NAME: SHUBHAM SHARMA
ROLL NO: 18i190002
MSC PHD (OR)

Exercise 1):
Q1)(R)

There are five different types of aircrafts given in the following question.

DEFINING THE DECISION VARIABLES:

X_1=No. of aircrafts of type 1

X_2=No. of aircrafts of type 2

X_3=No. of aircrafts of type 3

X_4=No. of aircrafts of type 4

X_5=No. of aircrafts of type 5

Where $X_i \geq 0$ and X_i € INTEGERS for all i€{1,2,3,4,5}

NOW, The capacities of each aircraft is given to be 2,1,2,1,2 and we want to maximize the capacity:

THEREFORE, THE OBJECTIVE FUNCTION IS :

MAXIMIZE: cap= $2X_1+X_2+2X_3+X_4+2X_5$

Subject to constraints:

a)(As it is given that the budget is limited to 13 units only and the costs per aircrafts are given(3,3,2,2,3) , then we define the first constraint):

$$3X_1+3X_2+2X_3+2X_4+3X_5\leq 13$$

b)(As it is given that parking space is limited to 11 units. The space requirements per aircraft are 2, 1, 3, 1, 2 units, respectively, then we define the second constraint:)

$$2X_1+X_2+3X_3+X_4+2X_5\leq 11$$

Q3)(R):

The question described above is solved with the help of AMPL and the files are attached in the zip(ex1.mod,ex1.run,ex1.dat):

```
AMPL
ampl: include ex1.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0.015625
Solve =  0 (summed over threads)
Output = 0
Elapsed = 1
Gurobi 8.0.0: optimal solution; objective 9
2 simplex iterations
1 branch-and-cut nodes
cap = 9

x [*] :=
1  3
2  0
3  1
4  1
5  0
;

ampl:
```

1) The optimal solution(i.e. the maximum capacity of aircrafts) came out to be: 9
2) The values are variables at optimal solution is:

X_1	3
X_2	0
X_3	1
X_4	1
X_5	0

These variables X_i's should be integer(≥0) values because they are representing the no of aircrafts of each type.

Q4)(R):

We have remove the restriction of variables as integers in the

Model that is described in Q1) . Therefore now the values of X_i€REALS and $X_i \geq 0$ for all i€{1,2,3,4,5}

The question described above is solved with the help of AMPL and the files are attached in the zip(ex1b.mod,ex1b.run,ex1b.dat):

AMPL

```
ampl: include ex1b.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0
Solve =  0 (summed over threads)
Output = 0.015625
Elapsed = 1
Gurobi 8.0.0: optimal solution; objective 9.6
2 simplex iterations
cap = 9.6

x [*] :=
1  3.4
2  0
3  1.4
4  0
5  0
;

ampl: |
```

1) The optimal solution(i.e. the maximum capacity of aircrafts) came out to be: 9.6

2) The values are variables at optimal solution is:

X_1	3.4
X_2	0
X_3	1.4
X_4	0
X_5	0

Q5)(R):

If there are only constraints that place a lower bound on the number of AIRCRAFTS, but no constraints that place an upper limit on the number of trucks, then of course we can round up. That will still give you a solution.
However, there are multiple caveats:

- First, this isn't always possible. Sometimes there are both constraints that place lower limits and constraints that place upper limits. It can happen that taking a solution and rounding gives you something that is no longer a solution.
- Even if the result of the rounding is a solution, there is no guarantee that it is the *best* out of all solutions that uses integers. There may be some other way to choose integers for all the variables that is better than the solution you got by rounding.

In our question, if we round of the solution then we we are getting maximum capacity as 8, but in our MILP , the optimal value is coming to be 9 which is greater than that of answer coming by rounding of of the answer if the LP model. So, we won’t get the optimal value or the optimal solution by rounding off.

Q6)(R):

(i) Suppose the limit on budget is increased to 33 units and the
parking space is increased to 21 units, then we have solved the following LPP in both old and new output:

```
AMPL
ampl: include lp1.run;
Gurobi 8.0.0: optimal solution; objective 9.6
2 simplex iterations
cap = 9.6

x [*] :=
1  3.4
2  0
3  1.4
4  0
5  0
;

ampl: include lp1.run;
Gurobi 8.0.0: optimal solution; objective 21
2 simplex iterations
cap = 21

x [*] :=
1  10.5
2   0
3   0
4   0
5   0
;

ampl:
```

OLD OBJECTIVE VALUE(LPP) (say x)	9.6
NEW OBJECTIVE VALUE(LPP)(say y)	21
CHANGE IN OBJECTIVE VALUE =|x-y|	11.5

(ii) Suppose the limit on budget is increased to 33 units and the parking space is increased to 21 units, then we have solved the following MILP in both old and new output:

```
AMPL
ampl: include m.run;
Gurobi 8.0.0: optimal solution; objective 9
2 simplex iterations
1 branch-and-cut nodes
cap = 9

x [*] :=
1  3
2  0
3  1
4  1
5  0
;

ampl: include m.run;
Gurobi 8.0.0: optimal solution; objective 21
1 simplex iterations
cap = 21

x [*] :=
1  10
2   0
3   0
4   1
5   0
;

ampl: |
```

