PROGRAM-03

Arpana M Ramaswamy
1BM18CS147
07/16/2020 Arpana

```
Class DisSets
{
 vector <int> rank, parent;
 int n;

 public:
   DisSets(int n)
  {
   rank.resize(n)
   parent.resize(n);
   this->n=n;
   makeSet();
  }

 void makeset()
 {
  for (i=0; i<n; i++)
    parent[i]=i;
 }
```

Arpana M Ramaswamy
1BM18CS147
07/10/20 Arpana

```
//recursively call to find representative
int find(int x)
{
 if(parent[x]!=x)
 {
   return find(parent[x]);
 }
 return x;
}


void Union(int x,int y)
{
  int xroot=find(x)
  int yroot=find(y)
   //if already in same set
  if(xroot=yroot)
      return;
   //Find lesser rank
  if(rank[xroot]<rank[yroot])
      parent[xroot]=yroot;

  else if(rank[yroot]<rank[xroot])
      parent[yroot]=xroot;

  else
  {
```

Arpana M Ramaswamy
1BM18CS144
Arpana

```
      parent[xroot]=yroot;

      rank[xroot]=rank[xroot]+1;
      }
   }
};


int IslandCount(vector<vector<int>>M)
{
  int n = M.size();
  int m = M.size();
  DisSets *islands = new DisSets(n*m);
 for( i=0; i<n; i++)
 {
  for(j=0; j<m; j++)
  {
   if (M[i][j]==0)
     continue;

   if(i+1<n && M[i+1][j] ==1)
    islands->Union(i*m+j,(i+1)*m+j);

   if(i-1>=0 && M[i-1][j]==1)
   islands->Union(i*m+j,(i-1)*m+j);
```

Arpana MR
1BM18CS147
Arpana

```
    if (j+1<m && M[i][j+1]==1)
        islands -> Union (i*m+j, (i-1)*m+j);

    if (j-1>=0 && M[i][j-1]==1)
        islands-> Union (i*m+j, i*m+j-1);

    if (i+1<n && M[i+1][j+1]==1 && j+1<m)
        islands-> Union (i*m+j, (i+1)*m+j+1);

    if (i+1<n && j-1>=0 && M[i+1][j-1]==1)
        islands->Union (i*m+j, (i+1)*m+j-1);

    if (i-1>=0 && j+1<m && M[i-1][j+1]==1)
        islands-> Union (i*m+j, (i-1)*m+k+1);

    if (i-1>=0 && j-1>=0 && M[i-1][k-1]==1)
        islands-> Union (i*m+j, (i-1)*m+j-1);

  }
}

int *freq= new int[n*m];
int num = 0;
```

Archana MR
1BM18CS147
Archana

```
for(i=0; i<n; i++)
{
 for(j=0; j<m; j++)
 {
  if(M[i][j]==1)
  {
   int x = islands->find(i*m+j);

   if(freq[x]==0)
   {
     num++;
     freq[x]++;
   }

  else
     freq[x]++;
  }
 }
}
~~cout~~return num;
}
```