Problem 1. Travelling Salesman Problem

a) Below you find a cost matrix for the asymmetric graph with 6 nodes.

Node	1	2	3	4	5	6
1	-	7	65	68	34	81
2	19	-	22	27	59	29
3	14	43	-	62	77	65
4	76	53	64	-	6	51
5	39	58	38	27	-	13
6	46	67	27	11	38	-

and the corresponding cost matrix

Node	1	2	3	4	5	6	7	8
1	-	28	55	42	57	37	31	34
2		-	67	26	29	48	57	22
3			-	61	96	19	60	52
4				-	39	43	70	12
5					-	77	85	44
6						-	52	34
7							-	59

Find the optimal Hamiltonian cycle by using the model for Symmetric TSP and iteratively eliminating generated sub-cycles (if any) by adding the corresponding sub-cycle eliminating constraints (SEC);

2) AMPL code

```
File task1b.dat:
param n:= 6;
param K := 2;
set SUBN[1] := 4 5 6;
set SUBN[2] := 1 2 3;
param cost:
  1  2  3  4  5  6  7  8:=
1 . 28 55 42 57 37 31 34
2 .  . 67 26 29 48 57 22
3 .  .  . 61 96 19 60 52
4 .  .  .  . 39 43 70 12
5 .  .  .  .  . 77 85 44
6 .  .  .  .  .  . 52 34
7 .  .  .  .  .  .  . 59;

File task1b.mod:
param n;
param K; # number of sub-cycles
set EDGES:= {i in 1..n,j in 1..n: i < j};
param cost {EDGES} >=0;
set SUBN {k in 1..K}; # set of nodes in sub-cycle Sk
set SUBE {k in 1..K}:= {i in SUBN[k], j in SUBN[k]: i < j}; #
edges
```

```
var Route {EDGES} binary;

minimize Total_Cost:
     sum {(i,j) in EDGES} cost[i,j] * Route[i,j];

subject to Degree_Const {i in 1..n}:
     sum {(i,j) in EDGES} Route[i,j] + sum {(j,i) in EDGES} Route[j,i] = 2;
subject to SEC {k in 1..K}: # SEC
     sum {(i,j) in SUBE [k]} Route [i,j] <= card(SUBN [k]) - 1;

file task1b.run:
option solver cplex;
model task1b.mod;
data task1b.dat;
option show_stats 1;
solve;
display Total_Cost > task1b.sol;
display Route > task1b.sol;

file task1b.sol:
Total_Cost = 205

Route :=
1 2   0
1 3   1
1 4   0
1 5   0
1 6   1
2 3   0
2 4   1
2 5   1
2 6   0
3 4   0
3 5   0
3 6   1
4 5   1
4 6   0
5 6   0
;
```

3) Solution

Optimal route should be 1-3-4-5-2-6-1.
Total cost in this case will be 205.

Problem 2. Hamiltonian Path Problem

Find the shortest Hamiltonian path for the graph in Problem 1 using the Asymmetric TSP model on the extended graph with one added artificial node and corresponding arcs.

2) AMPL code

```
File task1c.dat:
param n:= 7;
param cost:
  1  2  3  4  5  6  7:=
1 .  7  65 68 34 81 0
2 19 .  22 27 59 29 0
3 14 43 .  62 77 65 0
4 76 53 64 .   6 51 0
5 39 58 38 27 .  13 0
6 46 67 27 11 38 .  0
7  0  0  0  0  0  0 .;

File task1c.sol:
Total_Cost = 67

Route [*,*]
:   1   2   3   4   5   6   7    :=
1   .   1   0   0   0   0   0
2   0   .   0   1   0   0   0
3   1   0   .   0   0   0   0
4   0   0   0   .   1   0   0
5   0   0   0   0   .   1   0
6   0   0   0   0   0   .   1
7   0   0   1   0   0   0   .
;
```

3) Solution

Optimal route should be 1-2-4-5-6-7-3-1.
Total cost in this case will be 67.

Problem 3. Multiple Travelling Salesman Problem

Find for the graph from Problem 1 the optimal solution to 2-TSP problem using the Symmetric TSP model where two travelling persons start their tours from node 1, and they visit at least three nodes each.

2) AMPL code

File task1d.dat:

```
param n:= 8;
param m:= 4;
param v:= 3;
param cost:
  1  2  3  4  5  6  7  8:=
1 .  28 55 42 57 37 31 34
2 .  .  67 26 29 48 57 22
3 .  .  .  61 96 19 60 52
4 .  .  .  .  39 43 70 12
5 .  .  .  .  .  77 85 44
6 .  .  .  .  .  .  52 34
7 .  .  .  .  .  .  .  59;

param start:= 1;
```

