1. **Program: Find out duplicate number between 1 to N numbers.**

**Description: You have got a range of numbers between 1 to N, where one of the number is repeated. You need to write a program to find out the duplicate number.**

**Ans.**

import java.util.List;

/\*\*

\*

\* @author Waheguru

\*/

public class Assignment1 {

public static void main(String[] args){

int[] array=new int[10];

System.out.println("The System generated array is : ");

for(int i=0;i<array.length;i++){

array[i]=(int)(Math.random()\*100);

System.out.print(array[i]);

System.out.print(" ");

}

int[] array1;

array1 = new int[array.length];

int count=0;

for(int j=0;j<array.length;j++)

{

for(int k=j+1;k<array.length;k++)

{

if(array[j]==array[k])

{

array1[count]=array[j];

count++;

}

}

}

System.out.println("");

System.out.println("");

if(count!=0)

{

System.out.println("Duplicate values for the system generated array are :");

for(int l=0;l<array1.length;l++)

{

System.out.print(array1[l]);

System.out.print(" ");

}

System.out.println("");

System.out.println("GoodBye!!!!!");

}

else

{

System.out.println("There are no duplicate values for the sytem generated aarrays ");

System.out.println("GoodBye!!!!!!");

}

}

}

**Output:**

The System generated array is :

94 24 42 20 82 66 97 62 73 89

There are no duplicate values for the sytem generated aarrays

GoodBye!!!!!!

1. **Program: Write a program to reverse a number.**

**Description: Write a program to reverse a number using numeric operations. Below example shows how to reverse a number using numeric operations.**

**Ans.**

import java.util.Scanner;

public class Assignment2 {

public static void main(String args[]){

int n, reverse = 0;

System.out.println("Enter the number to reverse");

Scanner in = new Scanner(System.in);

n = in.nextInt();

while( n != 0 )

{

reverse = reverse \* 10;

reverse = reverse + n%10;

n = n/10;

}

System.out.println("Reverse of entered number is "+reverse);

}

}

**Output:**

Enter the number to reverse

321764

Reverse of entered number is 467123

1. **Program: Write a program to find perfect number or not.**

**Description: A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors.**

**The first perfect number is 6, because 1, 2 and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors:**

**Ans.**

import java.util.Scanner;

public class Assignment3 {

public boolean numCheck(){

int num;

Scanner s=new Scanner(System.in);

System.out.println("Enter the number : ");

num=s.nextInt();

int temp = 0;

for(int i=1;i<=num/2;i++){

if(num%i == 0){

temp += i;

}

}

if(temp == num){

System.out.println("It is a perfect number");

return true;

} else {

System.out.println("It is not a perfect number");

return false;

}

}

public static void main(String a[]){

Assignment3 as = new Assignment3();

as.numCheck();

}

}

**Output:**

Enter the number :

8

It is not a perfect number

Enter the number :

6

It is a perfect number

1. **Program: Write a program to implement ArrayList.**

**Description: Write a program to implement your own ArrayList class. It should contain add(), get(), remove(), size() methods. Use dynamic array logic. It should increase its size when it reaches threshold.**

**Ans.**

import java.util.Arrays;

public class Assignment4 {

private Object[] obj;

private int size = 0;

public Assignment4(){

obj = new Object[10];

}

public Object get(int index){

if(index < size){

return obj[index];

} else {

throw new ArrayIndexOutOfBoundsException();

}

}

public void add(Object obj1){

if(obj.length-size <= 5){

increaseListSize();

}

obj[size++] = obj1;

}

public Object remove(int index){

if(index < size){

Object obj1 = obj[index];

obj[index] = null;

int temp = index;

while(temp < size){

obj[temp] = obj[temp+1];

obj[temp+1] = null;

temp++;

}

size--;

return obj1;

} else {

throw new ArrayIndexOutOfBoundsException();

}

}

public int size(){

return size;

}

private void increaseListSize(){

obj = Arrays.copyOf(obj, obj.length\*2);

System.out.println("\nNew length: "+obj.length);

}

public static void main(String a[]){

Assignment4 as = new Assignment4();

as.add(new Integer(12));

as.add(new Integer(7));

as.add(new Integer(70));

as.add(new Integer(4));

as.add(new Integer(99));

for(int i=0;i<as.size();i++){

System.out.print(as.get(i)+" ");

}

as.add(new Integer(29));

System.out.println("Element at Index 5:"+as.get(5));

System.out.println("List size: "+as.size());

