```
#include <iostream>
#include <conio.h>

#define MAX 10

using namespace std;

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: " << endl
<< "";
for (int i = 0; i < n; i++)
if (i != j)
cout << (char)('A' + i) << "";
cout << "\nEnter 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' + i);
}
cout << endl;
}
void display()
{
int i;
cout << "\nDestination Router: ";
for (i = 0; i < n; i++)
cout << (char)('A' + i) << " ";
cout << "\nOutgoing Line: ";
for (i = 0; i < n; i++)
cout << adj_new[i] << "";
cout << "\nHop Count: ";
for (i = 0; i < n; i++)
cout << table_new[i] << "";
}
void build(int j)
{
router r[10];
for (int i = 0; i < n; i++)
if (table_old[i] != 99)
for (int k = 0; (i != j) && (k < n); k++)
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;
while (i != n)
{
for (i = j; i < n; i++)
{
r[i].copy();
r[i].build(i);
}
for (i = 0; i < n; i++)
if (!r[i].equal())
{
j = i;
break;
}
}
}
int main()
{
// clrscr();

cout << "Enter the number the routers(<" << MAX << "): ";
```

```
cin >> n;
for (int i = 0; i < n; i++)
{
r[i].input(i);
}
build_table();
for (int i = 0; i < n; i++)
{
cout << "Router Table entries for router " << (char)('A' + i) << ":-";
r[i].display();
cout << endl;
}
return 0;
// getch();
}
```