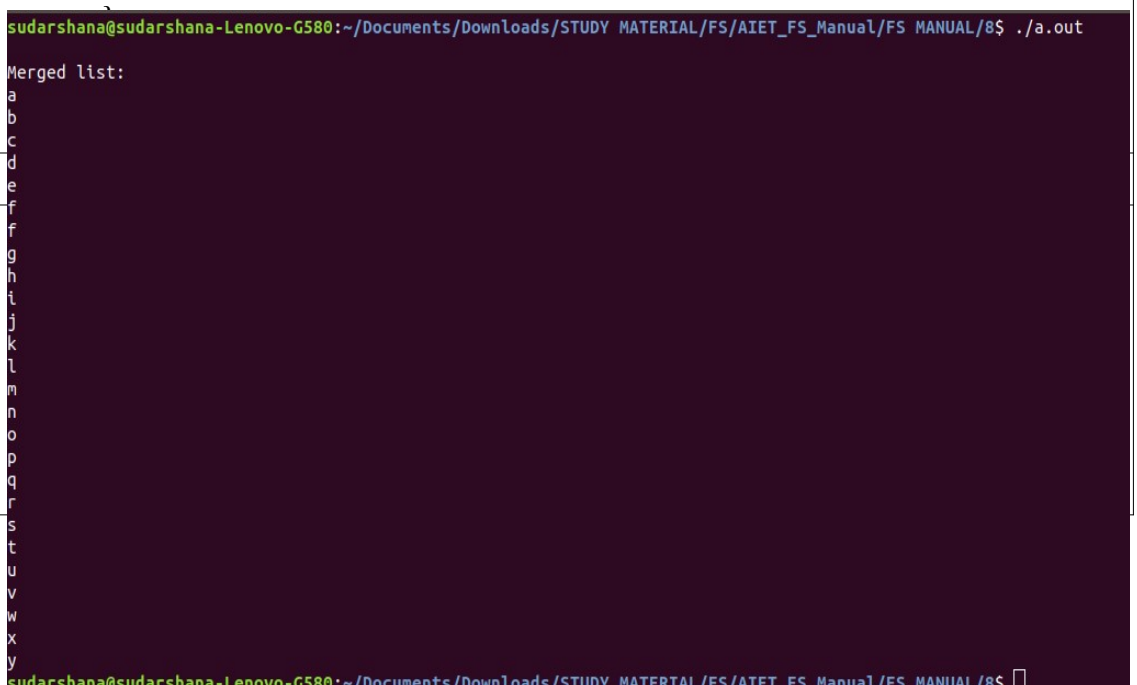
Date:	
Expt. No.:	8
Title:	8-Way Merge Algorithm
Aim:	Write a program to read k Lists of names and merge them using k-way merge algorithm with k = 8.
Program:	

```
#include<iostream>
#include<string>
#include<fstream>
#include<stdlib.h>

using namespace std;

class coseq
{
	public:
		string list[8][50];
		string outlist[200];
		int count1[8],count2[8];

		void read_file(int i);
		void sort_list(int i);
		void kwaymerge();
};
void error(int);

int main()
{
	system("clear");
	coseq c;
	for(int i=0; i<8; i++)
	{
		c.count1[i] = 0;
		c.read_file(i);
		c.sort_list(i);
	}
	c.kwaymerge();
	return 0;
}

void coseq::read_file(int i)
{
	fstream fp;
	string name;
	switch(i)
	{
		case 0:fp.open("n1.txt",ios::in);break;
		case 1:fp.open("n2.txt",ios::in);break;
		case 2:fp.open("n3.txt",ios::in);break;
		case 3:fp.open("n4.txt",ios::in);break;
		case 4:fp.open("n5.txt",ios::in);break;
```

```
                case 5:fp.open("n6.txt",ios::in);break;
                case 6:fp.open("n7.txt",ios::in);break;
                case 7:fp.open("n8.txt",ios::in);break;
        }
        if(!fp)
                error(1);
        while(fp)
        {
                getline(fp,name);
                if(name.length()>0)
                        list[i][count1[i]++]=name;
        }
        fp.close();
}

void coseq::sort_list(int k)
{
        int i,j;
        string temp;
        for(i=0;i<count1[k];i++)
        {
                for(j=i+1;j<count1[k];j++)
                {
                        if(list[k][i]>list[k][j])
                        {
                                temp=list[k][i];
                                list[k][i]=list[k][j];
                                list[k][j]=temp;
                        }
                }
        }
}

void coseq::kwaymerge()
{
        string sml;
        int s_list,count3=0,strt=0,avail_list=8,avail[8],i;
        for(i=0;i<8;i++)
        {
                avail[i]=1;
                count2[i]=0;
        }
        while(avail_list>1)
        {
                if(!avail[strt])
                {
                        strt++;
                        continue;
                }
                s_list=strt;
                sml=list[strt][count2[strt]];
                for(i=strt+1;i<8;i++)
```

```
		{
			if(!avail[i])
				continue;
			if(list[i][count2[i]]<sml)
			{
				sml=list[i][count2[i]];
				s_list=i;
			}
		}
		count2[s_list]++;
		if(count2[s_list]==count1[s_list])
		{
			avail[s_list]=0;
			avail_list--;
		}
		outlist[count3++]=sml;
	}
	for(i=0;i<8;i++)
		if(avail[i])
		{
			for(int j=count2[i];j<count1[i];j++)
				outlist[count3++]=list[i][j];
		}

	cout<<"\nMerged list:\n";
	for(i=0;i<count3;i++)
	{
		if(outlist[i]==outlist[i+1])continue;
			cout<<outlist[i]<<endl;
	}
}

void error(int error_type)
{
	switch(error_type)
	{
		case 1: cout<<"\nFATAL ERROR!: Unable to open the File(s)\n";
			exit(0);
```

Output:



Result:	A program to read two lists of names and then match the names in the two lists using Consequential Match based on a single loop is developed and Output the names common to both the lists.