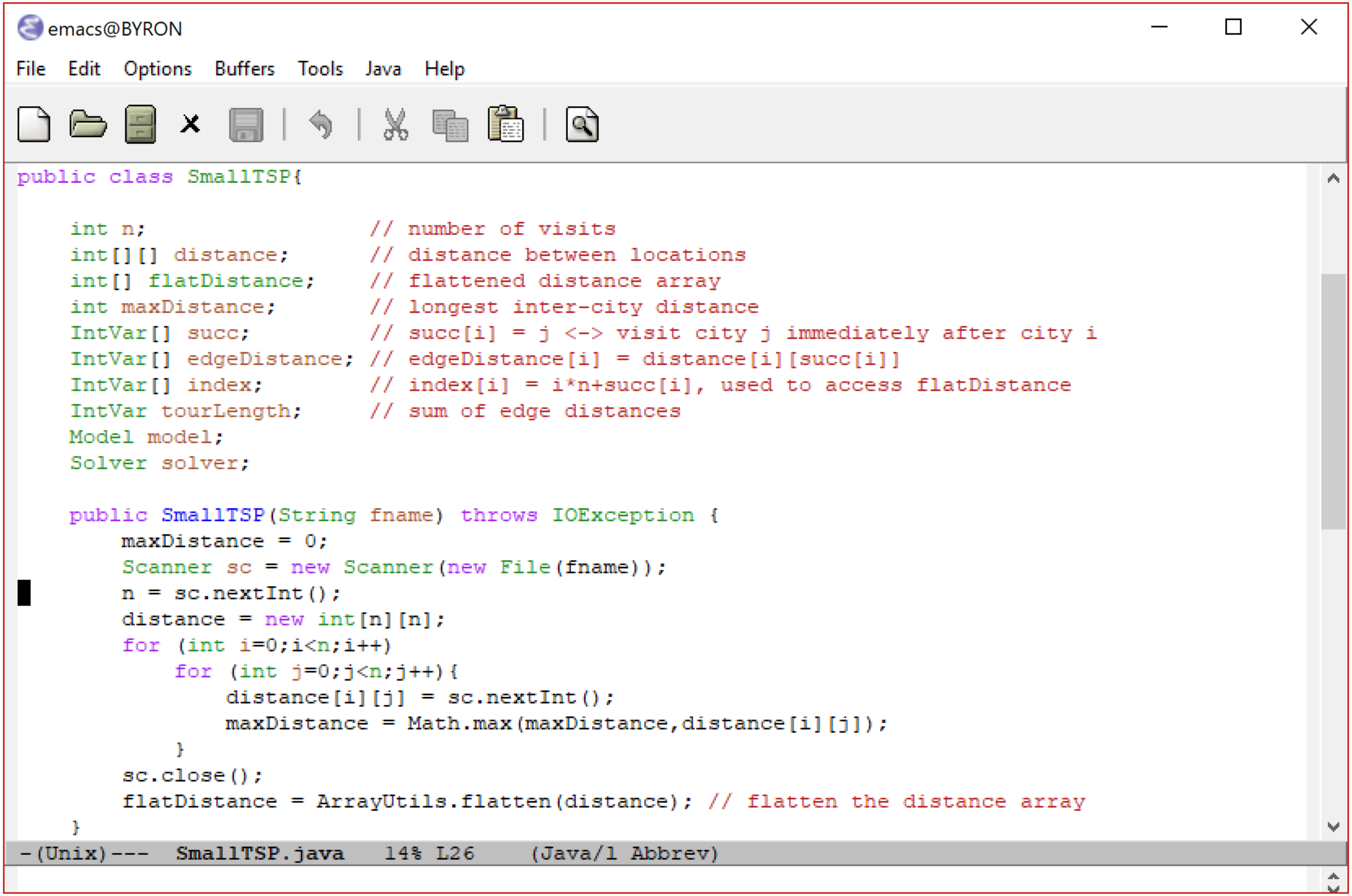
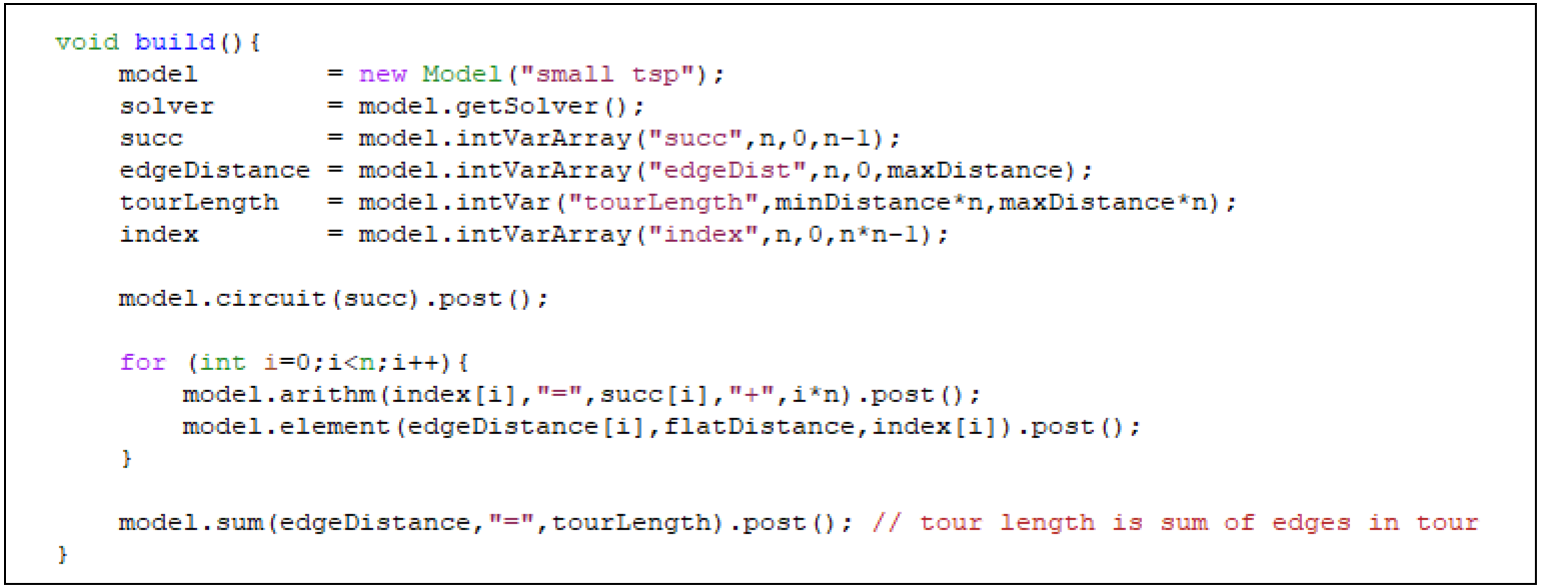
Small TSP
choco4

Example of using element constraint

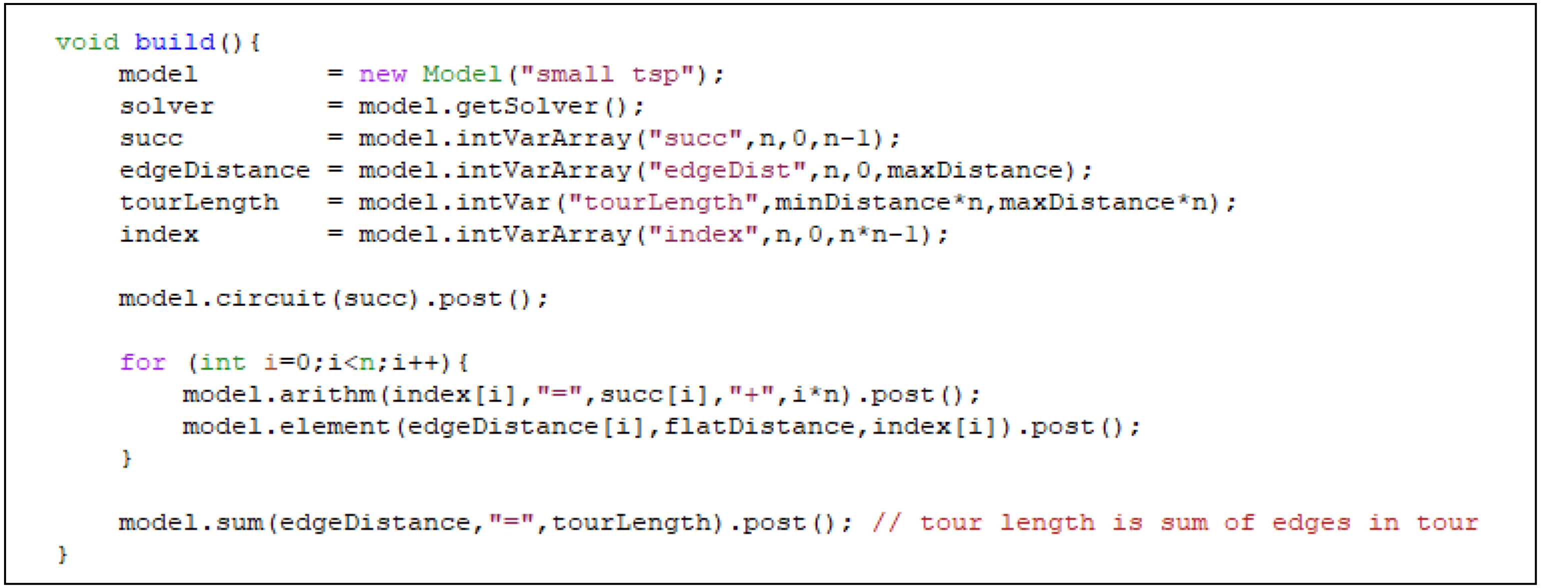


```
public class SmallTSP{

    int n;                     // number of visits
    int[][] distance;          // distance between locations
    int[] flatDistance;        // flattened distance array
    int maxDistance;           // longest inter-city distance
    IntVar[] succ;             // succ[i] = j <-> visit city j immediately after city i
    IntVar[] edgeDistance;     // edgeDistance[i] = distance[i][succ[i]]
    IntVar[] index;            // index[i] = i*n+succ[i], used to access flatDistance
