

Batch:A2 Roll No.: 16010123032

Experiment / assignment / tutorial No.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of the Staff In-charge with date

TITLE : Control Statements

AIM:

Write a Java program to generate and show all Kaprekar numbers less than 1000.
In number theory, a Kaprekar number for a given base is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because $45^2 = 2025$ and $20 + 25 = 45$.

Expected OUTCOME of Experiment:

CO1:Apply the features of object oriented programming languages. (C++ and Java)

CO2:Explore arrays, vectors, classes and objects in C++ and Java

Books/ Journals/ Websites referred:

1. E. Balagurusamy, "Programming with Java", McGraw-Hill.
2. E. Balagurusamy, "Object Oriented Programming with C++", McGraw-Hill.

Pre Lab/ Prior Concepts:

Java basic constructs (like if else statement, control structures, and data types)
Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of

statements multiple times and following is the general form of a loop statement in most of the programming languages –

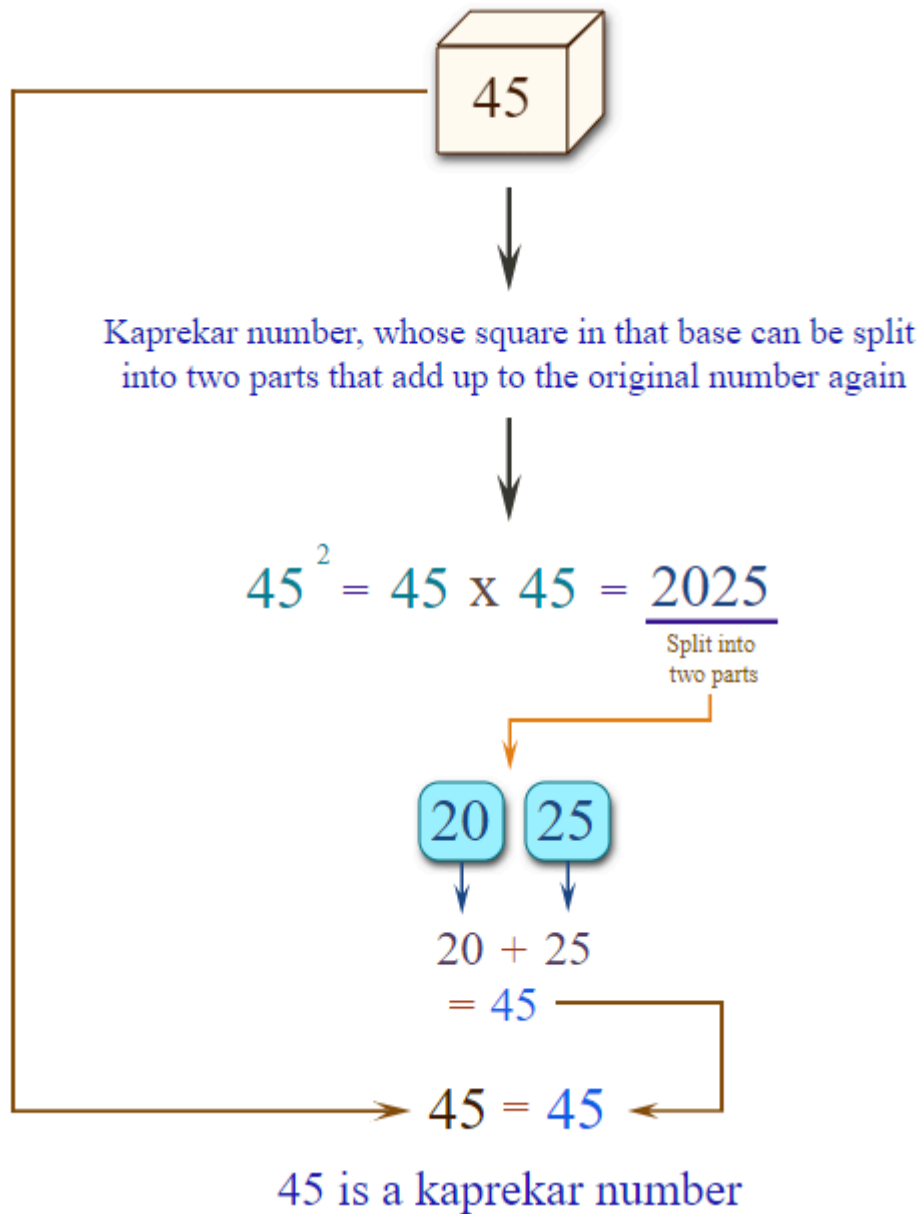
Sr.No.	Loop & Description
1	<u>while loop</u> Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
2	<u>for loop</u> Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
3	<u>do...while loop</u> Like a while statement, except that it tests the condition at the end of the loop body.

