**Course Outcome 1**

**Experiment 1 Date:**

**Basic Java Programs**

**Aim:**

Write the following programs

i) Print the prime numbers up to a limit

**Program**

import java.io.\*;

class PrimeInLimit

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("Enter Limit");

int n=Integer.parseInt(x.readLine());

System.out.println("Prime numbers up to "+n+":");

for(int num = 2; num <= n; num++)

{

int flag=0;

for (int i = 2; i<num/2; i++)

{

if (num % i == 0)

{

flag=1;

break;

}

}

if(flag==0)

{

System.out.println(num);

}

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java PrimeInLimit

Enter Limit

15

Prime numbers up to 15:

2

3

4

5

7

11

13

ii) Print the 3-digit Armstrong numbers between two intervals.

**Program**

import java.io.\*;

class ArmstrongInLimit

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("Enter Limit 1");

int num1=Integer.parseInt(x.readLine());

System.out.println("Enter Limit 2");

int num2=Integer.parseInt(x.readLine());

System.out.println("Armstrong Numbers:");

for (int i = num1; i<num2; i++)

{

int n=0;

int temp=i;

while (temp != 0)

{

temp=temp/10;

n=n+1;

}

int sum=0;

temp=i;

while (temp != 0)

{

int digit=temp%10;

sum=sum+(int)Math.pow(digit,n);

temp=temp/10;

}

if (sum==i)

{

System.out.println(i);

}

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java ArmstrongInLimit

Enter Limit 1:

100

Enter Limit 2:

500

Armstrong Numbers:

153

370

371

407

**Experiment 2 Date:**

**One-Dimensional Array**

**Aim:**

Write a Java program to search an element in an array

**Program**

import java.io.\*;

class ElementCheck

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("enter limit of array");

int n=Integer.parseInt(x.readLine());

int a[]=new int[n];

System.out.println("enter elements");

for(int i=0;i<n;i++)

{

a[i]=Integer.parseInt(x.readLine());

}

System.out.println("elements");

for(int i=0;i<n;i++)

{

System.out.print(a[i]+" ");

}

System.out.println();

int c=1,flag=0;

System.out.println("enter element to check");

int y=Integer.parseInt(x.readLine());

for(int i=0;i<n;i++)

{

if(a[i]==y)

{

flag=1;

break;

}

c=c+1;

}

if(flag==1)

{

System.out.println("element found at position "+c);

}

else

{

System.out.println("element not found");

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java ElementCheck

enter limit of array

4

enter elements

8

4

6

2

elements

8 4 6 2

enter element to check

6

element found at position 3

mits@mits-Veriton-M200-H510:~/gokul java$ java ElementCheck

enter limit of array

4

enter elements

8

4

6

2

elements

8 4 6 2

enter element to check

10

element not found

**Experiment 3 Date:**

**Two-Dimensional Array**

**Aim:**

Write a program to read a matrix from the console and check whether it is symmetric or not.

**Program**

import java.io.\*;

class SymmetricMatrix

{

public static void main(String args[]) throws IOException

{

int flag=0;

DataInputStream x=new DataInputStream(System.in);

System.out.println("Enter order of matrix");

int n=Integer.parseInt(x.readLine());

int a[][]=new int[n][n];

System.out.println("Enter elements of Matrix");

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

a[i][j]=Integer.parseInt(x.readLine());

}

}

System.out.println("Matrix elements");

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

System.out.print(a[i][j]+" ");

}

System.out.println();

}

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

if(a[i][j]!=a[j][i])

{

flag=1;

break;

}

}

}

if(flag==0)

{

System.out.println("Matrix is Symmetric");

}

else

{

System.out.println("Matrix is not Symmetric");