file task1d.mod:

```
param n;
param m;
param v;
set EDGES:= {i in 1..n,j in 1..n: i < j};
param cost {EDGES} >=0;
param start; #start

var Route {EDGES} binary;
minimize Total_Cost:
     sum {(i,j) in EDGES} cost[i,j] * Route[i,j];

subject to Start: #Start from depot
     sum {(start,j) in EDGES} Route[start,j] = m;
subject to Degree_Const {i in 2..n}:
     sum {(i,j) in EDGES} Route[i,j] + sum {(j,i) in EDGES}
Route[j,i] = 2 ;
subject to Visit {i in 1..n}:
     sum {(i,j) in EDGES} Route[i,j] >= v;
```

file task1d.run:

```
option solver cplex;
model task1d.mod;
data task1d.dat;
option show_stats 1;
solve;
display Total_Cost > task1d.sol;
display Route > task1d.sol;
```

```
file task1d.sol:
Total_Cost = 283

Route [*,*]
:   2   3   4   5   6   7   8    :=
1   1   0   0   0   1   1   1
2   .   0   0   1   0   0   0
3   .   .   0   0   1   1   0
4   .   .   .   1   0   0   1
5   .   .   .   .   0   0   0
6   .   .   .   .   .   0   0
7   .   .   .   .   .   .   0
;
```

3) Solution
There should be 2 optimal routes: 1-2-5-4-8-1 and 1-6-3-7-1.
Total cost in this case will be 283.

Problem 4. Travelling Salesman Problem with Profits

Assume that in the graph from Problem 1 the start node is 1 and the incomes collected when visiting nodes are: node 2 is 30, node 3 is 15, node 4 is 20, node 5 is 40, node 6 is 55, node 7 is 45, and node 8 is 45. Using the corresponding mathematical models for TSP with Profits

a) Find the optimal tour on this graph that maximizes the profit, and specify the income collected and the cost of travel;

2) AMPL code

```
File task1e.dat:
param n := 8;
param cost:
  1 2 3 4 5 6 7 8:=
1 . 28 55 42 57 37 31 34
2 . . 67 26 29 48 57 22
3 . .  . 61 96 19 60 52
4 . . .  . 39 43 70 12
5 . . . . . 77 85 44
6 . . . . . . 52 34
7 . . . . . . . 59;
param income := 2 30 3 15 4 20 5 40 6 55 7 45 8 45;

File task1e.mod:
param n;
set ARCS := {i in 1..n, j in 1..n: i<j};
```

```
param cost {ARCS} >= 0;
param income {2..n};

var x {ARCS} >= 0 binary;
var y {2..n} >= 0 binary;

maximize Total_Profit:
     sum  {i  in  2..n}  income[i]*y[i]  -  sum  {(i,j)  in  ARCS}
cost[i,j]*x[i,j];

subject to Start_Depot:
     sum {(1,j) in ARCS } x[1,j] = 2;
subject to Linking {i in 2..n}:
     sum {(i,j) in ARCS } x[i,j] + sum {(j,i) in ARCS } x[j,i]=
2*y[i];

File task1e.run:
option solver cplex;
model task1e.mod;
data task1e.dat;
option omit_zero_rows 1;
solve;
display Profit > task1e.sol;
display x > task1e.sol;
display sum {i in 2..n} income[i]*y[i] > task1e.sol;
display sum {(i,j) in ARCS} cost[i,j]*x[i,j] > task1e.sol;

File task1e.sol:
Profit = 19
x :=
1 2   1
1 6   1
2 4   1
4 8   1
6 8   1
;
sum{i in 2 .. n} income[i]*y[i] = 150
sum{(i,j) in ARCS} cost[i,j]*x[i,j] = 131
```

3) Solution

Optimal route should be 1-2-4-8-6-1.
Profit in this case will be 150 - 131 = 19.

b) Find the optimal tour that maximises the collected income within the travel budget Cmax = 120, and specify the budget used.

2) AMPL code

```
File task1e.dat:
param c_max = 120;

File task1e.mod:
param c_max;
subject to Travel_Budget:
     sum {(i,j) in ARCS} cost[i,j]*x[i,j] <= c_max;

File task1e.sol:
Income = 130
x :=
1 2   1
1 6   1
2 8   1
6 8   1
;
sum{i in 2 .. n} income[i]*y[i] = 130
sum{(i,j) in ARCS} cost[i,j]*x[i,j] = 115
```

3) Solution

Optimal route should be 1-2-8-6-1.
The budget used is 115, travel income is 130.

c) Find the minimum cost tour that satisfies the requirement on the minimal collected income Pmin = 200, and specify the income collected.

2) AMPL code

```
File task1e.dat:
param p_min = 200;

File task1e.mod:
param p_min;
subject to Minimal_Income:
     sum {i in 2..n} income[i]*y[i] >= p_min;

File task1e.sol:
Total_Cost = 204
x :=
1 2   1
1 7   1
2 4   1
3 6   1
3 7   1
```

```
4 8   1
6 8   1
;
sum{i in 2 .. n} income[i]*y[i] = 210
```

3) Solution

Optimal route should be 1-2-4-8-6-3-7-1.
Total income will then be 210.
Total cost in this case will be 204.