OLD OBJECTIVE VALUE(MILP) (say x)	9
NEW OBJECTIVE VALUE(MILP)(say y)	21
CHANGE IN OBJECTIVE VALUE =|x-y|	12

Q7)(R):

Now we are decreasing limit on budget from 33 units to 31, 30, 29 and 28 units. Correspondingly, we are increasing the parking space to 22, 24, 26 and 28 units. Solving the LP and MILP for each of these four possibilities, we have got the following results:

BUDGET	PARKING SPACE	OPTIMAL SOLUTION(LP)	OPTIMAL SOLUTION(MILP)
33	21	21	21

31	22	21.2	21
30	24	21.6	21
29	26	22	22
28	28	22.4	22

The common pattern that can be seen is that optimal solution in case of LP is always ≥ optimal solution in case of MILP. And , we can also see that greatest integer value of optimal solution of LP in each case is equal to the optimal solution value in case of MILP.

Exercise 2):

Q2)(R)

The given question has been solved by AMPL and the files(ex2.mod,ex2.run,ex2.dat) are attached in the zipped file

AMPL

```
ampl: include ex2.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0
Solve =  27.75 (summed over threads)
Output = 0.046875
Elapsed = 8
Gurobi 8.0.0: optimal solution; objective 295828
407360 simplex iterations
136472 branch-and-cut nodes
ampl:
```

1) The optimal solution came out to be:**295828**

Q3)(R)

Now we have to find the no of simplex iterations and time taken by the solver to solve the MILP. We have put the following command in our **AMPL** code that is attached with the question 2 and the output came as follows:

AMPL

```
ampl: include ex2.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0
Solve =  27.75 (summed over threads)
Output = 0.046875
Elapsed = 8
Gurobi 8.0.0: optimal solution; objective 295828
407360 simplex iterations
136472 branch-and-cut nodes
ampl:
```

i.e. The time taken (in seconds) to solve the MILP is:**27.75** (summed over threads)

input time(in seconds):**0**

output time(in seconds):**0.046875**

AND, The number of simplex iterations: **407360**

Q4)(R)

The given question has been solved by AMPL and the files(ex2b.mod,ex2b.run,ex2b.dat) are attached in the zipped file

```
AMPL
ampl: include ex2b.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0
Solve =  0 (summed over threads)
Output = 0
Elapsed < 1
Gurobi 8.0.0: optimal solution; objective 295896.377
20 simplex iterations
ampl: |
```

The optimal solution came out to be: **295896.377**

Q5)(R)

Now we have to find the no of simplex iterations and time taken by the solver to solve the LP. We have put the following command in our **AMPL** code that is attached with the question 4 and the output came as follows:

AMPL

```
ampl: include ex2b.run;
Gurobi 8.0.0: timing 1

Times (seconds):
Input =  0
Solve =  0 (summed over threads)
Output = 0
Elapsed < 1
Gurobi 8.0.0: optimal solution; objective 295896.377
20 simplex iterations
ampl:
```

i.e. The time taken (in seconds) to solve the LP is:**0**(summed over threads)

input time(in seconds):**0**

output time(in seconds):**0**

AND, The number of simplex iterations: **20**

Now , we have to compare the above results with those of MILP:

	INPUT TIME (SEC)	OUTPUT TIME (SEC)	SOLVING TIME (SEC)	#OF SIMPLEX ITERATIONS
MILP	0	0.046875	28.125	407360
LP	0	0	0	20

We have observed that the time taken for soling the LP problem is approximately equal to 0 but the time taken by MILP problem to solve is equal to 28.125 seconds that is very large as compared to that of LP problem . Similarly, it took only 20 simplex iterations to solve the LP problem but it took 407360 simplex iterations to solve the MILP problem.

Exercise 3):
Q1)(R)

DEFINING THE SETS THAT WILL HELP US FORMULATING THE MODEL.

Let S={0,1,2,3,4,5,6,7,8,9,10,11,12}

Where each i in S denotes the month

Now, Defining the decision variables:

Let $\mathbf{X_j}$ be the production of wire in length in the month i. Where j€S and $x_j \geq 0$ for all j€S.

For our convenience , we have defined the set S from 0, so $\mathbf{x_0=0}$ as per the question.

Now, let i_j be the inventory of each month j, where j€S.

Given that $\mathbf{i_0=0}$, $\mathbf{i_{12}=0}$, and $i_j \geq 0$ for all j€S.

DEFINING THE PARAMETERS:

Let the cost of storing the wire(per unit length per month) is denoted by INVENTORY_COST that is equal to RS 45.

As it is given that the production cannot exceed 15000, let MAX_PRODUCTION=15000.

Now we define the variable cost of each month by V_i where i€S.

i	V_i
0	0
1	250
2	300
3	700
4	500
5	450
6	190
7	200
8	400
9	200
10	250
11	350
12	150

Now we define the parameter for Demand of each month by

D_i where i€S.

i	D_i
0	0
1	2000
2	1000
3	4000
4	5000
5	500
6	3000

7	8000
8	1000
9	2000
10	4000
11	3000
12	6000

Now, defining the objective function with the help of the above definitions:

MINIMIZE cost: $\sum_{\{j \text{ in } S\}} V_j x_j$ + INVENTORY_COST*x_j.