System.out.println("Removing element at index 2: "+as.remove(2));

for(int i=0;i<as.size();i++){

System.out.print(as.get(i)+" ");

}

}

}

**Output**:

12 7 70 4 99

New length: 20

Element at Index 5:29

List size: 6

Removing element at index 2: 70

12 7 4 99 29

1. **Program to find second highest number in an integer array without sorting the elements.**

**Ans.**

public class Assignment5 {

public static void main(String args[]){

int numbers[] = {10,2,76,89,44,23,78,100,32};

int highest = 0;

int second\_highest = 0;

for(int n:numbers){

if(highest < n){

second\_highest = highest;

highest =n;

}

else if(second\_highest < n){

second\_highest = n;

}

}

System.out.println("Second Highest number is : "+second\_highest);

}

}

**Output:**

Second Highest number is : 89

**6. Java Program for Fibonacci series without using recursive function concept.**

**Ans.**

import java.util.Scanner;

public class Assignment6 {

public static void main(String args[ ]) {

Scanner input=new Scanner(System.in);

int i,a=1,b=1,c=0,t;

System.out.println("Enter value of t:");

t=input.nextInt();

System.out.print(a);

System.out.print(" "+b);

for(i=0;i<t-2;i++) {

c=a+b;

a=b;

b=c;

System.out.print(" "+c);

}

System.out.println();

System.out.print(t+"th value of the series is: "+c);

}

}

**Output:**

Enter value of t:

10

1 1 2 3 5 8 13 21 34 55

10th value of the series is: 5

1. **Program: Write a program to find sum of each digit in the given**

**number using recursion.**

**Description: Below example shows how to find out sum of each digit in the given number using recursion logic. For example, if the number is 259, then the sum should be 2+5+9 = 16**

**Ans.**

import java.util.Scanner;

public class Assignment7 {

int sum = 0;

public int addNum(int n){

if(n!=0)

{

sum+=n%10;

n=n/10;

addNum(n);

}

else

{

return sum;

}

return sum

}

public static void main(String a[]){

Assignment7 as = new Assignment7();

System.out.println("Enter the number");

Scanner s=new Scanner(System.in);

int n=s.nextInt();

as.addNum(n);

System.out.println("Sum is: "+as.sum);

}

}

**Output:**

Enter the number

598

Sum is: 22

**8. Program: Write a program to find the sum of the first 1000 prime numbers.**

**Description: Write a program to find the sum of the first 1000 prime numbers.**

**Ans.**

public class Assignment8 {

private static boolean checkPrime(int number){

for(int i=2; i<=number/2; i++){

if(number % i == 0){

return false;

}

}

return true;

}

public static void main(String args[]){

int n = 2;

int count = 0;

//int i[]=null;

long sum = 0;

while(count < 1000){

if(checkPrime(n)){

sum += n;

count++;

}

n++;

}

System.out.println(“The Sum Of first 1000 prime number is :”+sum);

}

}

**Output:**

The Sum Of first 1000 prime number is :3682913

**8b. Program: How to swap two numbers without using temporary variable?**

**Description: Write a program to swap or exchange two numbers. You should not use any temporary or third variable to swap.**

**Ans.**

import java.util.Scanner;

public class Assignment8b {

//int x,y;

public static void main(String args[]){

Scanner s=new Scanner(System.in);

System.out.println("Enter the value for x and y ");

int x=s.nextInt();

int y=s.nextInt();

System.out.println("Before Swapping");

System.out.println("x : "+x);

System.out.println("y : "+y);

x = x + y;

y = x - y;

x = x - y;

System.out.println("After Swapping");

System.out.println("x : "+x);

System.out.println("y : "+y);

}

}

**Output:**

Enter the value for x and y

2

5

Before Swapping

x : 2

y : 5

After Swapping

x : 5

y : 2

1. **Write a Program to find top 2 maximum numbers in array.**

**You should not use any sorting functions. You should iterate the array only once. You should not use any kind of collections in java.**

**Ans.**

public class Assignment9 {

public void firstTwoMaxNumbers(int[] nums){

int maxOne = 0;

int maxTwo = 0;

for(int n:nums){

if(maxOne < n){

maxTwo = maxOne;

maxOne =n;

} else if(maxTwo < n){

maxTwo = n;

}

}

System.out.println("First Max Number: "+maxOne);

System.out.println("Second Max Number: "+maxTwo);

}

public static void main(String a[]){

int num[] = {8,3,90,56,23,67,1,7,97,33,28};

