In this part of the project, you are asked to develop a routing tool for the standard-cell design style.

**Overview:**
The routing of the circuit is performed based on the results of the placement stage you have implemented in the second part of the project. In this stage, the geometrical configuration of nets (channel number, track number, left position and right position) connecting distinct components will be specified. This step concludes the physical design of a circuit.

*Routing model:* The routing of a standard-cell design consists of the handling of two groups of nets: **feed-throughs that cross multiple channels** and **nets within each individual channel**. For the routing of the regular nets (nets within each channel), a two-layer VH model has been utilized in the design. In order to minimize the impact of feed-throughs, the fabrication technology targeted by this tool provides a separate metal layer for the vertical wires of the feed-throughs. The horizontal wires of the feed-throughs still share the same layer with the horizontal wires of the regular nets. As a result, the routing of the feed-throughs does not impose any vertical constraints on the routing of regular nets, whereas its impact on horizontal constraints of each channel needs to be considered.

*Feed-through routing:* Due to the dependence of regular net routing on feed-through routing outlined above, the routing of the feed-throughs needs to be performed first. The problem that needs to be solved at this step is the specification of the channel in which the horizontal wire of each feed-through will be placed. In a hierarchical modeling of this problem, one could think of each channel as a big track, resulting in an $N$-row standard-cell layout to be considered as a single, overarching channel consisting of $N-1$ tracks, with feed-throughs envisioned as the nets to be routed within the big channel. By solving a channel routing problem for the feed-throughs, the assignment of the horizontal feed-through wires into channels can be attained.

*Regular-net routing:* After the feed-through routing phase, the track assignment of the wires within each individual channel needs to be specified. This is attained by performing a conventional channel routing algorithm. The only particularity in this subproblem is that, there might be several horizontal feed-through wires in each channel. Although those wires have no contribution to the vertical constraints, their impact on horizontal constraints needs to be considered for attaining an appropriate routing.

*Core routing algorithm:* Since both the feed-through and regular-net routing phases require the resolution of a channel routing problem, the Yoshimura-Kuh algorithm will be used as the core algorithm in this part of the project.

*Feed-through routing*
The problem of routing feed-throughs in a standard-cell design is modeled as a channel
routing problem, as illustrated in Figure 1. Each channel is modeled as a big track within a single channel. Therefore, the assignment of the horizontal wires of the feed-throughs can be resolved by performing a channel routing. The Yoshimura-Kuh algorithm is used to solve the channel routing problem based on vertical and horizontal constraints, as detailed in the textbook. However, there is a small difference between the modeled problem and the traditional channel routing problem. Since the terminals of feed-through nets may appear in intermediate channels, the terminal positions in the routing model may not necessarily be at the top or bottom edge of the channel. Consequently, the candidate tracks for distinct feed-through nets may vary depending on their terminal positions. Taking Figure 1 as an example, the range of candidate tracks for net 1 is tracks 1&2, whereas the range for net 2 is track 2&3. The track assignment must fulfill this additional constraint for achieving the optimal routing of feed-throughs.

As you can observe from the model, the feed-through routing stage only decides which channel to assign the horizontal wires of the feed-throughs. After the channel assignment has been concluded, these horizontal wires can be regarded as regular nets within each individual channel. Hence the concrete positions (tracks) of these wires within the channels will be specified in the next step, the channel routing of regular nets.

**Regular-net routing**
This step generates the track assignment of the nets for each individual channel. Since the vertical wires of the feed-throughs are fabricated in a separate metal layer, such wires have no impact on the routing within each specific channel. The horizontal wires of the feed-throughs will share the same horizontal routing layer with the regular nets, thus contributing only to the horizontal constraint graph. The Yoshimura-Kuh algorithm should be used again to generate the concrete routing within each channel based on its corresponding vertical and horizontal constraints.

