

DS Lab 3 Solutions

3.1.1

```
1  #include <stdio.h>
2  v int main(){
3      —>int m,n,i,j,mat[30][30];
4      —>printf("Enter the size of matrix (rows, columns): ");
5      —>scanf("%d%d",&m,&n);
6      —>printf("Enter elements of matrix\n");
7  v —>for(i=0;i<m;i++){
8  v —> —>for(j=0;j<n;j++){
9      —> —> —>printf("[%d][%d]: ",i,j);
10     —> —> —>scanf("%d",&mat[i][j]);
11     —> —>}
12     —>}
13     —>int size=0;
14  v —>for(i=0;i<m;i++){
15  v —> —>for(j=0;j<n;j++){
16  v —> —> —>if (mat[i][j]!=0){
17     —> —> —> —>size++;
18     —> —> —>}
19     —> —>}
20     —>}
21     —>int sparseMatrix[10][3];
22     —>int k=0;
23  v —>for(i=0;i<m;i++){
24  v —> —>for(j=0;j<n;j++){
```

```

25  v  —> —> —> —> if(mat[i][j]!=0){
26      —> —> —> —> sparseMatrix[k][0]=i;
27      —> —> —> —> sparseMatrix[k][1]=j;
28      —> —> —> —> sparseMatrix[k][2]=mat[i][j];
29      —> —> —> —> k++;
30      —> —> —> }
31      —> —> }
32      —> }
33      —> printf("Sparse form -- list of 3 triples\n");
34      —> printf("%d\t%d\t%d",m,n,k);
35      —> printf("\n");
36  v  —> for(i=0;i<size;i++){
37  v  —> —> for(j=0;j<3;j++){
38      —> —> —> printf("%d",sparseMatrix[i][j]);
39      —> —> —> if(j!=2)
40          —> —> —> printf("\t");
41          —> —> —> else
42              —> —> —> printf("\n");
43          —> —> }
44      —> }
45      }

```

3.1.2

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  typedef int it;
4  v it main(){
5      —>it n,i;
6      —>it *parr;
7      —>parr=(int*)malloc(n*sizeof(int));
8      —>printf("Enter number of elements:\n");
9      —>scanf("%d",&n);
10     —>printf("Enter %d elements into array:",n);
11     v —>for(i=0;i<n;i++){
12         —>—>scanf("%d",parr+i);
13     —>}
14     —>printf("Elements of array are as follow:\n");
15     v —>for(i=0;i<n;i++){
16         —>—>printf("%d ",*(parr+i));
17     —>}
18     —>printf("\n");
19
20
21     }

```

3.1.3

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  typedef int it;
4  v it main(){
5      —> it arr[30][30], arr2[30][30];
6      —>
7      —> it m, n, **parr, **parr2, i, j;
8      —> printf("Enter the order of the matrix 1 & 2: \n");
9      —> scanf("%d%d", &m, &n);
10     —> parr=(it**)malloc(n*sizeof(it*));
11     v —> for(i=0; i<m; i++){
12         —> —> parr[i]=(it*)malloc(n*sizeof(it));
13         —> }
14     —> parr2=(it**)malloc(n*sizeof(it*));
15     v —> for(i=0; i<m; i++){
16         —> —> parr2[i]=(it*)malloc(n*sizeof(it));
17         —> }
18     —> //arr=parr[i][j];
19     —> printf("Enter values into 2D array of rows=%d,
        cols=%d\n", m, n);
20     v —> for(i=0; i<m; i++){
21         v —> —> for(j=0; j<n; j++){
22             —> —> —> scanf("%d", (*(parr+i)+j));
23             —> —> }

```

```

23     —> —> }
24     —> }
25     —> printf("Enter values into 2D array of rows=%d,
        cols=%d\n",m,n);
26 v —> for(i=0;i<m;i++){
27 v —> —> for(j=0;j<n;j++){
28     —> —> —> scanf("%d",(*(parr2+i)+j));
29     —> —> }
30     —> }
31     —> printf("Matrix-1 is as follow:\n");
32 v —> for(i=0;i<m;i++){
33 v —> —> for(j=0;j<n;j++){
34     —> —> —> printf("%d ",(parr)[i][j]);
35     —> —> }
36     —> —> printf("\n");
37     —> }
38     —> printf("Matrix-2 is as follow:\n");
39 v —> for(i=0;i<m;i++){
40 v —> —> for(j=0;j<n;j++){
41     —> —> —> printf("%d ",(parr2)[i][j]);
42     —> —> }
43     —> —> printf("\n");
44     —> }
45     —> it.**sum;

