```
/*Lab 15: Network flow Calculation*/
#include<stdio.h>
#define NUM_NODES 5
#define SIZE 20
int res_graph[NUM_NODES][NUM_NODES];
int aug_path[NUM_NODES];
int queue[SIZE];
int rear,front;
void insert_queue(int data)
{
        rear=(rear+1)%SIZE;
        queue[rear]=data;
        if(front==-1)
         front=0;
}
int delete_queue()
{
        int data=queue[front];
        if(front==rear)
         front=rear=-1;
        else
         front=(front+1)%SIZE;
        return data;
}
int is_queue_empty()
{
        return front==-1;
}
int find_augmented_path(int s,int t)
{
        int visited[NUM_NODES]={0};
        insert_queue(s);
        int u,v,i;
        visited[s]=1;
        for(i=0;i<NUM_NODES;i++)
         aug_path[i]=-1;
        while(!is_queue_empty())
        {
          u=delete_queue();
          for(v=0;v<NUM_NODES;v++)
          {
                if(visited[v]==0&&res_graph[u][v]>0)
                {
                        aug_path[v]=u;
                        if(v==t)
                        {
                                printf("\nAgumented path found: ");
                                for(i=0;i<NUM_NODES;i++)
                {
                    printf("%d ",aug_path[i]);
                    }
                    printf("\n");
                return 1;
                }
                visited[v]=1;
                insert_queue(v);
            }
        }
    }
    return 0;
}
main()
{
    int graph[NUM_NODES][NUM_NODES]={{0,20,4,7,0},
                                    {0,0,5,3,7},
                                    {0,0,0,0,8},
                                    {0,0,0,0,15},
                                    {0,0,0,0,0}
                                   };
    int i,j;
    int u,v;
    int s,t;
    int total_flow=0;
    int min_flow=2147483627;
    for(i=0;i<NUM_NODES;i++)
    {
        for(j=0;j<NUM_NODES;j++)
        {
            res_graph[i][j]=graph[i][j];
        }
    }
    s=0;
    t=4;
    for(i=0;i<NUM_NODES;i++)
    {
        for(j=0;j<NUM_NODES;j++)
        {
            printf("%d ",res_graph[i][j]);
        }
        printf("\n");
    }
    while(find_augmented_path(s,t))
    {
        min_flow=2147483627;

        for(v=t;v!=s;v=u)
```

```
            {
                u=aug_path[v];
                if(res_graph[u][v]<min_flow)
                    min_flow=res_graph[u][v];
            }
            total_flow+=min_flow;
            for(v=t;v!=s;v=u)
            {
                u=aug_path[v];
                res_graph[u][v]-=min_flow;
            }
        printf("\nResidual Graph\n");
        for(i=0;i<NUM_NODES;i++)
        {
            for(j=0;j<NUM_NODES;j++)
            {
                printf("%d ",res_graph[i][j]);
            }
            printf("\n");
        }
  }
  printf("Total Flow: %d",total_flow);
}
```