Decision variables :

$xi = Inventory\ at\ end\ of\ week i$

$zi = Order\ of\ week i$

$bi = A\ binary\ indicator\ of\ order\ at\ week\ i$

Coefficients:

$yi = Demand\ of\ week\ i$

$mi = Minimum\ inventory\ requirement\ at\ week\ i$

Objective function:

$$Min \sum_{i=1}^{10} \text{bi} * 100 + \text{xi} * 2$$

s.t.

$$xi \geq mi$$

$$x1 = 20 - y1 + z1$$

$$x_i = x_{i-1} - y_i + z_i$$

$$bi = 0\ if\ zi = 0, \qquad bi = 1\ if\ zi > 0$$

Result & Code:

```
import gurobipy as gp
from gurobipy import GRB
import itertools
try:
    # Create a new model
    m = gp.Model("Group project assignment 2")

    #parameters
    weeklydemand=[10,10,10,0,0,15,20,20,0,10]
    mininventory=[1,1,1,0,0,1.5,2,2,0,1]

    #decision variables
    x = m.addVars(10,name="wendinv",lb=0.0,vtype='I')#inventory at end of week
    z = m.addVars(10,name="orderamount",lb=0.0,vtype='I')#how much did we order␣
↪that week
    b = m.addVars(10,name="ordertf",lb=0.0,vtype='B') #binary indicator␣
↪variable to check if we ordered something

    #constraints
    m.addConstrs((x[i]>=mininventory[i] for i in␣
↪range(10)),'minenventoryconstraints') #ensure inventory never goes below␣
↪minimum
    m.addConstr(x[0]==20-weeklydemand[0]+z[0],'week 1 inventory constraint')␣
↪#inventory at end of week 1 is sum of initial inventory-week 1 demand+ week␣
↪1 order amount
    m.addConstrs((x[i]==x[i-1]-weeklydemand[i]+z[i] for i in range(1,10)),␣
↪"weekly inventory constraints")
    m.addConstrs((b[i]==0)>>(z[i]==0) for i in range(10))

    # Set objective
    m.setObjective((gp.quicksum(b[i]*100+2*x[i] for i in range(10))), GRB.
↪MINIMIZE)

    # Optimize model
    m.optimize()

    # print optimal value of variables
```

```
    for v in m.getVars():
        print('%s %g' % (v.VarName, v.X))

    # print optimal objective value
    print('Obj: %g' % m.ObjVal)
    spObj = m.ObjVal

except gp.GurobiError as e:
    print('Error code ' + str(e.errno) + ': ' + str(e))
except AttributeError:
    print('Encountered an attribute error')
```

```
Gurobi Optimizer version 9.5.2 build v9.5.2rc0 (mac64[x86])
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads
Optimize a model with 20 rows, 30 columns and 39 nonzeros
Model fingerprint: 0x0f262c14
Model has 10 general constraints
Variable types: 0 continuous, 30 integer (10 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+00, 1e+02]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 2e+01]
  GenCon coe range [1e+00, 1e+00]
Presolve removed 12 rows and 5 columns
Presolve time: 0.00s
Presolved: 8 rows, 25 columns, 24 nonzeros
Presolved model has 8 SOS constraint(s)
Variable types: 0 continuous, 25 integer (10 binary)
Found heuristic solution: objective 680.0000000

Root relaxation: objective 8.200000e+01, 3 iterations, 0.00 seconds (0.00 work
units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0   82.00000    0    5  680.00000   82.00000  87.9%     -    0s
H    0     0                      582.0000000   82.00000  85.9%     -    0s
H    0     0                      578.0000000   82.00000  85.8%     -    0s
H    0     0                      556.0000000   82.00000  85.3%     -    0s
     0     0  182.00000    0    5  556.00000  182.00000  67.3%     -    0s
H    0     0                      492.0000000  182.00000  63.0%     -    0s
     0     2  182.00000    0    5  492.00000  182.00000  63.0%     -    0s
H    3     8                      464.0000000  182.00000  60.8%   0.3    0s
H   24     8                      442.0000000  304.00000  31.2%   0.3    0s


Explored 35 nodes (11 simplex iterations) in 0.04 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)

Solution count 7: 442 464 492 … 680

Optimal solution found (tolerance 1.00e-04)
Best objective 4.420000000000e+02, best bound 4.420000000000e+02, gap 0.0000%
wendinv[0] 10
wendinv[1] 11
wendinv[2] 1
wendinv[3] 1
wendinv[4] 1
wendinv[5] 22
wendinv[6] 2
wendinv[7] 11
wendinv[8] 11
wendinv[9] 1
orderamount[0] -0
orderamount[1] 11
orderamount[2] 0
orderamount[3] -0
orderamount[4] -0
orderamount[5] 36
orderamount[6] 0
orderamount[7] 29
orderamount[8] 0
orderamount[9] 0
ordertf[0] 0
ordertf[1] 1
ordertf[2] 0
ordertf[3] 0
ordertf[4] 0
ordertf[5] 1
ordertf[6] 0
ordertf[7] 1
ordertf[8] 0
ordertf[9] 0
Obj: 442
```