```

```

46     —>sum=(it**)malloc(n*sizeof(it*));
47 v —>for(i=0;i<m;i++){
48     —>—>sum[i]=(it**)malloc(n*sizeof(it*));
49     —>}
50     —>printf("Resultant Sum of 2 Matrices is as follow:\n");
51 v —>for(i=0;i<m;i++){
52 v —>—>for(j=0;j<n;j++){
53     —>—>—>(sum)[i][j]=(parr)[i][j]+(parr2)[i][j];
54     —>—>—>printf("%d ",(sum)[i][j]);
55     —>—>}
56     —>—>printf("\n");
57     —>}
58     —>it.**pro;
59     —>pro=(it**)malloc(n*sizeof(it*));
60 v —>for(i=0;i<m;i++){
61     —>—>pro[i]=(it**)malloc(n*sizeof(it*));
62     —>}
63     —>printf("Resultant difference of 2 Matrices is as follow:\n");
64 v —>for(i=0;i<m;i++){
65 v —>—>for(j=0;j<n;j++){
66     —>—>—>(pro)[i][j]=(parr)[i][j]-(parr2)[i][j];
67     —>—>—>printf("%d ",(pro)[i][j]);
68     —>—>}
69     —>—>printf("\n");
70     —>}
71 }

```

3.2.1

```

1  #include<stdio.h>
2  #include<conio.h>
3  #include<string.h>
4  #include<math.h>
5  int
6  k=1,i,j,m,num,n,p[10][10],q[10][10],r[10][10],ct=0,R=1,C,r1,r2,c1,c2;
7  void read(int a[10][10],int r,int c)
8  v {
9      —>k=1;
10     —>R=1;
11     —>ct=0;
12     —>printf("Enter the matrix:\n");
13     —>for(i=0;i<r;i++)
14     v —>{
15         —>—>for(j=0;j<c;j++)
16         v —>—>{
17             —>—>—>scanf("%d",&num);
18             —>—>—>if(num!=0)
19             v —>—>—>{
20                 —>—>—>—>a[R][C]=i;
21                 —>—>—>—>a[R][++C]=j;
22                 —>—>—>—>a[R][++C]=num;
23                 —>—>—>—>ct++;
24                 —>—>—>—>R++;
25                 —>—>—>—>C=0;
26                 —>—>—>}
27             —>—>}
28     —>}

```

```

29     └─> a[0][0]=r;
30     └─> a[0][1]=c;
31     └─> a[0][2]=ct;
32     └─> printf("Sparse Matrix in memory:\n");
33     └─> for(i=0;i<=ct;i++)
34 v └─> {
35     └─> └─> for(j=0;j<3;j++)
36     └─> └─> └─> printf("%d ",a[i][j]);
37
38     └─> └─> printf("\n");
39     └─> }
40 }
41
42 void add()
43 v {
44     ....printf("Enter the row & coloumn of 1st matrix: ");
45     ....scanf("%d %d",&r1,&c1);
46     ....printf("Enter the row & coloumn of 2nd matrix: ");
47     ....scanf("%d %d",&r2,&c2);
48     ....if(r1==r2&&c1==c2)
49 v ....{
50     ....read(p,r1,c1);
51     ....read(q,r2,c2);
52     └─> └─> k=1;
53     └─> └─> printf("Sum Matrix:\n");
54     └─> └─> for(i=0;i<r1;i++)
55 v └─> └─> {
56     └─> └─> └─> for(j=0;j<c1;j++)

```



```

85  v .....{
86      —>—>—>...for(j=0;j<c1;j++)
87  v .....{
88      .....if(p[k][0]==i&& p[k][1]==j)
89  v .....{
90      .....q[j][i]=p[k][2];
91      —>—>—>—>—>k++;
92      —>—>—>—>—>
93  }
94
95      —>—>—>—>else
96      —>—>—>—>—>q[j][i]=0;
97      —>—>—>—>
98      —>—>—>—>}
99      —>—>—>—>
100     —>—>—>}
101     —>—>—>for(i=0;i<c1;i++)
102
103  v —>—>—>{
104      —>—>—>—>for(j=0;j<r1;j++)
105
106  v —>—>—>—>{
107
108      —>—>—>—>—>printf("·%d·",q[i][j]);
109      —>—>—>—>—>
110      —>—>—>—>}
111
112     —>—>—>—>printf("\n");

