Manoj.H.A
1BM19CS083
4th sem, 'B' sec
ADA - Lab test 03.
6/07/2021

Warshall's algorithm

```
#include<stdio.h>
int a[10][10], p[10][10], i, j, k, n;
void warshalls()
{
  for(i=1; i<=n; i++)
  {
    for(j=1; j<=n; j++)
    {
      p[i][j] = a[i][j];
    }
  }
  for(k=1; k<=n; k++)
  {
    for(i=1; i<=n; i++)
    {
      for(j=1; j<=n; j++)
      {
        if((p[i][j]!=1) && (p[i][k]==1 && p[k][j]==1))
          p[i][j] = 1;
      }
    }
  }
}
```

```
void main()
{ printf("Enter number of vertices\n");
  scanf("%d", &n);
  printf("Enter adjacency matrix\n");
  for(i=1; i<=n; i++)
  {
    for(j=1; j<=n; j++)
    {
      scanf("%d", &a[i][j]);
    }
  }
  warshalls();
  printf("Path matrix\n");
  for(i=1; i<=n; i++)
  {
    for(j=1; j<=n; j++)
    {
      printf("%d\t", p[i][j]);
    }
    printf("\n");
  }
  for(i=0; i<n; i++)          /*Modification part
                                We have to check if
                                diagonal elements are
                                equal to 1*/
  { if(p[i][i]==1)
    { printf("Cycle exists\n");
      break; count++;          // Modification.
    }
  }
  if(count==0)
  { printf("Graph doesn't contain any cycle");
  }
```

```
else
{ printf ("Graph contains cycle with vertices");
    for (i=1; i<=n ; i++)
     { if (p[i][i] == 1)
       { printf ("%d", i);
       }
     }
  }
  getch();
}
```