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java SymmetricMatrix

Enter order of matrix

3

Enter elements of Matrix

1

0

1

0

1

0

1

0

1

Matrix elements

1 0 1

0 1 0

1 0 1

Matrix is Symmetric

mits@mits-Veriton-M200-H510:~/gokul java$ java SymmetricMatrix

Enter order of matrix

3

Enter elements of Matrix

1

0

4

5

1

6

1

0

5

Matrix elements

1 0 4

5 1 6

1 0 5

Matrix is not Symmetric

**Experiment 4 Date:**

**String Handling Methods- 1**

**Aim:**

Perform the following operations on strings

i. Find the length of the string

ii. Character at second and fourth position

iii. Find the sub string using start index only

iv. Find the sub string using start index and end index

v. Compare two strings lexicographically.

vi. Compare two strings lexicographically, ignoring case differences.

vii. Concatenate a given string to the end of another string.

viii. Replace a specified character with another character.

ix. Check whether a given string starts with another string.

x. Convert all characters in a string to lowercase

xii. Convert all characters in a string to uppercase.

**Program**

import java.io.\*;

class StringOperations

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("1.Length of string");

System.out.println("Enter a string");

String s=x.readLine();

System.out.println("string is: "+s);

System.out.println("length of string is: "+s.length());

System.out.println();

System.out.println("2.Character At Position");

System.out.println("Character at second position: "+s.charAt(1));

System.out.println("Character at fourth position: "+s.charAt(3));

System.out.println();

System.out.println("3.Substring Using Start Index");

System.out.println("Enter start index");

int st=Integer.parseInt(x.readLine());

System.out.println("Substring from start index: "+s.substring(st));

System.out.println();

System.out.println("4.Substring Using Start and End Index");

System.out.println("Enter start index");

int st1=Integer.parseInt(x.readLine());

System.out.println("Enter end index");

int ed=Integer.parseInt(x.readLine());

System.out.println("Substring from start to end index: "+s.substring(st1,ed));

System.out.println();

System.out.println("5.Compare Strings");

System.out.println("Enter a new string1");

String s8=x.readLine();

System.out.println("Enter a new string2");

String s9=x.readLine();

if(s8.equals(s9))

{

System.out.println("String equal");

}

else

{

System.out.println("String not equal");

}

System.out.println();

System.out.println("6.Compare Strings(Ignore Case)");

System.out.println("Enter a new string1");

String s10=x.readLine();

System.out.println("Enter a new string2");

String s11=x.readLine();

if(s10.equalsIgnoreCase(s11))

{

System.out.println("String equal");

}

else

{

System.out.println("String not equal");

}

System.out.println();

System.out.println("7.Concatenate Strings");

System.out.println("Enter a new string1");

String s1=x.readLine();

System.out.println("Enter a new string2");

String s2=x.readLine();

System.out.println("After Concatenate: "+s1.concat(s2));

System.out.println();

System.out.println("8.Character Replace");

System.out.println("Enter a new string");

String s3=x.readLine();

System.out.println("Enter a character to replace");

char ch1=(x.readLine().charAt(0));

System.out.println("Enter new character");

char ch2=(x.readLine().charAt(0));

System.out.println("After Replace: "+s3.replace(ch1,ch2));

System.out.println();

System.out.println("9.Start With a String");

System.out.println("Enter a new string");

String s4=x.readLine();

System.out.println("Enter start string");

String s5=x.readLine();

if(s4.startsWith(s5))

{

System.out.println("String start with "+s5);

}

else

{

System.out.println("String not start with "+s5);

}

System.out.println();

System.out.println("10.Uppercase");

System.out.println("Enter a new string");

String s6=x.readLine();

System.out.println("Uppercase: "+s6.toUpperCase());

System.out.println();

System.out.println("11.Lowercase");

System.out.println("Enter a new string");

String s7=x.readLine();

System.out.println("Lowercase: "+s7.toLowerCase());

