ASSIGNMENT NUMBER - 1

TOPIC - Decimal to Binary Conversion.

QUESTION

Java Program on decimal to Binary Conversion

PROGRAMMING

```
import java.util.*;
public class decimaltobinary
{
public static void main ()
{
int d, a; " ";
String x = " ";
Scanner sc = new Scanner (System.in);
System.out.println("Enter any decimal number:");
d = sc.nextInt(); // input taken by user
while (d > 0) // while loop
 {
 a = d%2; // stores the remainder either 1 or 0
 x = a+ " "+x; // binary conversion
 d = d/2; // divides the number by user
 }
 System.out.println("Binary number:"+x);
    }
 }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
d	int	input variable - stores the decimal number.
a	int	stores the remainder either 1 or 0.
x	string	output variable - Show the binary number.

OUTPUT

Enter any decimal number:
14
Binary Number: 1110

Enter any decimal number:
47
Binary Number: 101111

ASSIGNMENT NUMBER - 2

TOPIC - check unique number or not.

QUESTION

Java program to check for unique number.

PROGRAMMING

```
import java.util.*;
public class UniqueNumber {
public static void main()
{ int d,a,flag=0,i,j;
string x="";
Scanner sc=new Scanner(System.in);
System.out.println("Enter the number:");
d=sc.nextInt(); // input from user
x=Integer.toString(d); // converts int data type to string
a=x.length(); // determines length of string.
for(i=0; i<a-1; i++){ // checks the digits are repeated
for(j=i+1; j<a; j++){
if(x.charAt(i)==x.charAt(j)){
flag=1; // comparing each digits, repeated or not
break; }}}
if(flag==0) // if flag is 0, the number is unique.
System.out.println("The number is unique");
else
System.out.println("The number is not unique"); }}
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
d	int	input variable - number to check.
a	int	to store the length of string X.
flag	int	checks uniaue number or not.
i	int	for loop variable
j	int	for loop variable
X	string	convert input d into string type & stores.

OUTPUT

Enter the number:
478
The number is uniaue.

Enter the number:
1010

The number is not uniaue.

ASSIGNMENT NUMBER – 3

TOPIC – Java programming on Fibonacci series using recursion.

QUESTION

write a program in Java to print the Fibonacci series using recursion.

PROGRAMMING

```
import java.util.*;
public class Fibonacci {
public static void main() throws IO Exception
{ int n, i;
Scanner sc = new Scanner (System.in);
System.out.println("Enter the number to limit:");
n = sc.nextInt(); // input taken from user.
for (i=1; i<=n; i++) // loop till the last count
{
System.out.print (printFibonacci(i)+",");}
inpict.close(); } // recursion function.
private static int printFibonacci (int c) {
if (c==1 || c==2) { // exit condition
return 1; }
return printFibonacci (c-1)+printFibonacci (c-2);
} }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	input variable – the number to limit.
i	int	for loop variable.
c	int	parameterized input variable for recursion function.

OUTPUT

Enter the number to limit :
10
1, 1, 2, 3, 5, 8, 13, 21, 34, 55

Enter the number to limit :
15
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377,
610

ASSIGNMENT NUMBER - 4

TOPIC - Java program to find difference between two dates.

QUESTION

Write a program in Java to find difference between two dates.

PROGRAMMING

ASSIGNMENT NUMBER – 5

TOPIC - Every Alphabet frequency in a string.

QUESTION

Java program to find freauency of every alphabet in a string.

PROGRAMMING

```
import java.util.*;
public class frequency {
public static void main()
{ Scanner sc=new Scanner(System.in);
System.out.println("Enter the string:");
s=sc.nextLine();
s=s.trim();
int f[]=new int[s.length()];
int i, j;
// converts given string into character array.
char string[]=s.toCharArray();
for(i=0; i<s.length(); i++){
f[i]=1;
for(j=i+1; j<s.length(); j++){ // for loop
if(string[i]==string[j]){
f[i]++;
// Set string[j] to 0 to avoid printing visited char
string[j]='0'; ???
```

Characters & their frequencies :

p = 2
i = 1
c = 2
t = 2
∴ u = 1
r = 2
e = 3
f = 1

ASSIGNMENT NUMBER - 6

TOPIC - Pendulum Arrangement

QUESTION

Java program to find the pendulum arrangement of list of integers in an array.

PROGRAMMING

