//Traveling Salesman

int n, x[11], y[11], dist[11][11], memo[11][1 << 11], dp[11][1 << 11];

//dist[u][v] = distance from u to v

//dp[u][bitmask] = dp[node][set\_of\_taken\_nodes] (saves the best(min/max) path)

//call with tsp(starting node, 1)

int tsp(int u, int bitmask) { //startin node and bitmask of taken nodes

if(bitmask == ((1 << (1+n)) - 1)) //when it steps in this node, if all nodes are visited

return dist[u][0]; //then return to 0'th (starting) node [as the path is hamiltonian]

//or use return dist[u][start] if starting node is not 0

if(dp[u][bitmask] != -1) //if we have previous value set up

return dp[u][bitmask]; //use that previous value

int ans = 1e9; //set an infinite value

for(int v = 0; v <= n; v++) //for all the nodes

if(u != v && !(bitmask & (1 << v))) //if this node is not the same node, and if this node is not used yet(in bitmask)

ans = min(ans, dist[u][v] + tsp(v, bitmask | (1 << v))); //min(past\_val, dist u->v + dist(v->all other untaken nodes))

return dp[u][bitmask] = ans; //save in dp and return

}