**Data structure**
The routing tool needs to parse the connectivity matrix (defined in previous parts of the project) that specify the pair-wise connections between components. Figure 2 shows an example circuit and its wire specifications.
However, some wires may share the same net in the layout. For example, there are 5 nets in the circuit in Figure 2. Therefore, you will need to find the nets and their positions based on the result generated by the placement tool you implemented in Part 2 of the project. The following observations for this example may be illustrative. Wire 2 and Wire 4 are parts of the same net. Wire 2 connects port 3 of cell 3 to port 1 of cell 2, and Wire 4 connects port 2 of cell 4 to port 1 of cell 2. Therefore, they are on the same net.

You will need possibly to use a data structure in which every net is represented as a set of all its pins plus the leftmost and rightmost position of the net. The following structure is an illustrative data structure for this. It represents the nets in the given circuit. The pin positions follow the assumption from the second project - the pins are placed on every third position inside the cell. For this example the position of port 1 of cell 0 is 23, because cell 0 is the third cell on row 0 and the size of the two preceding cells, i.e., cells 3 and 5, is 10 (10+10+3=23).

Net (pin position1, row), (pin position2, row), ... [Leftmost position of the net, rightmost position of the net]
Net 1: (6,0) (23,0) [6,23]
Net 2: (9,0) (3,1) (26,1) [3,26]
Net 3: (29,1) (13,1) [13,29]
Net 4: (16,0) (3,0) [3,16]
Net 5: (19,0) (23,1) [19,23]

Since no pin information is available for horizontal feed-through wires, we only need to specify the channel within which it should be placed and its leftmost and rightmost positions. Thus a value of -1 is assigned to all of its pin positions to indicate that it is a horizontal feed-through wire. For example, Net 6 specified below is a horizontal feed-through wire within the first channel (between Row 0 and Row 1) with its leftmost and rightmost positions being 8 and 15, respectively.

Net 6: (-1,0) (-1,1) [8,15]
The following shows the port positions in one channel, corresponding to the above 6 nets.
0040010020000004005000010000000000
00000006000000600000000000000
00200000000300000000000005002003

The first and third rows correspond to the pin positions of the regular nets. The middle row shows the horizontal span of the horizontal feed-through wires.

The data structure outlined above can be constructed for each channel, and be fed to the Yoshimura-Kuh algorithm to generate the detailed routing.

**Handling of exception cases**
Since the number of channels is fixed, it is possible (but highly unlikely in our opinion) that the number of big tracks (channels) required for fulfilling the vertical and horizontal constraints of the feed-throughs exceeds the actual channel quantity. This will lead to a failure in the feed-through routing stage. If such a case is encountered, your program should output a message indicating the problem and enter directly the regular net routing stage. The routing process should then concentrate only on the regular nets and ignore any feed-through wires.

Your program needs to also search for possible loops in the vertical constraint graph that can not be handled by the Yoshimura-Kuh algorithm. Once a loop is detected during the routing of any channel, a message indicating the loop problem should be output, and the algorithm should immediately proceed to the routing of the next channel.

**Output Specification:**
For each channel, your program should first output the components in its adjacent rows. The list of the nets together with their port positions, start and end position of the net, and the track number should be output. More formally the format of the file is:

Net i : (PinPosition1, PinPosition2, ...) [LeftmostPosition, RightmostPosition] TrackNumber

---- Output for the first channel starts here ----
Row 0: 3 5 0
Row 1: 2 1 4

Net 1: (6, 23) [6,23] 3
Net 2: (9, 3, 26) [3,26] 1
Net 3: (29, 13) [13,29] 4
Net 4: (16, 3) [3,16] 2
Net 5: (19, 23) [19,23] 2
Net 6: (-1, -1) [8, 15] 5

---- Output for the first channel ends here ----
The above results correspond to the following visual representation of the routed channel.

```
0123456789012345678901234567890
--4--1--2-------4--5---1-------
-----------------------------
-----------------------------
-----------------------------
-----------------------------
--2---------3---------5--2--3-
```

**What to turn in:**

1. A file including your program together with the results of your program for the two example circuits given in the writeup of Project 2. Your program output should follow the format specified above.
2. A document detailing how you implement the algorithm in your code.
3. Your program source code files and a README file that explains how to compile and run your program.

**Turn-in Method:**

Send all requested files to the TA via email. The email must be time-stamped by the deadline.