h>
#define max 100
typedef struct{
		int row;
		int col;
		int value;
}sparse;
sparse A[max];
sparse B[max];
void main(){
		int a[100][100],i,j,r,c,k=1,p=1;
		printf("SPARSE MATRIX\n");
		for(int i=0;i<15;i++){
				printf("%c",'-');
		}printf("\n");
		printf("Enter no of rows = ");
		scanf("%d",&r);
		printf("Enter no of cols = ");
		scanf("%d",&c);
		printf("\nEnter the array elements\n");
		for(i=0;i<r;i++){
				for(j=0;j<c;j++){
						scanf("%d",&a[i][j]);
				}
		}
		A[0].row = r;
		A[0].col = c;
		for(i=0;i<r;i++){
				for(j=0;j<c;j++){
						if(a[i][j] != 0){
								A[k].row = i;
								A[k].col = j;
								A[k].value = a[i][j];
								k++;
						}
				}
		}
		A[0].value = k-1;
		printf("\nThe sparse matrix\n");
		for(i=0;i<k;i++){
				printf("%d  %d  %d  \n",A[i].row, A[i].col, A[i].value);
		}
		B[0].row = A[0].col;
		B[0].col = A[0].row;
		B[0].value = A[0].value;
		for(i=0;i<=A[0].col;i++){
				for(j=1;j<=A[0].value;j++){
						if( A[j].col == i){
								B[p].col = A[j].row;
								B[p].row = A[j].col;
								B[p].value = A[j].value;
								p++;
						}
				}
		}
		printf("\nThe transpose form\n");
		for(i=0;i<=B[0].value;i++){
				printf("%d  %d  %d  \n",B[i].row,B[i].col,B[i].value);
```

```
        }
}
```

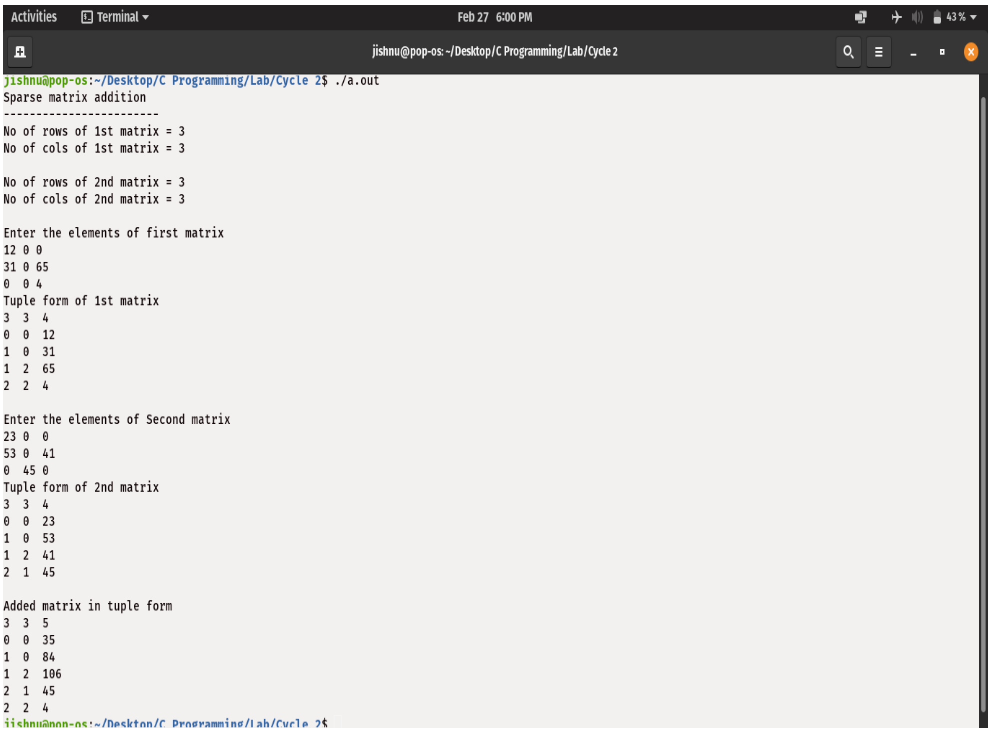


```
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$ gcc Sparsetrans.c
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$ ./a.out
SPARSE MATRIX
---------------
Enter no of rows = 3
Enter no of cols = 3

