// LINDO was used to solve the problems.

Problem #1

(a)

<Objective Function>
max dc

<Constraints>
dg = 0
dv – du <= w(u, v) for every edge from u to v, where w(u, v) is the length of edge from u to v.

<Code>

```
max dc
ST
 dg = 0
 dh - dg <= 3
 dg - de <= 7
 dd - dg <= 2
 db - dh <= 9
 da - dh <= 4
 db - da <= 8
 de - db <= 10
 dd - de <= 9
 de - dd <= 25
 df - da <= 10
 da - df <= 5
 db - df <= 7
 dc - db <= 4
 dc - df <= 3
 dd - dc <= 3
 df - dd <= 18
 de - df <= 2
END
```

<Output>

```
LP OPTIMUM FOUND AT STEP      6

        OBJECTIVE FUNCTION VALUE

        1)      16.00000
```

VARIABLE	VALUE	REDUCED COST
DC	16.000000	0.000000
DG	0.000000	0.000000
DH	3.000000	0.000000
DE	0.000000	0.000000
DD	0.000000	0.000000
DB	12.000000	0.000000
DA	4.000000	0.000000
DF	13.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	1.000000
3)	0.000000	1.000000
4)	7.000000	0.000000
5)	2.000000	0.000000
6)	0.000000	1.000000
7)	3.000000	0.000000
8)	0.000000	0.000000
9)	22.000000	0.000000
10)	9.000000	0.000000
11)	25.000000	0.000000
12)	1.000000	0.000000
13)	14.000000	0.000000
14)	8.000000	0.000000
15)	0.000000	1.000000
16)	0.000000	0.000000
17)	19.000000	0.000000
18)	5.000000	0.000000
19)	15.000000	0.000000

NO. ITERATIONS= 6

The distance of shortest path is 16.

(b)

<Objective Function>
max dc + da + db + dd + de + df + dg + dh

<Constraints>
dg = 0
dv – du <= w(u, v) for every edge from u to v, where w(u, v) is the length of edge from u to v.

<Code>

```
max dc + da + db + dd + de + df + dg + dh
ST
 dg = 0
 dh - dg <= 3
 dg - de <= 7
 dd - dg <= 2
 db - dh <= 9
 da - dh <= 4
 db - da <= 8
 de - db <= 10
 dd - de <= 9
 de - dd <= 25
 df - da <= 10
 da - df <= 5
 db - df <= 7
 dc - db <= 4
 dc - df <= 3
 dd - dc <= 3
 df - dd <= 18
 de - df <= 2
END
```

<Output>

```
 LP OPTIMUM FOUND AT STEP      9

       OBJECTIVE FUNCTION VALUE

       1)      76.00000

  VARIABLE        VALUE          REDUCED COST
       DC       16.000000          0.000000
       DA        7.000000          0.000000
       DB       12.000000          0.000000
       DD        2.000000          0.000000
```

DE	19.000000	0.000000
DF	17.000000	0.000000
DG	0.000000	0.000000
DH	3.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	8.000000
3)	0.000000	6.000000
4)	26.000000	0.000000
5)	0.000000	1.000000
6)	0.000000	2.000000
7)	0.000000	3.000000
8)	3.000000	0.000000
9)	3.000000	0.000000
10)	26.000000	0.000000
11)	8.000000	0.000000
12)	0.000000	2.000000
13)	15.000000	0.000000
14)	12.000000	0.000000
15)	0.000000	1.000000
16)	4.000000	0.000000
17)	17.000000	0.000000
18)	3.000000	0.000000
19)	0.000000	1.000000

NO. ITERATIONS= 9

The distance of shortest path from G to

A = 7
B = 12
C = 16
D = 2
E = 19
F = 17
G = 0
H = 3

Problem #2

sp = selling price
lc = labor cost
mc = material cost
s = the number of Silk tie
p = the number of Polyester tie
b = the number of Blend1 tie
c = the number of Blend2 tie

<Objective Function>
max sp - lc - mc

<Constraints>
sp = 6.7s + 3.55p + 4.31b + 4.81c
lc = 0.75s + 0.75p + 0.75b + 0.75 c
mc = (0.125 * 20)s + (0.08 * 6)p + (0.05 * 6 + 0.05 * 9)b + (0.03 * 6 + 0.07 * 9)
6000 <= s <= 7000
10000 <= p <= 14000
13000 <= b <= 16000
6000 <= c <= 8500
0.125s <= 1000
0.08p + 0.05b + 0.03c <= 2000
0.05b + 0.07c <= 1250

