

class Player {  
 private int noOfMatches;  
 private int pointsEarned;  
  
 public Player(int noOfMatches){  
 this.noOfMatches=noOfMatches;  
 }  
  
 public int getPointsEarned(){  
 return this.pointsEarned;  
 }  
  
 public void setPointsEarned(int pointsEarned){  
 this.pointsEarned = pointsEarned;  
 }  
  
 public int getNoOfMatches(){  
 return this.noOfMatches;  
 }  
  
 //To\_Trainees  
 public Boolean validateNoOfMatches(){  
 //write your logic here  
 boolean x= false;  
 if (getNoOfMatches()>0 && getNoOfMatches()<=100)  
 {  
 x= true;  
 }  
 else x= false;  
 //change return statement accordingly  
 return x;  
 }  
  
 @Override  
 public String toString() {  
 return "Player (noOfMatches=" + this.noOfMatches + ")";  
 }  
}  
  
class Batsman extends Player {  
 private int runsScored;  
 private int centuryCount;  
 private int batsmanRank;  
  
 public Batsman(int noOfMatches,int runsScored,int centuryCount){  
 super(noOfMatches);  
 this.runsScored = runsScored;  
 this.centuryCount = centuryCount;  
 }  
 public int getRunsScored(){  
 return this.runsScored;  
 }  
  
 public int getCenturyCount(){  
 return this.centuryCount;  
 }  
  
 public int getBatsmanRank(){  
 return this.batsmanRank;  
 }  
  
 @Override  
 public String toString() {  
 return "Batsman (Player (noOfMatches=" + this.getNoOfMatches() + ")+runsScored=" + this.runsScored + ", centuryCount="  
 + this.centuryCount + ")";  
 }  
  
 //To\_Trainee  
 public Boolean validateBatsmanRecord(){  
 //write your logic here  
 if(centuryCount>=0 && centuryCount<=getNoOfMatches())  
 {  
 int totalRun = centuryCount\*100;  
 if(totalRun <=this.runsScored) {  
 return true;  
 }  
 }  
  
 //change return statement accordingly  
 return false;  
 }  
  
 //To\_Trainees  
 public void calculatePoints(int index){  
  
 //write your logic here  
 if(!super.validateNoOfMatches() || !validateBatsmanRecord() )  
 {  
 setPointsEarned(-1);  
 batsmanRank=-1;  
 }  
 else {  
 setPointsEarned((runsScored\*2)+(centuryCount\*25));  
 int rank = PlayerDetails.*rankPlayer*(getPointsEarned(), index);  
 batsmanRank = rank;  
 }  
  
 }  
}  
  
class PlayerDetails {  
 static int [] *playersPointsArr*={1000,934,800,550};  
  
 public static void swap(int[] numbers, int firstIndex, int secondIndex) {  
 int temp = numbers[firstIndex];  
 numbers[firstIndex] = numbers[secondIndex];  
 numbers[secondIndex] = temp;  
 }  
  
 public static int[] sort(int[] pointsArr){  
 for(int index2=0;index2<PlayerDetails.*playersPointsArr*.length;index2++) {  
 boolean swapped = false;  
 for(int index3=0;index3<(PlayerDetails.*playersPointsArr*.length- index2 - 1);index3++) {  
 if (PlayerDetails.*playersPointsArr*[index3] < PlayerDetails.*playersPointsArr*[index3 + 1]) {  
 *swap*(PlayerDetails.*playersPointsArr*, index3, index3 + 1);  
 swapped = true;  
 }  
 }  
 if (swapped == false)  
 break;  
 }  
 return pointsArr;  
 }  
  
 //To\_Trainees  
 public static Integer rankPlayer(int pointsEarned, int index){  
 //write your code here  
 *playersPointsArr*[index]= pointsEarned; //update poitsEarened  
 //change return statement accordingly  
 *sort*(*playersPointsArr*);// sort playerPointsArr  
 for(int i=0;i<*playersPointsArr*.length;i++) {  
  
 if(*playersPointsArr*[i]==pointsEarned) {  
 int playerRank = i+1;  
 return playerRank;  
 }  
 }  
  
 return null;  
 }  
  
}  
  
class Tester {  
 public static void main(String[] args){  
 Batsman obj1 = new Batsman(10, 420, 3);  
 obj1.calculatePoints(0);  
  
 System.*out*.println("Batsman Points:" + obj1.getPointsEarned());  
 System.*out*.println("Batsman Rank:"+ obj1.getBatsmanRank());  
 System.*out*.println("Player Points Array:");  
 for(int index=0; index < PlayerDetails.*playersPointsArr*.length;index++){  
 System.*out*.print(PlayerDetails.*playersPointsArr*[index]+" ");  
 }  
  
 }  
}