Recursion in java

Assignment Questions

Assignment Questions

1. **Given an integer, find out the sum of its digits using recursion.**

   Input: n = 1234
   Output: 10
   Explanation: 1 + 2 + 3 + 4 = 10

   Ans :-

```java
import java.util.*;
public class Assignment_Q_1 {

    public static int digitSum(int n){
        if(n == 0){
            return 0;
        }
        else{
            return (n % 10) + digitSum(n/10);
        }
    }
    public static void main(String[]args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number which sum of digits you want");
        int n = sc.nextInt();

        int ans = digitSum(n);

        System.out.println(ans);

    }
}
```

   Output :-

Enter the number which sum of digits you want
1234
10

2. **Given a number n. Find the sum of natural numbers till n but with alternate signs. That means if n = 5 then you have to return 1 - 2 + 3 - 4 + 5 = 3 as your answer.**

   Constraints: 0 <= n <=le6
   Input1 : n = 1 * 0
   Output 1:-5
   Explanation: 1 - 2 + 3 - 4 + 5 - 6 + 7 - 8 + 9 - 10 = - 5
   Input 2 / n = 5
   Output 2:3

   Ans :-

```java
import java.util.*;
public class Assignment_Q_2 {

    public static int alternateOperation(int n){
        if(n == 1){
            return 1;
        }
        if (n%2 == 0){
            return alternateOperation(n-1)-n;
        }
        else {
            return alternateOperation(n-1)+n;
        }
    }
    public static void main(String[]args){
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter a number");
        int n = sc.nextInt();

        int ans = alternateOperation(n);
        System.out.println(ans);
    }
}
```

   Output :-

   Enter a number
   10

-5

3. **Print the max value of the array [13, 1, -3, 22, 5].**

   Ans :-

```
public class Assignment_Q_3{
    public static void main(String[]args){
        int ar[] = {13, 1, -3, 22, 5};
        int n = ar.length;
        int maxInd = 0;
        for(int j = 0; j<ar.length; j++){
            if(ar[maxInd] < ar[j]){
                maxInd = j;
            }
        }
        System.out.println("The maximum value is : " + ar[maxInd]);
    }
}
```

   Output :-

   The maximum value is : 22

4. **Find the sum of the values of the array [92, 23, 15, -20, 10].**

   Ans :-

```
public class Assignment_Q_4 {
    public static void main(String[]args){
        int a = 0;
        int ar [] = {92, 23, 15, -20, 10};
        for(int i = 0; i<ar.length; i++){
            a += ar[i];
        }
        System.out.println("The sum of all the elements present in this array is "
+ a);
    }
}
```

   Output :-

   The sum of all the elements present in this array is 120

5. **Given a number n. Print if it is an armstrong number or not. An armstrong number is a number if the sum of every digit in that number raised to the power of total digits in that number is equal to the number. Example : 153 = 1 ^ 3 + 5 ^ 3 + 3 ^ 3 = 1 + 125 + 27 = 153 hence 153 is an armstrong number.**

   Input1: 153
   Output1: Yes
   Input 2: 134
   Output2: No

   Ans :-

```
import java.util.*;
public class Assignment_Q_5 {

    public static int powerFinder(int p, int count){

        if (count == 0){
            return 1;
        }
        else{
            return  p * powerFinder(p,count-1);
        }

    }

    public static void main(String[]args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number which you have check it is Armstrong
or not");
        int n = sc.nextInt();
        int a = n;
        int c = n;
        int count = 0;
        int p = 0;

        while (a!=0){
            a = a/10;
            count++;
        }
        int z = 0;
        for(int i = 0; i<count; i++){
            p = c%10;
```

```
            z = z + powerFinder(p,count);

            c = c/ 10;
        }

        if(z == n){
            System.out.println("Yes");
        }
        else{
            System.out.println("No");
        }

    }

}
```

Output :-

Enter the number which you have check it is Armstrong or not
153
Yes