Assignment9 as = new Assignment9();

as.firstTwoMaxNumbers(num);

}

}

**Output:**

First Max Number: 97

Second Max Number: 90

1. **Program: Floyd Triangle**

**1**

**2 3**

**4 5 6   
7 8 9 10**

**Ans.**

public class Assignment10 {

public static void main(String args[]){

int num = 1;

System.out.println("Floyd's triangle :-");

for (int i = 1 ; i <= 4 ; i++ )

{

for (int j = 1 ; j <= i ; j++ )

{

System.out.print(num+" ");

num++;

}

System.out.println();

}

}

}

**Output:**

Floyd's triangle :-

1

2 3

4 5 6

7 8 9 10

1. How to count occurrence of a given character in a String?

Ans.

import java.util.Scanner;

public class Assignment11 {

public static void main(String[] args) {

String str ;

Scanner in= new Scanner(System.in);

System.out.println("Please enter a String");

str=in.nextLine();

System.out.println("Please enter a Character");

String chr=in.next();

int charCount = str.length() - str.replaceAll(chr, "").length();

System.out.println("Number of occurances of given character:"+charCount);

}

}

**Output:**

Please enter a String

DJkljdlkJAKLadhajKDHJKALK

Please enter a Character

J

Number of occurances of given character:3

1. **Program: Write a program to check whether accepted number is Magic number.**

**Description:**

**A Magic number is a number whose sum of digits eventually leads to 1. Example#1: 19 ; 1+9 =10 ; 1+0 = 1. Hence a magic number. Example#2: 226; 2+2+6=10; 1+0 =1. Hence a magic number. Example#3: 874; 8+7+4=19; 1+9=10; 1+0=1. Hence a magic number.**

**Ans.**

import java.util.Scanner;

public class Assignment12 {

public static void main(String args[])

{

Scanner ob=new Scanner(System.in);

System.out.println("Enter the number to be checked.");

int n=ob.nextInt();

int sum=0,num=n;

while(num>9)

{

sum=num;int s=0;

while(sum!=0)

{

s=s+(sum%10);

sum=sum/10;

}

num=s;

}

if(num==1)

{

System.out.println(n+" is a Magic Number.");

}

else

{

System.out.println(n+" is not a Magic Number.");

}

}

}

**Output:**

Enter the number to be checked.

1927

1927 is a Magic Number.

1. **Program: Write a program How to check if a number is binary.**

**Description:**

**For this question, you need to write a function which will accept an integer and return true if it contains only 0 and 1 e.g. if input is 123 then your function will return false, for 101 it should return true.**

**Ans.**

import java.util.Scanner;

public class Assignment13 {

public boolean isBinaryNumber(int num){

boolean status = true;

while(true){

if(num == 0 || num==1){

System.out.println("It is a binary number");

break;

} else {

int tmp = num%10;

if(tmp > 1){

status = false;

System.out.println("The number is not binary");

break;

}

num = num/10;

}

}

return status;

}

public static void main(String a[]){

Scanner s=new Scanner(System.in);

int x=s.nextInt();

Assignment13 as= new Assignment13();

as.isBinaryNumber(x);

}

}

**Output:**

1235

The number is not binary

10101

It is a binary number

1. **Program: Write a program to generate 10 random numbers using Random class within the range of 1 to 100 using for loop.**

**Description:**

**By using Math.random() method also we can generate random number in java.**

**Ans.**

import java.util.Random;

import java.util.Scanner;

public class Assignment14 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("For range enter minimum number");

int minimum=in.nextInt();

System.out.println("For range enter maximum number");

int maximum=in.nextInt();

Random r = new Random();

int range = maximum - minimum + 1;

System.out.println("10 Random Numbers are ");

for(int i=0;i<10;i++){

int randomNum = r.nextInt(range) + minimum;

System.out.println(randomNum);

}

}

}

**Output:**

For range enter minimum number

1

For range enter maximum number

100

10 Random Numbers are

94

80

88

31

82

81

21

88

59

90

1. **Program: Write a program to use super() method.**

**Description:**

**Basically super keyword used to refer super class methods and variables.**

**Ans.**

public class Assignment15 {

int x=10;

int y=20;

}

class Child extends Assignment15{

int x=100;

int y=200;

void display()

{

System.out.println("Before using super");

System.out.println("Value of x :"+x);

System.out.println("Value of y :"+y);

System.out.println("After using super");

System.out.println("Value of x :"+super.x);

