



Write a program to book train tickets as per the details given below:

The program should first present a menu with the following options:

1. Book Ticket
2. Cancel Ticket
3. Print All
4. Print Availability
5. Exit

Details and conditions that are common

- The train has only one coach with 16 berths (4 Lower, 4 Middle, 4 Upper, 2 Side Upper and 2 Side Lower)
- 14 confirmed tickets will be allotted a berth (4 Lower, 4 Middle, 4 Upper, 2 Side Upper)
- The remaining 2 Side Lower berths are issued as 4 RAC tickets (Reservation against Cancellation)
- No ticket for infants below 5 years of age
- Wait list tickets should be issued for bookings beyond the confirmed 14 and 4 RAC. There can only be 2 waitlist tickets

Book Ticket

- When Book Ticket is chosen
 - It should prompt for the passenger details like:
Enter Name :
Enter Age :
Enter Preference :
Do you want to book one more ? (Y/N) :
 - Kids/infants below age of 5 should also be accepted the same way.
 - After entering the details of all the passengers in the group, it should print the ticket.
- Conditions for booking a ticket
 - When no berth preference is given for a passenger, the following rules should be applied:
 - Passengers with kids should be given a lower berth.
 - Passengers with age >60 should be allotted a lower berth, if available

- If the preferred berth is not available, get confirmation from the passenger before booking the ticket.
- Passengers travelling in groups should have the seats allotted together or placed close to each other
- A ticket should be generated with a PNR number (can be a random unique number) and the seats allotted with the status
- Print the details of the ticket after booking.

Cancel Ticket

- Cancel Ticket is based on the PNR number.
- After the PNR Number is provided, it should list all the passengers in that ticket as
1 Name1, Age1, Berth No.
2 Name2, Age 2, Berth No.
and so on
- The serial numbers of the passenger whose ticket has to be cancelled is taken as the input.
- When a ticket is cancelled, the passengers in RAC, if any, will be allotted that berth and all the other RAC and Waitlist passengers move up in their current position. For example, if 1 confirmed ticket is cancelled, RAC 1 will get a confirmed berth, RAC 2 will become RAC 1 and WL 1 will become RAC 4.

Print All

- Print All should print the details of all the booked tickets and the details of the vacant berths. For example
 - Berth No, PNR, Name, Age
1, 1234, Name1, 25
2, vacant
3, 1235, Name2, 22
4, 1235, Name3, 22
and so on

Print Availability

- Should print all the vacant berths with summary. For example
 - 5 Berths and 4 RAC tickets available
 - Berth No, Status
4, vacant
7, RAC vacant
and so on



- Java or C can be used for writing the programs.
- Create a folder in the Desktop with your name and department and save the source files there. Example folder name Arun_CSE, Geetha_MCA etc...

1. There is a school with 100 students, and correspondingly 100 lockers, all of which start off closed. The first student opens every locker. The second student closes every other locker, starting with the second(2,4,6,etc...). The third student changes the state of every third locker, starting with the third(3,6,9 etc...). The fourth would change the status of lockers numbered 4,8,12 etc... That is, if the locker is open, it is closed, and if it is closed, it is opened. This continues until all 100 students have passed along the lockers. After the 100th student is done, which lockers are open and which are closed?
[Note : Program should work for any number of students/lockers]

2. Write a program to accept two strings S1 and S2 and reverse the words of S1, starting from the word where the first occurrence of S2 present in S1. Same empty spaces between the words must be maintained in the output. Write the program without splitting up the strings into array of words.

Note: If you are writing this program in JAVA, don't use built-in functions like split(), indexOf(), replace(), substring(), etc present in String Class.

Examples:

Input: S1=This is a test input string S2=st

Output: This is a string input test

3. Given an (m x n) matrix, write a program to traverse the cells and print the values present in the given path. Include necessary validation and proper error messages in case of given path has out of bounds.

5x5 matrix:

[1 2 3 4 5] (row 1)

[6 7 8 9 0] (row 2)

[1 2 3 4 5] (row 3)

[6 7 8 9 0] (row 4)

[1 2 3 4 5] (row 5)

Path Notations: ">" is going right, "v" going down, "<" is going left, "^" is going up.

Example Input 1:

Start at (Row, Column): 1,2

Path: > > > v

Output: 2 3 4 5 0

Example Input 2:

Start at (Row, Column): 2,3

Path: v > > v < < ^ > > v v

Output: 8 3 4 5 0 9 8 3 4 5 0 5

Example Input 3:

Start at (row, Column): 1,4

Path: > v > >

Output: Invalid Path

4. You are given n pairs of numbers. In every pair, the first number is always smaller than the second number. A pair (c, d) can follow another pair (a, b) if $b < c$. Chain of pairs can be formed in this fashion. Find the longest chain which can be formed from a given set of pairs.

Examples:

Input: {{5, 24}, {39, 60}, {15, 28}, {27, 40}, {50, 90}}

Output: Length=3, Pairs={{5, 24}, {27, 40}, {50, 90}}

Input: {{20, 30}, {12, 25}, {12, 20}, {2, 10}, {15, 25}, {25, 40}, {40, 60}}

Output: Length: 4, Pairs= {{2, 10} {12, 20} {20, 30} {40, 60}} {{2, 10} {12, 25} {25, 40} {40, 60}}

5. Given array of words, group the anagrams and print. Any word or phrase that exactly reproduces the letters in another order is an anagram. Arrive most efficient algorithm.

Examples:

Input: {tar, rat, banana, atr, nanaba}

Output: Anagrams: {[tar, rat, atr], [banana, nanaba]}

Input: {abc, cde, xyz, dec}

Output: Anagrams: {[cde, dec]}, Others: {[abc], [xyz]}

Input: {a, bc, c}

Output: Anagrams: {}, Others: {[a], [bc], [c]}