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java StringOperations

1.Length of string

Enter a string

gokulrajc

string is:

length of string is: 9

2.Character At Position

Character at second position: o

Character at fourth position: u

3.Substring Using Start Index

Enter start index

4

Substring from start index:

ulrajc

4.Substring Using Start and End Index

Enter start index

2

Enter end index

6

Substring from start to end index:

Okul

5.Compare Strings

Enter a new string1

abcd

Enter a new string2

ABCD

String not equal

6.Compare Strings(Ignore Case)

Enter a new string1

abcd

Enter a new string2

ABCD

String equal

7.Concatenate Strings

Enter a new string1

gokul

Enter a new string2

raj

After Concatenate:

gokulraj

8.Character Replace

Enter a new string

malayalam

Enter a character to replace

m

Enter new character

x

After Replace:

xalayalax

9.Start With a String

Enter a new string

hi welcome

Enter start string

hi

String start with hi

10.Uppercase

Enter a new string

abcd

Uppercase: ABCD

11.Lowercase

Enter a new string

ABCD

Lowercase: abcd

**Experiment 5 Date:**

**String Handling Methods- 2**

**Aim:**

Write a java program to

i. Check whether a given string is palindrome or not.

**Program**

import java.io.\*;

class StringPallindrome

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("Enter a string");

String s1=x.readLine();

String s2="";

System.out.println("String:"+s1);

int l = s1.length();

for(int i=l-1;i>=0;i--)

{

s2=s2+s1.charAt(i);

}

System.out.println("Reversed String:"+s2);

if(s1.equals(s2))

{

System.out.println("pallindrome");

}

else

{

System.out.println("not pallindrome");

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java StringPallindrome

Enter a string

malayalam

String:malayalam

Reversed String:malayalam

pallindrome

mits@mits-Veriton-M200-H510:~/gokul java$ java StringPallindrome

Enter a string

welcome

String:welcome

Reversed String:emoclew

not pallindrome

ii. Sorting a given list of names in ascending order

**Program**

import java.io.\*;

class NameSort

{

public static void main(String args[]) throws IOException

{

DataInputStream x=new DataInputStream(System.in);

System.out.println("Enter limit");

int n=Integer.parseInt(x.readLine());

String str[]=new String[n];

String temp;

System.out.println("Enter names");

for(int i=0;i<n;i++)

{

str[i]=x.readLine();

}

for (int i=0;i<n;i++)

{

for (int j=0;j<n;j++)

{

if(str[i].compareTo(str[j]) > 0)

{

temp=str[i];

str[i]=str[j];

str[j]=temp;

}

}

}

System.out.println();

System.out.println("Names");

for(int i=0;i<n;i++)

{

System.out.println(str[i]);

}

}

}

**Output**

mits@mits-Veriton-M200-H510:~/gokul java$ java NameSort

Enter limit

5

Enter names

thomas

abhijith

allen

gokul

adwaith

Names

abhijith

adwaith

allen

gokul

thomas

**Experiment 6 Date:**

**StringBuffer Class Methods**

**Aim:**

Write a program in java for string handling which performs the following

i. Check the capacity of the StringBuffer object.

ii. Reverse the content of this string and convert the resultant string in upper case

iii. Read another string and append it to the resultant string of above.

**Program**

import java.io.\*;

class StringBufferExample

{

public static void main(String args[]) throws IOException

{

DataInputStream d = new DataInputStream(System.in);

System.out.println("Enter a string:");

String str = (d.readLine());

StringBuffer s = new StringBuffer(str);

System.out.println("Capacity is "+s.capacity());

s.reverse();

String s2 = s.toString().toUpperCase();

StringBuffer ss = new StringBuffer(s2);

System.out.println("After resversing and converting to uppercase: "+ss);

System.out.println("Enter a string to append:");

String s1 = (d.readLine());

System.out.println("New String: "+ss.append(s1));

}

}

**Output**

Enter a string:

gokul

Capacity is 21

After resversing and converting to uppercase: LUKOG

Enter a string to append:

raj

New String: LUKOGraj