Enter the array elements
12 23 0
0  0  1
0  87 0

The sparse matrix
3  3  4
0  0  12
0  1  23
1  2  1
2  1  87

The transpose form
3  3  4
0  0  12
1  0  23
1  2  87
2  1  1
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$
```

SPARSE MATRIX ADDITION

```
#include<stdlib.h>
#include <stdio.h>
#define max 100
typedef struct
{
        int row ;
        int col ;
        int value ;
}sparce;
sparce A[max],B[max],C[max];
void main()
{
        int i,j,r1,r2,c1,c2,x,n=1,m=1,sum=0,k=1,c=0;
        printf("Sparse matrix addition\n");
        for(int i=0;i<24;i++){
                printf("%c",'-');
        }printf("\n");
        printf("No of rows of 1st matrix = ");
        scanf("%d",&r1);
        printf("No of cols of 1st matrix = ");
        scanf("%d",&c1);
        A[0].row=r1;
        A[0].col=c1;
        printf("\nNo of rows of 2nd matrix = ");
        scanf("%d",&r2);
        printf("No of cols of 2nd matrix = ");
        scanf("%d",&c2);
        B[0].row=r2;
        B[0].col=c2;
        if(r1!=r2 || c1!=c2){
                printf("Matrix addition not poosible !!!\nexiting the program\n!");
                exit(0);
        }
```

```
printf("\nEnter the elements of first matrix\n");
for (i=0;i<r1;i++){
        for (j=0;j<c1;j++){
                scanf("%d",&x);
                if(x!=0){
                        A[m].row=i;
                        A[m].col=j;
                        A[m].value=x;
                        m++;
                }
        }
}
A[0].value=m-1;
printf("Tuple form of 1st matrix\n");
for(i=0;i<m;i++){
        printf("%d  %d  %d \n",A[i].row,A[i].col,A[i].value);
}
printf("\nEnter the elements of Second matrix\n");
for (i=0;i<r2;i++){
        for (j=0;j<c2;j++){
                scanf("%d",&x);
                if(x!=0){
                        B[n].row=i;
                        B[n].col=j;
                        B[n].value=x;
                        n++;
                }
        }
}
B[0].value=n-1;
printf("Tuple form of 2nd matrix\n");
for(j=0;j<n;j++){
        printf("%d  %d  %d \n",B[j].row,B[j].col,B[j].value);
}
i=1;
j=1;
int p=A[0].value;
int q=B[0].value;
while(i<=p && j<=q){
        if(A[i].row<B[j].row || A[i].col<B[j].col){
                C[k].col=A[i].col;
                C[k].row=A[i].row;
                C[k].value=A[i].value;
                i++;
                k++;
                c++;
        }
        else if(A[i].row>B[j].row || A[i].col>B[j].col){
                C[k].col=B[j].col;
                C[k].row=B[j].row;
                C[k].value=B[j].value;
                j++;
                k++;
                c++;
        }
        else{
                sum=A[i].value + B[j].value;
                if(sum!=0){
                        C[k].col=A[i].col;
                C[k].row=A[i].row;
                C[k].value=sum;
                i++;
                k++;
                c++;
                j++;
            }
            else{
                i++;
                j++;
            }
        }
    }
    while(i<=p){
        C[k].col=A[i].col;
        C[k].row=A[i].row;
        C[k].value=A[i].value;
        i++;
        k++;
        c++;
    }
    while(j<=q){
        C[k].col=B[j].col;
        C[k].row=B[j].row;
        C[k].value=B[j].value;
        j++;
        k++;
        c++;
    }
    C[0].value=c;
    C[0].row=r1;
    C[0].col=c1;
    printf("\nAdded matrix in tuple form\n");
    for(i=0;i<k;i++){
        printf("%d  %d  %d \n",C[i].row,C[i].col,C[i].value);
    }
}
```



```
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$ ./a.out
Sparse matrix addition
-----------------------
No of rows of 1st matrix = 3
No of cols of 1st matrix = 3

No of rows of 2nd matrix = 3
No of cols of 2nd matrix = 3

Enter the elements of first matrix
12 0 0
31 0 65
0  0 4
Tuple form of 1st matrix
3  3  4
0  0  12
1  0  31
1  2  65
2  2  4

Enter the elements of Second matrix
23 0  0
53 0  41
0  45 0
Tuple form of 2nd matrix
3  3  4
0  0  23
1  0  53
1  2  41
2  1  45

Added matrix in tuple form
3  3  5
0  0  35
1  0  84
1  2  106
2  1  45
2  2  4
jishnu@pop-os:~/Desktop/C Programming/Lab/Cycle 2$
```