    IntVar tourLength;         // sum of edge distances
    Model model;
    Solver solver;

    public SmallTSP(String fname) throws IOException {
        maxDistance = 0;
        Scanner sc = new Scanner(new File(fname));
        n = sc.nextInt();
        distance = new int[n][n];
        for (int i=0;i<n;i++)
            for (int j=0;j<n;j++){
                distance[i][j] = sc.nextInt();
                maxDistance = Math.max(maxDistance,distance[i][j]);
            }
        sc.close();
        flatDistance = ArrayUtils.flatten(distance); // flatten the distance array
    }
```

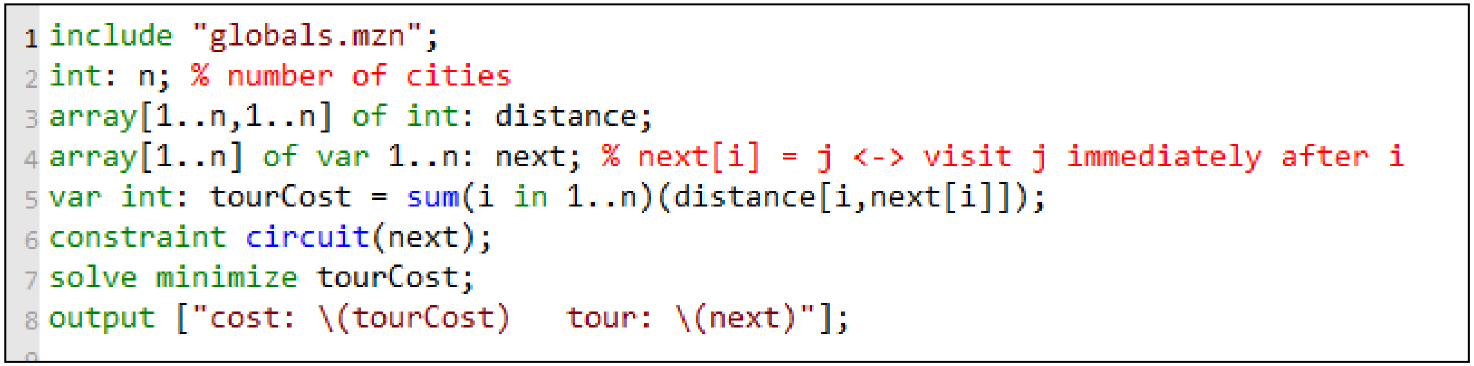
```
-(Unix)---   SmallTSP.java    14% L26     (Java/l Abbrev)
```



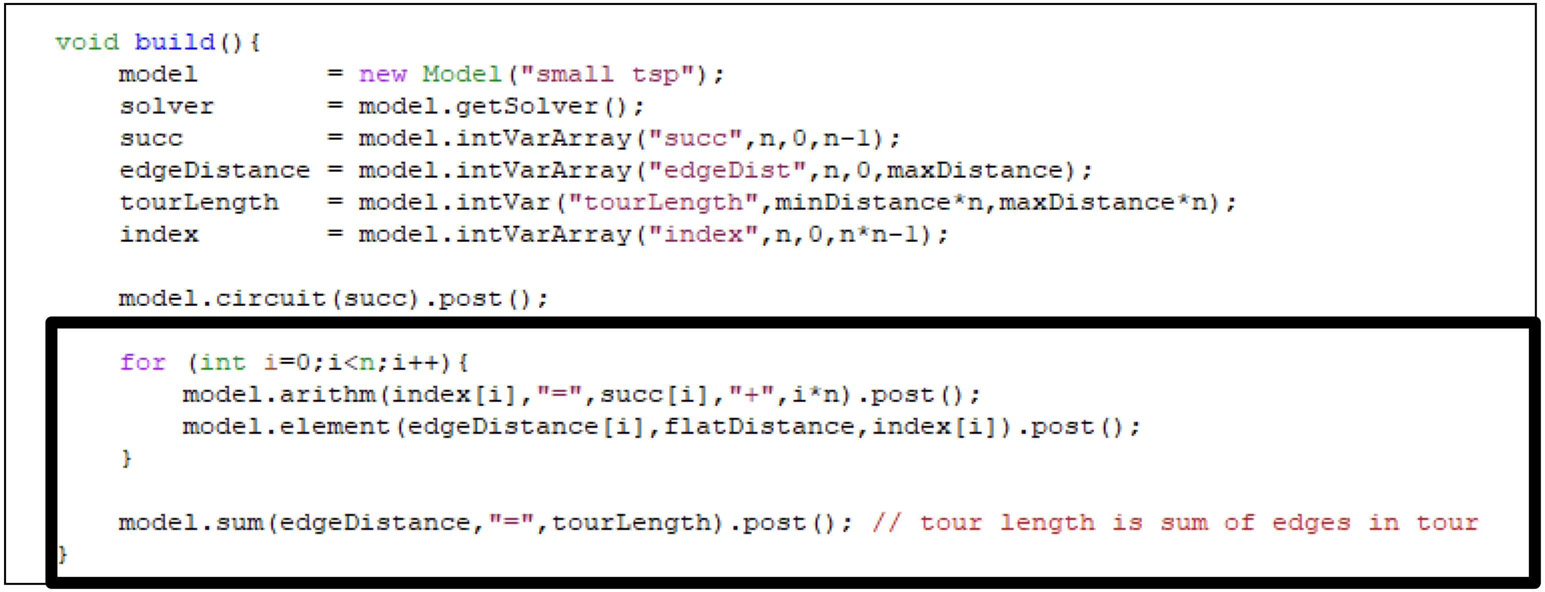