```

```

113     —> —> —>
114     —> —> }
115     —> —>
116     —> }
117
118     —> —> else
119
120     —> —> printf("Addition not possible!");
121     }
122
123     —> —> void main()
124
125     v {
126     —> char ch;
127     —> fflush(stdin);
128     —> ct=0;
129     —> fflush(stdin);
130     —> fflush(stdout);
131     —> add();
132     }

```

3,2,2

```

1     #include <stdio.h>
2     v int main(){
3         —> int i,j,m,n,count=0,x,y;
4         —> float k;
5         —> printf("Enter the number of rows and columns of the matrix:");
6         —> scanf("%d%d",&m,&n);
7         —> int a[m][n];
8         —> printf("Enter the elements of the matrix:");
9         v —> for(i=0;i<m;i++){
10        v —> —> for(j=0;j<n;j++){
11            —> —> —> scanf("%d",&a[i][j]);
12        v —> —> —> if(a[i][j]!=0){
13            —> —> —> —> count++;
14            —> —> —> }
15        —> —> }
16        —> }
17        —> x=4*m*n;
18        —> y=count*3*4;
19        —> k=(float)(x-y)/x*100;
20        —> printf("Size of the matrix = %d bytes\n",x);
21        —> printf("Size of the sparse matrix = %d bytes\n",y);
22        —> printf("Memory saved with sparse matrix representation = %.2f%%",k);
23        —> printf("\n");
24    }

```

3.2.3

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  v void main(){
4      —>int **a,m,n,i,j,count=0;
5      —>printf("Enter the dimensions of the matrix:");
6      —>scanf("%d%d",&m,&n);
7      —>printf("Enter the elements of the matrix:");
8      —>a=(int*)malloc(m*sizeof(int));
9  v —>for(i=0;i<m;i++){
10     —>—>a[i]=(int*)malloc(n*sizeof(int));
11  v —>—>for(j=0;j<n;j++){
12     —>—>—>scanf("%d",&a[i][j]);
13     —>—>—>if(a[i][j]!=0)
14     —>—>—>count++;
15     —>—>}
16     —>}
17     —>printf("Sparse Matrix Representation:\n");
18     —>printf("%d %d %d\n",m,n,count);
19  v —>for(i=0;i<m;i++){
20  v —>—>for(j=0;j<n;j++){
21  v —>—>—>if(a[i][j]!=0){
22     —>—>—>—>printf("%d %d %d\n",i,j,a[i][j]);
23     —>—>—>}
24     —>—>}
25     —>}
26 }
27
```

3.2.4

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  typedef int it;
4  v int main(){
5      —>it n,*parr,i;
6      —>printf("Enter the size of the array: ");
7      —>scanf("%d",&n);
8      —>parr=(it*)calloc(n,sizeof(it));
9      —>printf("Enter the elements of the array:\n");
10     v —>for(i=0;i<n;i++){
11         —>—>scanf("%d",parr+i);
12     —>}
13     —>printf("The elements of the array are:\n");
14     v —>for(i=0;i<n;i++){
15         —>—>printf("%d ",*(parr+i));
16     —>}
17     —>free(parr);
18 }

```

3.2.5

```

1  #include<stdio.h>
2  v int read(int a[10][10][10],int m,int n,int k){
3      —>int i,j,l;
4      v —>for(i=0;i<m;i++){
5          v —>—>for(j=0;j<n;j++){
6              v —>—>—>for(l=0;l<k;l++){
7                  —>—>—>scanf("%d",&a[i][j][l]);
8              —>—>—>}
9          —>—>}
10     —>}
11 }
12 v int main(){
13     —>int a[10][10][10],b[10][10][10],i,j,l,m,n,k;
14     —>printf("Enter the dimensions of the array: ");
15     —>scanf("%d%d%d",&m,&n,&k);
16     —>printf("Enter the elements of the first array:\n");
17     —>read(a,m,n,k);
18     —>printf("Enter the elements of the second array:\n");
19     —>read(b,m,n,k);
20     —>printf("Result of addition:\n");
21     v —>for(i=0;i<m;i++){
22         —>—>printf("Array %d:\n",i+1);
23         v —>—>for(j=0;j<n;j++){
24             v —>—>—>for(l=0;l<k;l++){
25                 —>—>—>printf("%d ",a[i][j][l]+b[i][j][l]);
26             —>—>—>}
27         —>—>printf("\n");
28     —>—>}
29     —>}
30 }

```