Experiment No 2
Problem Statements

1.Write a program to identify if the entered matrix is sparse or not.

```
//1) Write a program to identify if the entered matrix is sparse or not
/*Name:-Anuj Rajendra Mane
Roll:-65
Div:-A
Subject:- Data Structures*/
#include<stdio.h>
int main()
{
    int matrix[4][4];/*= {{1,0,0,3},
                     {0,0,0,9},
                     {7,0,0,0},
                     {0,0,6,0}};*/


    int count = 0;


    printf("Enter 4 by 4 matrix elements to chexk given matrix is sparse or not:\n");
    for(int i=0;i<4;i++)
    {
        for(int j=0;j<4;j++)
        {
            scanf("%d",&matrix[i][j]);
        }
    }
    for(int i=0;i<4;i++)
    {
        for(int j=0;j<4;j++)
        {
            printf("%d ",matrix[i][j]);
        }
        printf("\n");
    }
    for(int i=0;i<4;i++)
    {
        for(int j=0;j<4;j++)
        {
            if(matrix[i][j] == 0 )
            {
                count++;
            }
        }
    }
    if(count>16/2)
    {
        printf("\nSparse");
    }
    else {
       printf("\nNot");
    }
}
```

2.Write a Program to present a sparse matrix into array representation (row, column, value).

```
//2) Write a Program to present a sparse matrix into array representation (row, column, value)
/*/*Name:-Anuj Rajendra Mane
Roll:-65
Div:-A
Subject:- Data Structures*/
#include<stdio.h>
int main() {

    int Sparse[4][4] = {{0,0,2,3},
                        {3,0,0,0},
                        {0,0,0,0},
                        {9,3,0,0}};

    int size = 0;
    for(int i=0;i<4;i++)
    {
       for(int j=0;j<4;j++)
       {
          if(Sparse[i][j]!=0)
          {
             size++;
          }
       }
    }
    int compact[3][size];

    int k = 0;
    for(int i=0;i<4;i++)
    {
       for(int j=0;j<4;j++)
       {
          if(Sparse[i][j]!=0)
          {
             compact[0][k] = i;
             compact[1][k] = j;
             compact[2][k] = Sparse[i][j];
             k++;
          }
       }
    }
```

```
   for(int i=0;i<3;i++)
   {
      for(int j=0;j<size;j++)
      {
         printf("%d ",compact[i][j]);
      }
      printf("\n");
   }
}
```

3. Write a program to implement hashing using arrays.

```
//3) Write a program to implement hashing using arrays.
/*Name:- Anuj Rajendra Mane
ROll No:- 65
Div:-A
Subject:- Data Structures*/

#include<stdio.h>
int main()
{
   int key[50],size,hash[50];
   printf("Enter Size of hash table:\n");
   scanf("%d",&size);
   for(int i=0;i<=size;i++)
   {
      printf("Enter Key: %d\n",i+1);
      scanf("%d",&key[i]);
   }
   for(int i=0;i<=key[i];i++)
   {
      hash[i] = key[i] % size;
   }
   printf("Hash Index\tKeys\n");
   for(int i=0;i<=key[i];i++)
   {
      printf("\n%d\t\t%d ",hash[i],key[i]);
   }
}
```

4.Write a program to implement Linear probing for collision resolution.

```
//4.Write a program to implement Linear probing for collision resolution.
/*Name:- Anuj Rajendra Mane
ROll No:- 65
Div:-A
```

```
Subject:- Data Structures*/



#include<stdio.h>
int main()
{
    int key,hashTable[10] = {0,0,0,0,0,0,0,0,0,0};
    int i,loc,size,no_key;

    printf("Enter Size of the hash table:\n");
    scanf("%d",&size);

    printf("Enter how many keys:\n");
    scanf("%d",&no_key);

    for(i=0; i<no_key; i++)
    {
        printf("Enter Keys:\n");
        scanf("%d",&key);

        loc = key % size;
        while(hashTable[loc]!=0)
        {
            loc++;
        }
        hashTable[loc] = key;
    }
    for(i=0; i<size; i++)
    {
        printf("\n%d %d",i,hashTable[i]);
    }
}
```