Loop Control Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Java supports the following control statements. Click the following links to check their details.

Sr.No.	Control Statement & Description
1	<u>break statement</u> Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.
2	<u>continue statement</u> Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.



In number theory, a Kaprekar number for a given base is a non-negative integer, the representation of whose square in that base can be split into two parts that add up to the original number again. For instance, 45 is a Kaprekar number, because $45^2 = 2025$ and $20 + 25 = 45$.

Algorithm:

- Initialize Scanner and Read Input.
- Create a Scanner object to read input from the user.
- Prompt the user to enter a number.
- Store the input number in an integer variable number.
- Calculate Square of the Number
- Compute the square of the number and store it in a long variable square.
- Determine the Number of Digits:
- Initialize a counter variable count to zero.
- Create a copy of number in variable n.
- Use a while loop to count the number of digits in number:
- Inside the loop, divide n by 10 and increment count by 1.
- Continue the loop until n becomes 0.
- Split the Square:
- Calculate the right part of the square by taking the remainder when square is divided by 10^{count} . Store this value in right.
- Calculate the left part of the square by dividing square by 10^{count} . Store this value in left.
- Check the Sum:
- If the sum of left and right equals the original number, print "true".
- Otherwise, print "false".

Implementation details:

```
import java.util.*;

public class Main
{
    public static void main(String[] args)
    {
        int number;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number: ");
```

```
number = sc.nextInt();

long square = number * number;

int count=0;

int n = number;

while(n!=0)

{

    n/=10;

    count++;

}

long right = square % (long)Math.pow(10,count);

long left = square / (long)Math.pow(10,count);

if(left + right == number)

{

    System.out.println("true");

}

else

{

    System.out.println("false");

}

}
```

Output:

```
Enter a number:
45
true
```

Conclusion:

Learnt about control statements

Date: _____

Signature of faculty in-charge

Post Lab Descriptive Questions:

Q.1 Write a program to find the largest of three numbers using the if-else construct.

```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        //to calculate max of 3 numbers
        int a,b,c;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter 3 Numbers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        if(a > b && a > c)
        {
            System.out.println("Largest is: "+ a);
        }
        else if(b > a && b > c)
```

```
{  
    System.out.println("Largest is: "+ b);  
}  
else if(c > a && c > b)  
{  
    System.out.println("Largest is: "+ c);  
}  
}  
}
```

Q.2 Write a program to determine the sum of the following series for a given value of n: $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$

```
import java.util.*;  
public class Main  
{  
    public static void main(String[] args)  
    {  
        int n;  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter a number: ");  
        n = sc.nextInt();  
        float ans = 0;  
        for(int i=1; i<=n; i++)  
        {  
            ans += (float)1/i;  
        }  
        System.out.println("Sum is: "+ ans);  
    }  
}
```

Output:

1)

```
Enter 3 Numbers:  
72 54 100  
Largest is: 100
```

2)

```
Enter a number:  
10  
Sum is: 2.9289684  
  
Process finished with exit code 0
```