System.out.println("Value of y :"+super.y);

}

public static void main(String args[]){

Child c=new Child();

c.display();

}

}

**Output:**

Before using super

Value of x :100

Value of y :200

After using super

Value of x :10

Value of y :20

1. **Program: Write a program to reverse vowels of a given string.**

**Description:**

**Let's string is InstanceOfJava**

**After reversing vowels in a string=anstancOefjavI**

**Ans.**

import java.util.Scanner;

public class Assignment16 {

public static String reverseVowels(String string) {

String vowelsStr = "aeiouAEIOU";

int low = 0;

int high = string.length() - 1;

char[] ch = string.toCharArray();

while (low < high) {

if (!vowelsStr.contains(String.valueOf(string.charAt(low)))) {

low++;

continue;

}

if (!vowelsStr.contains(String.valueOf(string.charAt(high)))) {

high--;

continue;

}

swap(ch, low, high);

low++;

high--;

}

return String.valueOf(ch);

}

private static void swap(char[] ch, int low, int high) {

char temparray = ch[low];

ch[low] = ch[high];

ch[high] = temparray;

}

public static void main (String args[]) {

String str;

Scanner s=new Scanner(System.in);

str=s.next();

System.out.println(str.length());

System.out.println("After reversing vowels in a string");

System.out.println(reverseVowels(str));

//need to try for string array..

}

}

**Output:**

InstanceofJava

14

After reversing vowels in a string

anstancoefJavI

**17. WAP to implement Bubble sort in java.  
Ans.**

public class BubbleSort {

public static void bubSort(int[] x) {

int n = x.length;

int temp = 0;

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

for(int j=1; j < (n-i); j++){

if(x[j-1] > x[j]){

temp = x[j-1];

x[j-1] = x[j];

x[j] = temp;

}

}

}

}

public static void main(String[] args) {

int x[] ={10,8,99,87,58,76,3,95,110,25,75};

System.out.println("Array Before Bubble Sort");

for(int i=0; i < x.length; i++){

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

}

System.out.println();

bubSort(x);

System.out.println("Array After Bubble Sort");

for(int i=0; i < x.length; i++){

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

}

}

}

**Output:**

Array Before Bubble Sort

10 8 99 87 58 76 3 95 110 25 75

Array After Bubble Sort

3 8 10 25 58 75 76 87 95 99 110

**18. WAP to print the following pattern (example3)**

**\***

**\* \* \***

**\* \* \* \* \***

**\* \* \* \* \* \* \***

**\* \* \* \* \* \* \* \* \***

**Ans.**

public class Assignment18 {

public static void main(String[] args) {

// TODO code application logic here

for(int i=1; i<10; i+=2)

{

for(int j=1; j<=i; j++)

{

System.out.print("\*");

}

System.out.println();

}

}

}

**Output:**

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

**19. WAP to get factorial of a number without using recursion.**

**Ans.**

import java.util.Scanner;

public class Assignment19 {

public static int factorial(int num) {

int result = 1;

for (int i = 1; i <= num; i++) {

result = result \* i;

}

return result;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number whose factorial is to be found: ");

int num = scanner.nextInt();

int result = factorial(num);

System.out.println("The factorial of " + num + " is " + result);

}

}

**Output:**

Enter the number whose factorial is to be found: 8

The factorial of 8 is 40320

**20. WAP to take a sentence from user and do unique word count using split().**

**Ans.**

import java.io.\*;

public class RemoveDuplicatesWords

{

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

{

BufferedReader br = new BufferedReader(

new InputStreamReader(System.in));

System.out.println("Enter the sentence");

String input = br.readLine();

countWords(input);

}

// static void a(Int id)

static void countWords(String st)

{

// split text to array of words

String[] words = st.split("\\s");

// clean duplicates

for (int i = 0; i < words.length; i++)

{

for (int j = 0; j < words.length; j++)

{

if (words[i].equals(words[j]))

{

if (i != j){

words[i] = "";

}

}

}

}

// show the output

System.out.println("");

System.out.println("Removing Duplicate words:\n");

int count=0;

for (int i = 0; i < words.length; i++)

{

if (!"".equals(words[i]))

{

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

count++;

}

}

System.out.println("");

System.out.println("Total number of Unique Words in the sentence is "+count);

}

}

**Output:**

Enter the sentence

this is is a a prog to check it is a

Removing Duplicate words:

this prog to check it is a

**Total number of Unique Words in the sentence is 7**