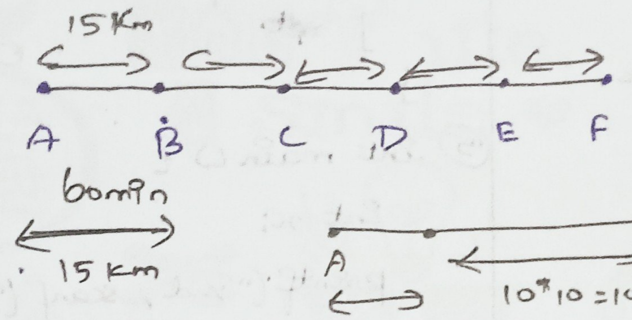
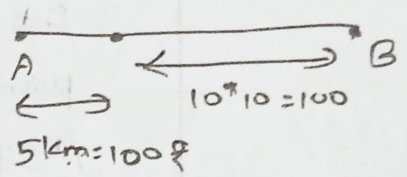
Taxi Booking System:

There are n no. of. taxi's. For simplicity, assume 2

But it should work for any no. of Taxi'



Railway Time



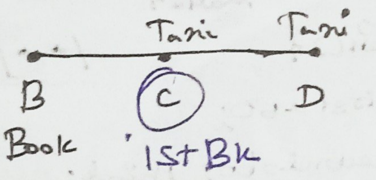
5km=100 ₹

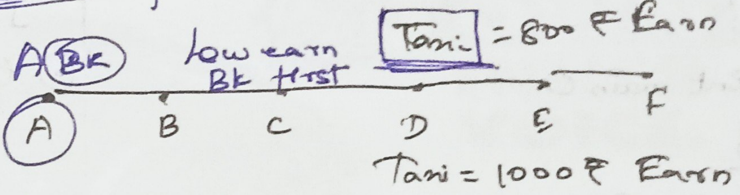
N.xt 1 km = 10₹

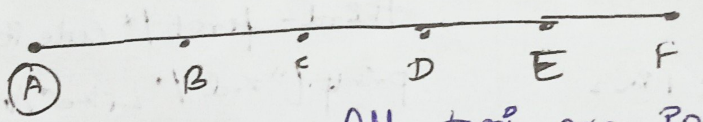
10 km = 100

Initial from A to nxt

D person bk car in D first book







All taxi are in ride

Booking Rejected

Coding: Taxi.java

```
util.*;
Class Taxi
    int id -> Unique id
    Char current = 'A' -> Name
    int earning = 0 -> 0 Balance
Array List (Size Not defined & maintain order)
Access index easily)
    List<Bk> booking = new Arraylist<>();
  public taxi (int id) {
        this. id = id;          ] Constructor
  }
```

```
public boolean isava (int reqtime){                Time:17
    if (booking.empty()) return true;             Bking empty then
                                                    Bk taxi
    Booking lastBking = Bkng.get (bkng.size() -1);
    return lastBking.droptime <= reqtime;
                        17        <= 16  ⟶  Cancel Bking
Calculate Earning
  public int calcearn (char from, char to) {      loc  From - To
                                                       A - C
      int distance= Math.abs (to -from) * 15;     C-A = 2 * 15 = 30
                                                       30 km
      return 100 + Math.max (0, (distance - 5) * 10);
                                                  100 + 25 * 10
                                                  100 + 250
Add Bking                                         Amnt 350
  public void addBking (Bking Booking) {
      bks.add (Booking);
      totalearn += Booking.amount;
      current = Booking.to;
  }
}
New class  Booking.java          Contains Booking details
Class Booking {
   int bkid, customerid, pickuptime, droptime, amnt;
   char from, to;
   public Booking (int Bkid, int customerid, int . . . . amnt)
                                       char from, char to
       this.Bkid = bookingid;
         "
         "
         :
       this.amnt = amnt;
```

Final class Taxi Booking. java

```
util.*;
class TaxiBkingSystem{
  Static List <Taxis> taxis = new ArrayList<>();
  Static Scanner Sc = new Scanner(System.in);
  Static int customercounter = 1;
  public static void main (String[] args){
    S.o.p ("Enter no of taxis: ");
    int numTaxis = Sc. nextInt(); //2   A-C
    initializeTaxis (numTaxis);
  Conditions for taxi reservation =>
    while (true)
      S.o.p ("\n1. Bk Taxi \n2. Display Taxi details \n3.Exit");
      S.o.p ("Enter your choice: ");
      int Choice = Sc. nextInt ();
      Switch (choice)
        Case 1: bookTaxi(); Break;
        Case 2: displayTaxidetails (); break;
        Case 3: S.Exit (); break;
        default: S.o.p ("Invalid. Try again");
Initialize Taxis
  public static void initializeTaxis (int n){.
    for (int i=1; i<=n; i++)
      taxis.add (new Taxis(i));
  Public static void book taxi(){
    int customerid = customercounter++;
  [ S.o.p ("Enter Pickup Point (A-F)");
    char pickup = Sc. next ~~Int~~ ().toUpperCase().charAt(0);
  Repeat
```

~~for~~ Also Drop Point (A-F)
Enter Pickup Time (in hrs):

```
int pickuptime = Sc.nextInt();

Taxi selectedTaxi = null;
int mindistance = Integer.MIN_VALUE;   → Minimum distance
    min                                   for eg ✓
                                          Ⓐ BK D or E
for (Taxi taxi : taxis){
  if(taxi.isAvailable (pickup time))
     int dis = Math.abs(taxi.current - pickup);
     if (dis ~~<~~ min ||(dis == min && taxi.Earning < SelectedTaxi.
                                                        earning)) ;

     SelectedTaxi = taxi;
     min = dis;
  if (selected Taxi == null)
     S.o.p("No taxis Available, Bkng rejected);
     return;                                          A - B = ③
}                                          9+3=12   ③     1
       12                    9am
int droptime = pickuptime + Math.abs(drop - pickup);
int amnt = Selected Taxi . calculate earning(pickup, drop);
int bkingId = SelectedTaxi . bookings. size() + 1;

Booking booking = new Booking (bkid, customerid, pickup, drop,
                               pickuptime, droptime, amnt);

Selected Taxi. add Bking (booking);
S.o.p ("Taxi - " + selected Taxi.id + " is.allocated.");
```

display Taxi details

```
public static void displaytaxidet(){        Repeat same details for
                                              Booking booking : taxi.bookings
 for (Taxi Taxi : taxis){
  S.o.p (" Taxi-"+ taxi.id + "Total Earnings : Rs !" + taxi.Earning )
  S.o.p("%-10S %-10S  %-5S  %-5S %-12S  %-9S  %-6S%n"),
      bkid to ... amnt)  booking. bkid . . .  booking .amnt)
```