Assignment11

// Objective function // Each of the x is a set of days off starting with SundayMonday // Each of the values is the cost per worker min: 775x1 + 800x2 + 800x3 + 800x4 + 800x5 + 775x6 + 750x7;

// Constraints // Each constraint is the combination of all shifts that could cover the days off excluding any which partially cover days off.

SundayMonday: x3 + x4 + x5 + x6 >= 27; MondayTuesday: x4 + x5 + x6 + x7 >= 27; TuesdayWednesday: x1 + x5 + x6 + x7 >= 26; WednesdayThursday: x1 + x2 + x6 + x7 >= 26; ThursdayFriday: x1 + x2 + x3 + x7 >= 25; FridaySaturday: x1 + x2 + x3 + x4 >= 21; SaturdaySunday: x2 + x3 + x4 + x5 >= 19;

int x1, x2, x3, x4, x5, x6, x7;

```
library(lpSolveAPI)
x <- read.lp("assignment11-4.lp")
x
```

```
## Model name:
##                      x1   x2   x3   x4   x5   x6   x7
## Minimize            775  800  800  800  800  775  750
## SundayMonday          0    0    1    1    1    1    0  >=  27
## MondayTuesday         0    0    0    1    1    1    1  >=  27
## TuesdayWednesday      1    0    0    0    1    1    1  >=  26
## WednesdayThursday     1    1    0    0    0    1    1  >=  26
## ThursdayFriday        1    1    1    0    0    0    1  >=  25
## FridaySaturday        1    1    1    1    0    0    0  >=  21
## SaturdaySunday        0    1    1    1    1    0    0  >=  19
## Kind                Std  Std  Std  Std  Std  Std  Std
## Type                Int  Int  Int  Int  Int  Int  Int
## Upper               Inf  Inf  Inf  Inf  Inf  Inf  Inf
## Lower                 0    0    0    0    0    0    0
```

```
solve(x)
```

```
## [1] 0
```

```
get.objective(x)
```

```
## [1] 33550
```

```
get.variables(x)
```

```
## [1]  4  2  9  6  2 10 10
```

The total cost is 33550 Dollars per week.

The number of workers each day meets the minimum. Each shift 1 -> 7 needs the following number of workers at a minimum 4, 2, 9, 6, 2, 10, 10.

This means that the two workers who are on shift 1 will work Tuesday -> Saturday and so on.

Total workers coming from each shift: Sunday 0 2 9 6 2 10 0
Monday 0 0 9 6 2 10 10
Tuesday 4 0 0 6 2 10 10
Wednesday 4 2 0 0 2 10 10
Thursday 4 2 9 0 0 10 10
Friday 4 2 9 6 0 0 10
Saturday 4 2 9 6 2 0 0

Totals: Sunday - 29 Monday - 37 Tuesday - 32 Wednesday - 28 Thursday - 35 Friday - 31 Saturday - 23