

Distance vector algorithm:

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-new[i] != adj-old[i])

return 0;

return 1;

}

Void input (int j) {

Revanth. NM
IB M18CS081

Edo-14)

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

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

```
{ if (i != j)
```

```
    cout << (char) ('A' + j) << " ";
```

```
cout << "\n Enter matrix: ";
```

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

```
    if (i == j)
```

```
        table_new[i] = 0;
```

```
    else
```

```
        cin >> table_new[i];
```

```
        adj_new[i] = (char) ('A' + j);
```

```
    cout << endl;
```

```
}
```

```
Void display () {
```

```
    cout << "\n Destination Router: ";
```

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

```
        cout << (char) ('A' + i) << " ";
```

```
    cout << "\n Outgoing line: ";
```

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

```
        cout << adj_new[i] << " ";
```

```
    cout << "\n Hop count: ";
```

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

```
        cout << table_new[i] << " ";
```

```
}
```



```
void build(int j) {
```

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

```
        for(int k=0; (i!=j) && (k<n); k++)
```

```
            if (table.old[i] != 99)
```

```
                if ((table.new[i] + r[i].table.new[k]) < table.new[k]) {
```

```
                    table.new[k] = table.new[i] + r[i].table.new[k];
```

```
                    adj.new[k] = (char)('A' + i);
```

```
            }
```

```
        }
```

```
    } r(10);
```

```
void build table() {
```

```
    int i=0, j=0; if (i==0) {
```

```
        while (i != n) {
```

```
            for (i=j; i<n; i++) {
```

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

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

```
            }
```

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

```
                if (!r[i].equal()) {
```

```
                    j=i;
```

```
                    break;
```

```
                }
```

```
            }
```

```
        }
```

```
void main() {
```

```
    clear();
```

```
    cout << "Enter the number of routers (<< n << max << n): ";
```

```
    (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();
```

```
    }
```