```
import java.util.*;
public class PendulumArrangement {
public static void main() {
Scanner sc = new Scanner (System.in);
System.out.println("Enter number of elements :");
int n = sc.nextInt();
int a[] = new int[n];
System.out.println("Enter the elements :");
for(i=0; i<n; i++) {
a[i] = sc.nextInt(); }
System.out.println("Pendulum Arrangement:");
  //Step 1 : Sort the array
for(int i=0; i<n-1; i++) {
for(int j=0; j<n-i-1; j++) {
 if (a[j] > a[j+1]) {
   int t = a[j];
   a[j] = a[j+1];
   a[j+1] = t; } }
```

```
//Step 2: Get elements at odd indexes to the right
int e=n-1;
int s;
if(n%2==0){
s=n-1; }
else{
 s=n-2; }
   while(s>0){
       int t=a[s];
       int idx=s;
   while(idx!=e){
     a[idx]=a[idx+1]
     idx++; }
a[idx]=t;
 s-=2;
 e-=1; }

//Step 3: Reverse the sub array from 0 to (n-1)/2
for(int i=0; j=(n-1)/2; i<j; i++; j--){
     int t=a[i];
     a[i]=a[j];
     a[j]=t; }

//Print the final array
   for(int i=0; i<n; i++){
System.out.print (a[i]+" "); } } }
```


VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	input number of elements in the array.
a	int	input the elements of the array.
i, j	int	for loop variables.
t	int	temporarily stores the integer.
e	int	End Index integer stores.
s	int	Starting Index integer stores.
idx	int	Mentioned Index Integer.

OUTPUT

```
Enter the number of elements:
5
Enter the elements:
11
12
```

31
14
5
Pendulum Arrangement:
31 12 5 11 14

Enter the number of elements:
5
Enter the elements
1
2
3
4
5
Pendulum Arrangement:
5 3 1 2 4

ASSIGNMENT NUMBER - 7

TOPIC - Reverse sentences without punctuations.

QUESTION

Java program to input multiple sentences and reverse it without punctuation mark.

PROGRAMMING

```
import java.io.*;
import java.util.*;
public class reversesentence
{ public static void main ()
{ BufferedReader br = new BufferedReader (new
  InputStreamReader (System.in));
System.out.print ("Enter the number of sentences:");
int n = Integer.parseInt (br.readline ());
// inputting the number of sentences to accept.
String s= "";
for (int i= 1; i<=n; i++)
{
System.out.print ("Enter Sentence "+i+":");
s= s+br.readline (); }
//inputting multiple sentences and joining them
into the same string */
```

```
String Tokenizer str = new StringTokenizer(s,".,;:!?");
//converting sentence into StringTokenizer.

int c = str.countTokens();
String w = "" ; rev = "";
for (int i=1; i<=c; i++) {
w = str.nextToken();
// extracting one word at a time
rev = w + " " + rev; //joining words in reverse order
}
System.out.println("Output:" + rev); }}
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	input variable - stores the number of sentences.
s	String	input variable - stores the sentences.
i	int	for loop variable.
str	String	converts & stores the sentences into StringTokenizer.
c	int	count the total number of words.

w	String	extracts & stores one word at a time.
rev	String	output - reverse the multiple sentences.

OUTPUT

```
Enter the number of Sentences:
2
Enter Sentence 1:
Emotions, controlled and directed to work, is
character.
Enter Sentence 2:
By Swami Vivekananda.

Output: Vivekananda Swami By character is work
to directed and controlled, Emotions.
```

ASSIGNMENT NUMBER – 8

TOPIC – Remove Consecutive Repeated characters.

QUESTION

Write a program in Java to remove the consecutive repeated characters from a word & print it.

PROGRAMMING

```
import java.util.*;
public class repeatedcharacters
{
 public static void main()
{ Scanner sc=new Scanner(System.in);
  System.out.println("Enter any word:");
 String s=sc.nextLine();
 s=s+" ";
int l=s.length();//stores length of the word
String rmv="";
char ch1,ch2;
for(int i=0;i<l-1;i++){ //for loop
 ch1=s.charAt(i);//stores the character at i.
 ch2=s.charAt(i+1);//stores character at i+1
if(ch1!=ch2){//Checkes whether ch1 & ch2 are same.
rmv=rmv+ch1;}}//output
System.out.println("Output:"+rmv);}}
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
s	string	input variable - stores the word.
l	int	stores the length of the string s.
i	int	for loop variable.
ch1, ch2	string	stores the character at i & (i+1) respectively.
rmv	string	output variable - removes the repeated characters.

OUTPUT

Enter any word:
Hello
Output : Helo

Enter any word:
commputter
Output : Computer

ASSIGNMENT NUMBER – 9

TOPIC – Circular (Spiral Matrix)

QUESTION

Write a program in java to print Circular (Spiral) Matrix.

PROGRAMMING