```
void build(){
    model        = new Model("small tsp");
    solver       = model.getSolver();
    succ         = model.intVarArray("succ",n,0,n-1);
    edgeDistance = model.intVarArray("edgeDist",n,0,maxDistance);
    tourLength   = model.intVar("tourLength",minDistance*n,maxDistance*n);
    index        = model.intVarArray("index",n,0,n*n-1);

    model.circuit(succ).post();

    for (int i=0;i<n;i++){
        model.arithm(index[i],"=",succ[i],"+",i*n).post();
        model.element(edgeDistance[i],flatDistance,index[i]).post();
    }

    model.sum(edgeDistance,"=",tourLength).post(); // tour length is sum of edges in tour
}
```



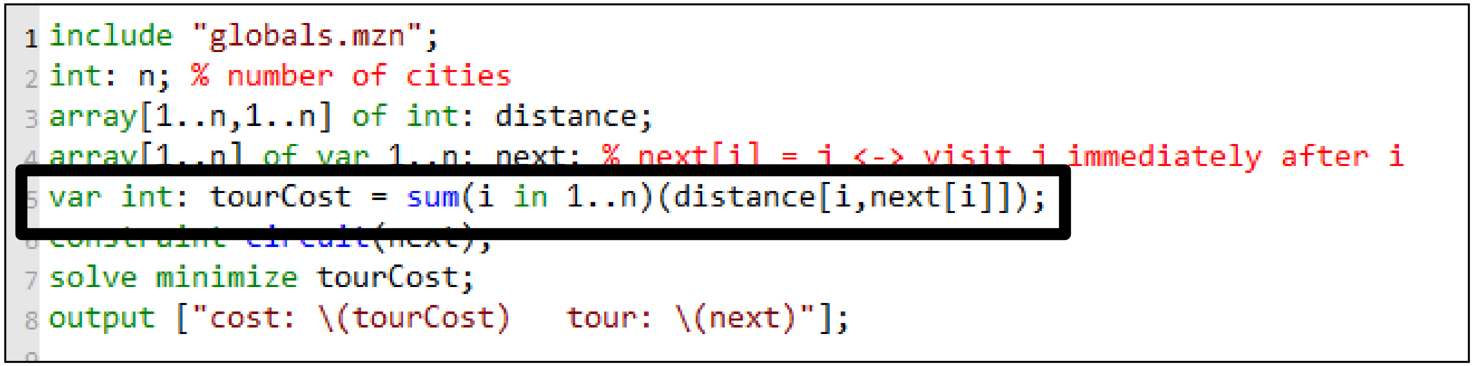
```
void build(){