Subject to constraints:

a)(As the demand of each month should be fulfilled)

$x_j + i_{j-1} = D_j + i_j$ for each $j \in S$ and $j \neq 0$.

b) (As the production cannot exceed 15000 units)

$x_j \leq$ MAX_PRODUCTION for each $j \in S$

(The constraints on initial inventory and final and on x_0 are defined above when we defined the variables)

Q2)(R)

The following question has been solved by AMPL and the code has been attached(ex3.mod,ex3.run,ex3.dat) in the zipped file

```
AMPL
Output = 0
Elapsed < 1
Gurobi 8.0.0: optimal solution; objective 9860000
13 simplex iterations
x [*] :=
 0      0
 1  12500
 2      0
 3      0
 4      0
 5      0
 6   3000
 7   9000
 8      0
 9   9000
10      0
11      0
12   6000
;

i [*] :=
 0      0
 1  10500
 2   9500
 3   5500
 4    500
 5      0
 6      0
 7   1000
 8      0
 9   7000
10   3000
11      0
12      0
;

ampl:
```

The optimal solution came out to be : 9860000

The production quantity in each month i is :

i	PRODUCTION QUANTITY
0	0
1	12500
2	0
3	0
4	0
5	0

6	3000
7	9000
8	0
9	9000
10	0
11	0
12	6000

Q3)(R)

For defining the variable y_t (t=1,2,3,4,5,6,7,8,9,10,11,12)

So , we'll define the following constraints:

a) y_t€{0,1}

b) $y_j \geq x_j/10000000$.

The whole model is described in the question below.

Q4)(R)

DEFINING THE SETS THAT WILL HELP US FORMULATING THE MODEL.

Let S={0,1,2,3,4,5,6,7,8,9,10,11,12}

Where each i in S denotes the month

Now, Defining the decision variables:

Let $\mathbf{X_j}$ be the production of wire in length in the month i. Where j€S and $x_j \geq 0$ for all j€S.

For our convenience , we have defined the set S from 0, so $\mathbf{x_0=0}$ as per the question.

Now, let i_j be the inventory of each month j, where j€S.

Given that $\mathbf{i_0=0, i_{12}=0,}$ and $i_j \geq 0$ for all j€S.

Now, let y_i€{0,1} for all i€S be a variable that takes values only 0 or 1 , so this variable will help us defining the new objective function.

DEFINING THE PARAMETERS:

Let the cost of storing the wire(per unit length per month) is denoted by INVENTORY_COST that is equal to RS 45.

As it is given that the production cannot exceed 15000, let MAX_PRODUCTION=15000.

Now we define the variable cost of each month by V_i where i€S.

i	V_i
0	0
1	250
2	300
3	700
4	500
5	450
6	190

7	200
8	400
9	200
10	250
11	350
12	150

Now we define the parameter for Demand of each month by

D_i where i€S.

i	D_i
0	0
1	2000
2	1000
3	4000
4	5000
5	500
6	3000
7	8000
8	1000
9	2000
10	4000
11	3000
12	6000

Now, defining the objective function with the help of the above definitions:

MINIMIZE cost: $\sum_{\{j \text{ in } S\}} V_j x_j$ + INVENTORY_COST*x_j+500000y_i.

(here 500000 is the fixed cost that is given in the question)

<u>Subject to constraints:</u>

a)(As the demand of each month should be fulfilled)

$x_j+i_{j-1}=D_j+i_j$ for each j€S and j≠0.

b) (As the production cannot exceed 15000 units)

x_j≤MAX_PRODUCTION for each j€S.

c) $y_j \geq x_j/10000000$ for all j€S.

(The value on the RHS belongs to [0,1] for all j€S. This is happening because of the constraint b, so if x_j is 0, then y_j will be 0, else y_j will be 1.)

Q5)(R)

The following question has been solved by AMPL and the code(ex3b.mod,ex3b.run,ex3b.dat) has been attached.

AMPL

```
1 branch-and-cut nodes
plus 17 simplex iterations for intbasis
cost = 12175000

x [*] :=
 0      0
 1  12500
 2      0
 3      0
 4      0
 5      0
 6  12000
 7      0
 8      0
 9   9000
10      0
11      0
12   6000
;

i [*] :=
 0      0
 1  10500
 2   9500
 3   5500
 4    500
 5      0
 6   9000
 7   1000
 8      0
 9   7000
10   3000
11      0
12      0
;
```

The value of the objective function:12175000

The production quantity in each month i is :

i	**PRODUCTION QUANTITY**
0	0
1	12500
2	0
3	0
4	0
5	0
6	12000

7	0
8	0
9	9000
10	0
11	0
12	6000

The months that have 0 productions are : 2,3,4,5,7,8,10,11

(We have neglected month zero because we defined it so that we can make the code easily)