Salary Calculation

Problem Statement:

You are required to write a Java program to calculate the total salary of an employee based on their hourly wage, hours worked in a week, and the number of weeks they worked. The program should consider the following rules:

- If an employee works more than 40 hours in a week, they are paid 1.5 times their hourly wage for the overtime hours.

- If an employee works less than 20 hours in a week, they are penalized with a deduction of 10% of their weekly salary.

- The program should handle invalid inputs (e.g., negative values for hours or wages).

```java
import java.util.Scanner;

public class SC {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double wage = sc.nextDouble();

if (wage < 0) {

System.out.println("Hourly wage cannot be negative.");

return;

}

int hPerWk = sc.nextInt();

if (hPerWk < 0) {

System.out.println("Hours per week cannot be negative.");

return;

}

int wks = sc.nextInt();
```

```java
if (wks < 0) {

System.out.println("Weeks worked cannot be negative.");

return;

}

double wkSal;

if (hPerWk > 40) {

double regHrs = 40;

double ovtHrs = hPerWk - 40;

wkSal = (regHrs * wage) + (ovtHrs * wage * 1.5);

} else {

wkSal = hPerWk * wage;

}

if (hPerWk < 20) {

wkSal *= 0.9;

}

double totSal = wkSal * wks;

System.out.printf("Total salary is %.1f\n", totSal);

}

}
```

Problem Statement:

You are required to calculate the total cost of purchasing tickets for an event based on the ticket type and the number of tickets bought.

The program should consider the following rules:
- Regular Ticket: 50 each. If more than 10 tickets are bought, a discount of 10% is applied.
- VIP Ticket: 100 each. If more than 5 tickets are bought, a discount of 15% is applied.
- Premium Ticket: 150 each. If more than 3 tickets are bought, a discount of 20% is applied.
- If the total cost before any discount is less than 200, an additional service fee of 20 is applied.
- The program should handle invalid inputs (e.g., negative values for number of tickets, or invalid ticket types).

```java
import java.util.Scanner;

public class BG {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter ticket type (Regular, VIP, Premium): ");

String type = sc.nextLine().trim();

System.out.print("Enter number of tickets: ");

int qty = sc.nextInt();

if (qty < 0) {

System.out.println("Number of tickets cannot be negative.");

return;
}

double pricePerTicket = 0.0;

double discount = 0.0;

switch (type) {
```

```java
case "Regular":

pricePerTicket = 50.0;

if (qty > 10) discount = 0.10;

break;

case "VIP":

pricePerTicket = 100.0;

if (qty > 5) discount = 0.15;

break;

case "Premium":


pricePerTicket = 150.0;

if (qty > 3) discount = 0.20;

break;

default:

System.out.println("Invalid ticket type.");

return;


}

double totalCost = qty * pricePerTicket;

totalCost -= totalCost * discount;

if (totalCost < 200) totalCost += 20;

System.out.printf("%.1f\n", totalCost);

}

}
```

```
E:\230701297>javac BG.java

E:\230701297>java BG
Enter ticket type (Regular, VIP, Premium): Regular
Enter number of tickets: 12
540.0
```

Largest and smallest digit of a number

Problem Statement:

Given a number N. The task is to find the largest and the smallest digit of the number.
Input Format:
A positive number in the range 1 <=n<=10000
Output Format:
Print the largest digit and the smallest digit

```java
import java.util.Scanner;

public class LSN{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int num = sc.nextInt();

int min = 9, max = 0;

while(num != 0){

int rem = num % 10;

if(rem < min) min = rem;

if(rem > max) max = rem;

num /= 10;

}

System.out.println(min + " " + max);

}
}
```

```
E:\230701297>javac LSN.java

E:\230701297>java LSN
37827
2 8

E:\230701297>java LSN
6
6 6
```