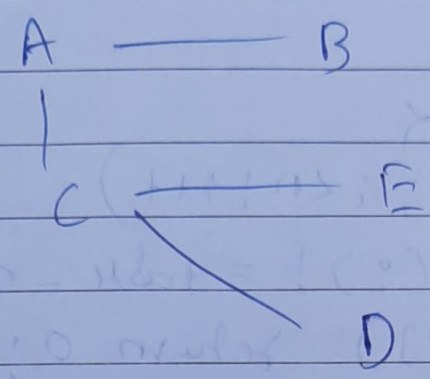
Sumalata
1BM19CS409

Saathi

Date __/__/____

1) Distance vector algorithm.




```
# 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);
```

Date ___/___/_____

```
table_old[i] = table_new[i];
}
}
int equal() {
for(int i=0; i<n; i++)
if(table_old[i] != table_new[i] || adj_
~~new~~ old[i]) return 0;
return 1;
}
void input(int i) {
cout << "Enter 1 if the corresponding
router is adjacent to router ";
for(int i=0; i<n; i++)
if(i==j) cout << (char)('A'+i) << "
cout << "\n Enter matrix:\n";
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;
}
```

Date __/__/____

Saathi

```
void display() {
cout << "\n Destination router: ";
for(int i=0; i<n; i++) cout << (char)('A'+i) << " ";
cout << "\n Outgoing line: ";
for(i=0; i<n; i++) cout << adj_new[i] << " ";
cout << "\n 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] + d[i], table_new[k])
    < table_new[k]) {
  table_new[k] = table_new[i] + d[i].table_new[k];
  adj_new[k] = (char)('A'+i);
}
}
build(0);

void build_table() {
int i=0, j=0;
while(i!=n) {
for(i=j; i<n; i++) {
  d[i].copy();
```

Date __/__/____

```
    r[i].build(i);
  }
  for(i=0; i<n; i++)
  if (!r[i].equal()) {
     j=i;
    break;
    }
  }
}
void main() {
  clrscr();
cout << "enter the no. the routers (<" << MAX << "
   ; cin>>n;
  for (int i=0; i<n; i++) r[i].inPW(i);
  build_table();
  for (i=0; i<n; i++) {
  cout << "routes table entries for routers"
    << (char)('A'+i) << ":-";
    r[i].display();
    cout << endl << endl;
  }
  getch();
}
```