<Code>

```
max sp - lc - mc

ST
sp - 6.7s - 3.55p - 4.31b - 4.81c = 0
lc - 0.75s - 0.75p - 0.75b - 0.75c = 0
mc - 2.5s - 0.48p - 0.75b - 0.81c = 0
s >= 6000
s <= 7000
p >= 10000
p <= 14000
b >= 13000
b <= 16000
c >= 6000
c <= 8500
0.125s <= 1000
0.08p + 0.05b + 0.03c <= 2000
0.05b + 0.07c <= 1250
END
```

<Output>

LP OPTIMUM FOUND AT STEP 4

OBJECTIVE FUNCTION VALUE

1) 120196.0

VARIABLE	VALUE	REDUCED COST
SP	192614.750000	0.000000
LC	31668.750000	0.000000
MC	40750.000000	0.000000
S	7000.000000	0.000000
P	13625.000000	0.000000
B	13100.000000	0.000000
C	8500.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	1.000000
3)	0.000000	-1.000000
4)	0.000000	-1.000000
5)	1000.000000	0.000000
6)	0.000000	3.450000
7)	3625.000000	0.000000
8)	375.000000	0.000000
9)	100.000000	0.000000
10)	2900.000000	0.000000
11)	2500.000000	0.000000
12)	0.000000	0.476000
13)	125.000000	0.000000
14)	0.000000	29.000000
15)	0.000000	27.200001

NO. ITERATIONS= 4

Optimal profit = $120,196

Optimal numbers of ties of each type to maximize profit
silk = 7000
poly = 13625
blend1 = 13100
blend2 = 8500

Problem #3

p = the number of coin 1
n = the number of coin 5
d = the number of coin 10
q = the number of coin 25

(a)
<Objective Functions>
min p + n + d + q

<Constraints>
25q + 10d + 5n + p = 202
q >= 0
d >= 0
n >= 0
p >= 0

<Code>
```
min p + n + d + q
ST
25q + 10d + 5n + p = 202
END
GIN p
GIN n
GIN d
GIN q
```

Minimum number of used coins
10

The number of each coin
p = 2
n = 0
d = 0
q = 8

(b)
a = the number of coin 1
b = the number of coin 3
c = the number of coin 7
d = the number of coin 12
e = the number of coin 27

<Objective Functions>
min a + b + c + d + e

<Constraints>
a + 3b + 7c + 12d + 27e = 293
a >= 0
b >= 0
c >= 0
d >= 0
e >= 0

<Code>
```
min a + b + c + d + e
ST
a + 3b + 7c + 12d + 27e = 293
END
GIN a
GIN b
GIN c
GIN d
GIN e
```

Minimum number of used coins
14

The number of each coin
a = 0
b = 0
c = 2
d = 3
e = 9

Problem #4

(a)

1) The objective function is a maximization. So, we can use this.
2) Every variable is greater than or equal to 0. So, we don't need a substitution.
3) We can change the second and third inequalities to have a less-than-or-equal-to sign by multiplying -1 to both right-hand and left-hand side.

Maximize $2x1 - 6x3$
Subject to
$x1 + x2 - x3 <= 7$
$- 3x1 + x2 <= - 8$
$x1 - 2x2 - 2x3 <= 0$
$x1, x2, x3 >= 0$

4) We put slack variables to constraints to remove inequalities sign, and move all non-basic variables to left-hand side.

Maximize $2x1 - 6x3$
Subject to
$s1 = 7 - x1 - x2 + x3$
$s2 = -8 + 3x1 - x2$
$s3 = -x1 + 2x2 + 2x3$
$x1, x2, x3, s1, s2, s3 >= 0$

5) Change to slack form

$z = 2x1 - 6x3$
$s1 = 7 - x1 - x2 + x3$
$s2 = -8 + 3x1 - x2$
$s3 = -x1 + 2x2 + 2x3$
$x1, x2, x3, s1, s2, s3 >= 0$

(b)
basic variables: s1, s2, s3
non-basic variables: x1, x2, x3