    model          = new Model("small tsp");
    solver         = model.getSolver();
    succ           = model.intVarArray("succ",n,0,n-1);
    edgeDistance = model.intVarArray("edgeDist",n,0,maxDistance);
    tourLength     = model.intVar("tourLength",minDistance*n,maxDistance*n);
    index          = model.intVarArray("index",n,0,n*n-1);

    model.circuit(succ).post();

    for (int i=0;i<n;i++){
        model.arithm(index[i],"=",succ[i],"+",i*n).post();
        model.element(edgeDistance[i],flatDistance,index[i]).post();
    }

    model.sum(edgeDistance,"=",tourLength).post(); // tour length is sum of edges in tour
}
```



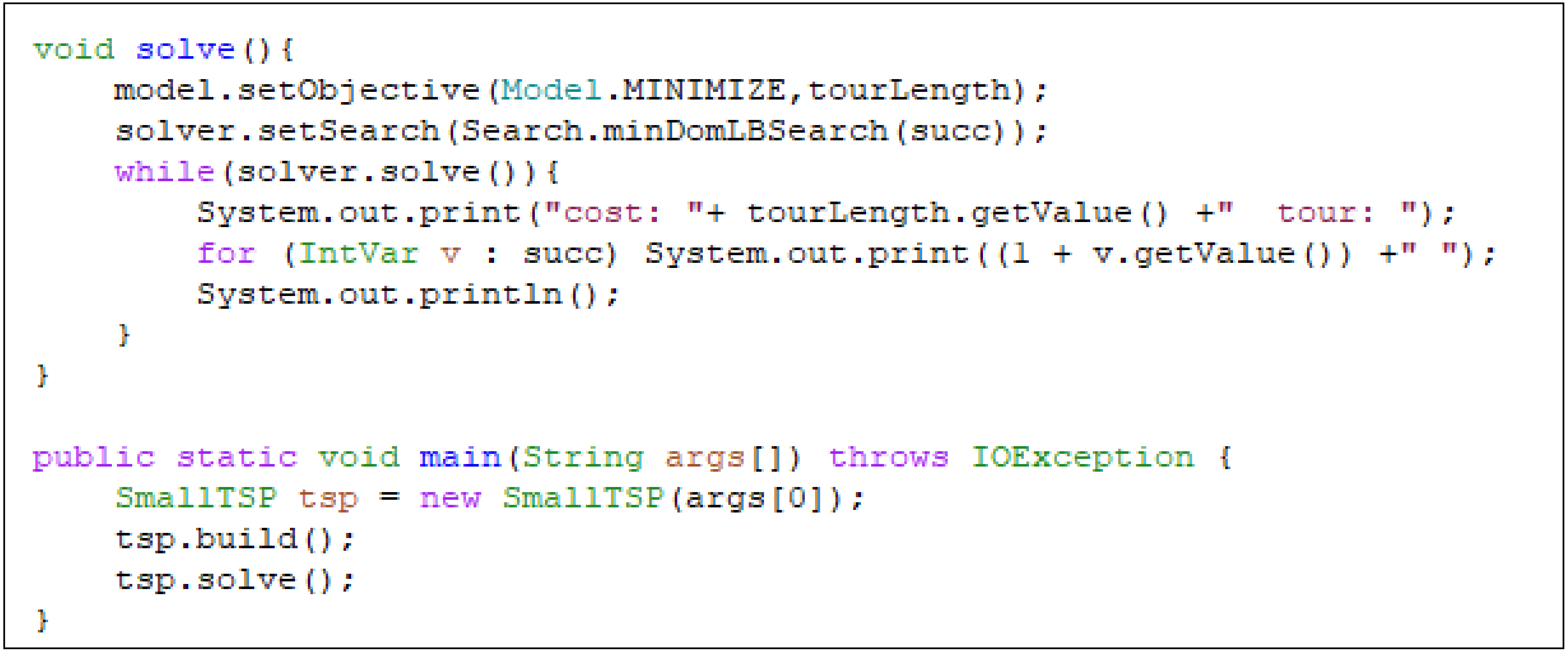
```
include "globals.mzn";
