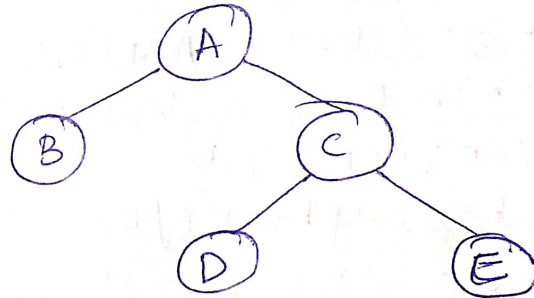


A \rightarrow BA \rightarrow CC \rightarrow DC \rightarrow E5 Routers

Topology.

Routing table entries.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 1 | 2 | 2 |
| B | 1 | 0 | 2 | 2 | 2 |
| C | 1 | 2 | 0 | 1 | 1 |
| D | 2 | 2 | 1 | 0 | 2 |
| E | 2 | 2 | 1 | 2 | 0 |

Code

```

#include <conio.h>
#include <iostream.h>
#define Max 10
int n;
class router {
    char adj-new[Max],
    adj-old[Max];
    int table-new[Max],
    table-old[Max];
public: router() {
    for (int i = 0; i < Max; i++)
        table-old[i] = table-new[i] = 99;
}
}

```

```

void copy() {
    for (int i=0; i<n; i++)
        { adj-old[i] = adj-new[i];
          table-old[i] = table-new[i];
        }
}

```

```

{
}

```

```

int equal() {
    for (int i=0; i<n; i++)
        if (table-old[i] != table-new[i] || adj-old[i] != adj-new[i])
            return 0;
}

```

```

return 1;
}

```

```

void input(int j) {

```

```

    cout << "Enter 1 if the corresponding router is adjacent to.
    router" << (char) ('A'+j) << " else enter 99" << endl;

```

```

    for (int p=0; p<n; p++)

```

```

        if (p != j) cout << (char) ('A'+p) << " ";

```

```

        cout << " In Enter matrix: ";

```

```

        for (p=0; p<n; p++)

```

```

            if (i == j)

```

```

                table-new[p] = 0;

```

```

            else

```

```

                cin >> table-new[p];

```

```

                adj-new[p] = (char) ('A'+i);

```

```

            }

```

```

        cout << endl;

```

```

    }

```

```

void display()

```

```

{
    cout << "In Destination Router: ";
}

```

```

for (int i=0; i<n; i++)

```



```

for (int p=0; p<n; p++)
    cout << (char)('A'+p) << " ";
cout << "\n Outputing km: ";
for (i=0; i<n; i++)
    cout << adjNew[i] << " ";
cout << "\n Hop count: ";
for (k=0; k<n; k++)
    cout << table - new[k] << " ";
}

```

```

void build.(int j)
{

```

```

    for (int p=0; p<n; p++)
        for (int k=0; (i!=p) && (k<n); k++)
            if (table - old[i] == 99)
                if (table - new[i] + x[i].table - new[k];
                    adj - new[k] = (char)('A'+p);
}

```

```

}
}
}

```

```

void build_table()
{

```

```

    int p=0, j=0;
    while (p!=n)
        for (p=j; p<n; p++)

```

```

        {
            x[p].copy();
            x[i].build(p);
        }

```

```

    for (p=0; p<n; p++)

```

```

        if (!x[p].equal(i))
            if (j=p; break;

```

Teacher's Signature: _____

Expt. No.

Page No.

```
void main()
```

```
{ class();
```

```
cout << "Enter the number the routers (< " << max << "
```

```
" ; cin >> n;
```

```
for (int i=0; i<n; i++)
```

```
    r[i].input(i);
```

```
    build-table();
```

```
for (i=0; i<n; i++)
```

```
{
```

```
    cout << "Router Table entries for router "
```

```
    << (char)('A'+i) << " :- "
```

```
    r[i].display();
```

```
    cout << endl << endl;
```

```
}
```

```
getch();
```

```
}
```

Teacher's Signature : _____