

PROGRAM - 1

Write program for distance vector algorithm

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

```

for (int i = 0; i < n; i++)
    if (i != j) cout << (char)('A' + i) << " ";
    cout << "\nEnter Matrix: ";
    for (i = 0; i < n; i++) {
        if (i == j)
            table_new[i] = 0;
        else
            cin >> table_new[i];
        adj_new[i] = (char)('A' + i);
    }
    cout << endl;
}

```

```

void display () {
    cout << "\nDestination Router: ";
    for (int i = 0; i < n; i++) cout << (char)('A' + i) << " ";
    cout << "\nEnter Outgoing Line: ";
    for (int i = 0; i < n; i++) cout << adj_new[i] << " ";
    cout << "\nEnter Hop Count: ";
    for (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] + (i).table_new[k] < table_new[k]) {
                    table_new[k] = table_new[i] + (i).table_new[k];
                    adj_new[k] = (char)('A' + i);
                }
}
} h[10];

```

```

void build-table() {
    int i = 0, j = 0;
    while (i != n) {
        for (j = i; j < n; j++) {
            r[i].copy(j);
            r[i].build(j);
        }
        for (i = 0; i < n; i++) {
            if (!r[i].equal()) {
                j = i;
                break;
            }
        }
    }
}

```

```

void main() {
    clrscr();
    cout << "Enter 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)
        << ":\n";
        r[i].display();
        cout << endl << endl;
    }
    getch();
}

```