Problem Statement:

Airline crew scheduling process begins with the daily crew-pairing optimization problem. In the daily problem, all flight segments (flight legs) are assumed to be flown every day. After solving the daily problem, adjustments are made for weekly exceptions and crew base balancing. Once pairings are found that exactly cover every flight leg for the month, bid lines or rosters are made up for the month and are assigned by a bidding process to crew member. The purpose of the airline crew pairing problem is to generate a set of minimal cost crew pairings covering all flight segments.

Input:

1. List the set of flight segments to be covered.
2. Departure and arrival time of each flight segments along with its originating and terminating stations.
3. List crew base stations (stations were crew has to return by the end of day work).

Duty:

A sequence of flights that can be flown by a single crew over the course of a work day is a duty period. Flights are grouped together to form duty periods, which are series of sequential flight segments comprising a day's work for a crew. For a duty the crew member need not to return to the base station for duty formation. Duties are constrained by a number of restrictions.

a) A minimum of one flight segment can also act as a duty.
b) A single duty can have maximum three flight segments.
c) Minimum 1 hour rest provided between two segments.
d) Maximum length of a duty should be 8 hours including rest between segments.
e) Each flight segment can act as a duty.
f) Combine two flight segments to form a duty.
g) Then combine three flight segments to form a duty.

Crew Pairing:

A pairing is a sequence of duties that begin and end at a crew base such that in a sequence the arrival city of a duty coincides with the departure city of the next duty. The objective is to find a subset of these pairings with minimal cost that covers all the flight segments in the schedule exactly once.

a) The crew should return to the base station within a day (24 hours)
b) A maximum of two duties are combined to form a crew pairings.

Cost:

The cost of a pairing is generally a complex function of:

- flying time per duty (minimum guaranteed)
- span of each duty
- span of the pairing
- deadhead costs
- accommodation fees (for rests outside the base)

In this problem the cost is directly proportional to the time span of the pairing with proportionality constraint equals to 1.

Problem Formulation:

Number of decision variables = Number of crew pairings (x_1, x_2,…….., x_n)

Number of constraints = Number of flight segments

$$Min \sum_{k=1}^{K} c_k x_k$$

Subject to

$$\sum_{k=1}^{K} R(k)\, x_k = 1$$

where

$R(k)$ is a 0-1 vector with components $R_i(k)$ $(i = 1, 2, \ldots.., NF)$

$NF =$ No. of flight segments

$$R_i(k) = \begin{cases} 1 & if\ flight\ segment\ i\ belongs\ to\ crew\ pairing\ k \\ 0 & otherwise \end{cases}$$

$$x_k = \begin{cases} 1 & if\ the\ crew\ pairing\ k\ is\ selected\ in\ the\ solution \\ 0 & otherwise \end{cases}$$

Algorithm:

Step 1: Start

Step 2: Read the input file ‘input.txt’.

Step 3: Declare each flight segments, number of stations, base stations, each flight originating and terminating time as well as stations.

Step 4: Form the duties combining flight segments using duty formation constraints defined above. Combine each base duty to the list of duty of lengths n to create a new list of duties of length n+1. At any step, new duties will store duties of length n. length of a duty means number of trips it is composed of. Repeat till we cannot construct any more new duty.

Step 5: Combine duties to form crew pairings. Each crew pairing should follow crew pairing restrictions.

a) Check for overlap of duties in forming crew pairing.
b) Check for duplication of pairings and remove if found.

Step 5: Calculate each crew pairing cost *(c_k)*.

Step 6: Declare each crew pairings as variables x_1, x_2, x_3, ……………. x_n.

Step 7: Define the objective function and constraints of the master problem.

Step 8: Run the master problem.

Step 9: Display optimal solution and optimal value.

Step 10: Stop