int: n; % number of cities
array[1..n,1..n] of int: distance;
array[1..n] of var 1..n: next; % next[i] = j <-> visit j immediately after i
var int: tourCost = sum(i in 1..n)(distance[i,next[i]]);
constraint circuit(next);
solve minimize tourCost;
output ["cost: \(tourCost)   tour: \(next)"];
```



```
void build(){
    model        = new Model("small tsp");
    solver       = model.getSolver();
    succ         = model.intVarArray("succ",n,0,n-1);
    edgeDistance = model.intVarArray("edgeDist",n,0,maxDistance);
    tourLength   = model.intVar("tourLength",minDistance*n,maxDistance*n);
    index        = model.intVarArray("index",n,0,n*n-1);

    model.circuit(succ).post();

    for (int i=0;i<n;i++){
        model.arithm(index[i],"=",succ[i],"+",i*n).post();
        model.element(edgeDistance[i],flatDistance,index[i]).post();
    }

    model.sum(edgeDistance,"=",tourLength).post(); // tour length is sum of edges in tour
}
```

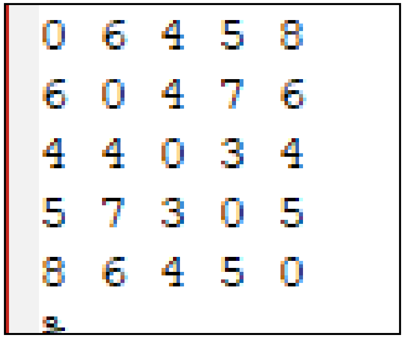


```
include "globals.mzn";
int: n; % number of cities
array[1..n,1..n] of int: distance;
array[1..n] of var 1..n: next; % next[i] = j <-> visit j immediately after i
var int: tourCost = sum(i in 1..n)(distance[i,next[i]]);
constraint circuit(next);
solve minimize tourCost;
output ["cost: \(tourCost)   tour: \(next)"];
```

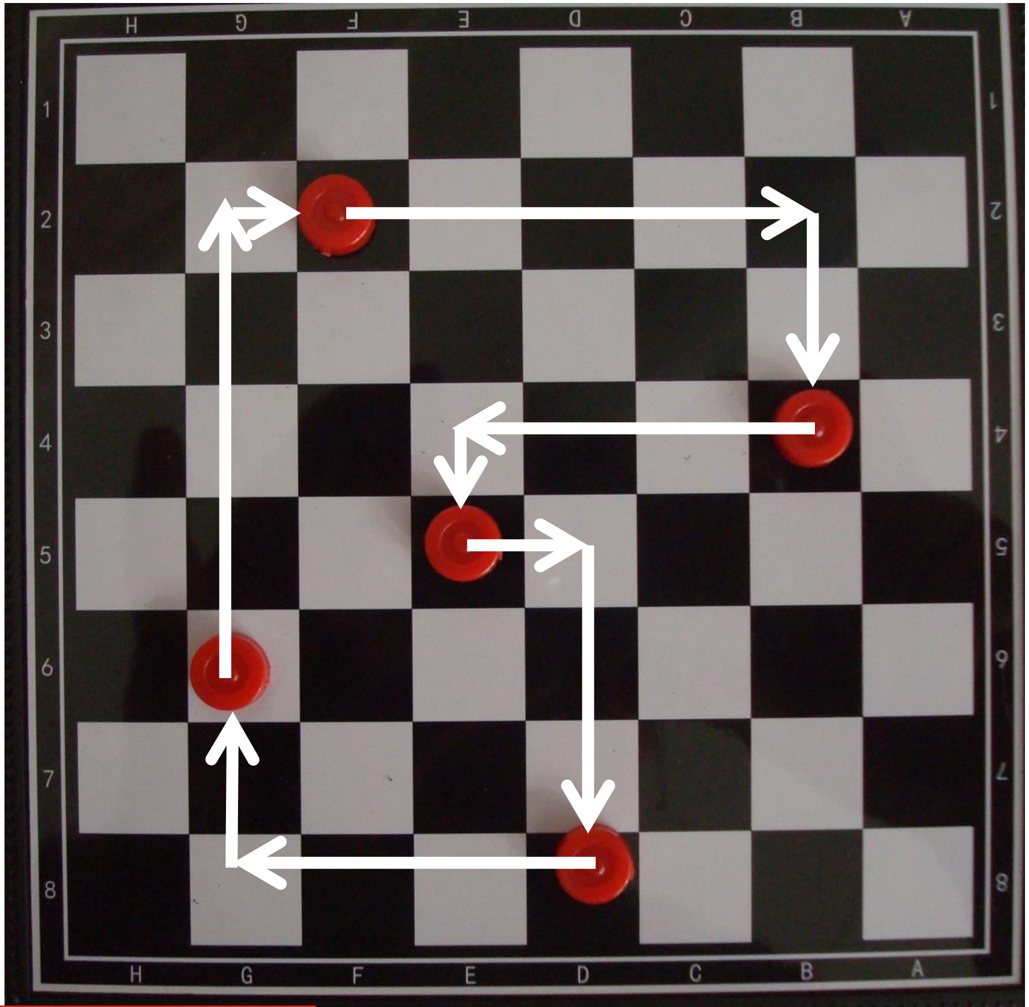


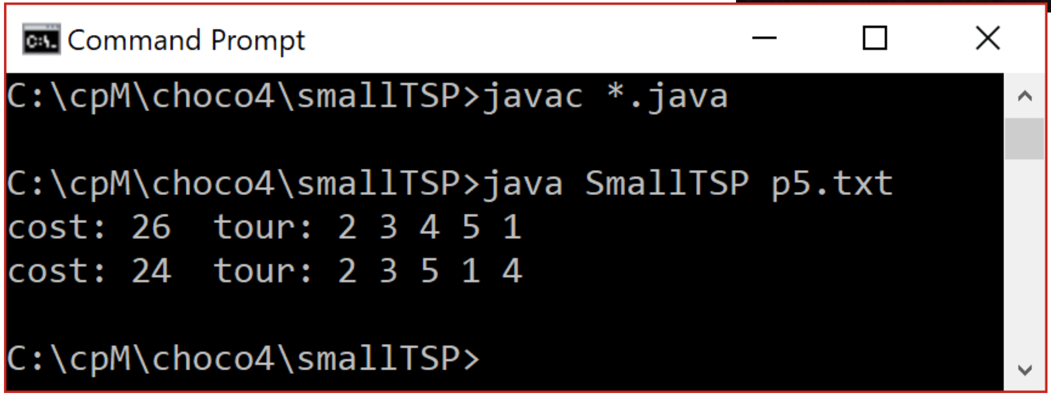
```
void solve(){
    model.setObjective(Model.MINIMIZE,tourLength);
    solver.setSearch(Search.minDomLBSearch(succ));
    while(solver.solve()){
        System.out.print("cost: "+ tourLength.getValue() +"  tour: ");
        for (IntVar v : succ) System.out.print((1 + v.getValue()) +" ");
        System.out.println();
    }
}

public static void main(String args[]) throws IOException {
    SmallTSP tsp = new SmallTSP(args[0]);
    tsp.build();
    tsp.solve();
}
```

choco4



```
0 6 4 5 8
6 0 4 7 6
4 4 0 3 4
5 7 3 0 5
8 6 4 5 0
```





```
C:\cpM\choco4\smallTSP>javac *.java

C:\cpM\choco4\smallTSP>java SmallTSP p5.txt
cost: 26  tour: 2 3 4 5 1
cost: 24  tour: 2 3 5 1 4

C:\cpM\choco4\smallTSP>
```

6 + 4 + 4 + 5 + 5 = 24

Feic!