```
import java.util.*;
public class circularmatrix {
public static void main()
{ Scanner sc=new Scanner(System.in);
 System.out.println("Enter no of elements:");
int n= sc.nextInt();
int A[][]=new int[n][n];
int k=1,c1=0,c2=n-1,r1=0,r2=n-1;
while(k<=n*n){
for(int i=c1;i<=c2;i++){
A[r1][i]=k++; }
//Print the first row from remaining rows
for(int j=r1+1;j<=r2;j++){
A[j][c2]=k++; }
//Print the last column from remaining columns
for(int j=c2-1;j>=c1;j--){
A[r2][i]=k++; }
//Print the last row from remaining rows.
```

```
for (int j=r2-1; j>=r1+1; j--) {
A[i][c1]=k++; }
// Print the first column from remaining columns.
c1++;
c2--;
r1++;
r2--; }
System.out.println("The Circular Matrix is :");
for (int i=0; i<n; i++) {
 for (int j=0; j<n; j++) {
System.out.print(A[i][j]+"\t"); }
// Printing the Circular Matrix.
System.out.println(); } } }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	no of row and column Input variable
k	int	counter variable.
i	int	for loop variable
j	int	for loop variable

c1	int	First column Index.
c2	int	Last column Index
r1	int	First Row Index
r2	int	Last Row Index

ASSIGNMENT NUMBER - 10

TOPIC - Student Report

QUESTION

A super class Record has been defined to store the names and ranks of 50 students. Define a sub class Rank to find the highest rank along with the name. The details of both classes are given below:

Class name : Record

Data Members / Instance Variables:
name[] : to store the names of students
rnk[] : to store the ranks of students.

Member functions:
Record() : constructor to initialize data members.
void readvalues() : to store names and ranks.
void display() : displays the names and corresponding ranks.

Class Name : Rank.

Data Members / Instance Variables:
index : Integer to store the index of topmost rank.

Member functions:

Rank() : Constructor to invoke the base class constructor and initialize index to 0.

void highest() : finds the index location of topmost rank and stores it in index without sorting the array.

void display() : displays the name and ranks along with the name having the topmost rank.

Specify the class Record giving details of the constructor (), void readvalues(), void display(). Using the concept of inheritance, specify the class Rank giving details of constructor (), void highest() and void display().

write the main function.

PROGRAMMING

```
import java.util.*;
public class Record {
Scanner sc = new Scanner (System.in);
String name [] = new String [50];
int rnk [] = new int [50];
public Record () {
for(int i=0; i<50; i++) {
name [i] = "";
rnk [i] = 0; } }
public void readvalue () {
int i = 0;
System.out.println ("Enter names & ranks");
for (i=0; i<50; i++) {
name [i] = sc.nextLine (); }
rnk [i] = sc.nextInt (); }
// to store the names and ranks of the students
public void display () {
int i = 0;
for (i=0; i<50; i++) {
System.out.println (name [i] + " " + rnk [i]); }}}
// to display the names & ranks correspondingly.

import java.util.*;
public class Rank {
Scanner sc = new Scanner (System.in);
int index;
public Rank () {
super ();
index = 0; }
// constructor to initialize the data members.
public void highest () {
int i = 0;
for (i=0; i<50; i++) {
```

```
if (rank[index] > rank[i]) {
index = i;    } } }
//to calculate the highest rank holder name-index.
public void display () {
readvalue ();
highest ();
super.display ();
System.out.println ("Name of highest rank :" + name[index]);
} } //to display the name of highest rankholder
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
name []	String	to store the names of 50 students.
rank []	int	to store the ranks according to the name
index	int	to store the index of the highest rank holder.
i	int	For loop variable.

OUTPUT

ASSIGNMENT NUMBER - 11

TOPIC - Merge Arrays.

QUESTION

Java program for merging arrays inside object.

PROGRAMMING

```
import java.util.*;
public class MergeArrays
{
public static void main()
{
int i,b,j,m,n,a[]=new int[];
Scanner sc=new Scanner(System.in);
System.out.println("Enter no of arrays:");
n=sc.nextInt(); // Total no of arrays to be merged
for(i=1;i<=n;i++)
{
System.out.println("Enter Array "+i+" size:");
m=sc.nextInt(); // size of the arrays
System.out.println("Enter Array "+i+" elements:");
for(j=1;j<=m;j++)
{
a[j]=sc.nextInt(); // elements of the arrays
}
b=b+"\t"+a[j]; // merging all the arrays.
```

```
}
System.out.println("Merged Array :\n" +b); //display
}
}
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
a[]	int	to store the elements of arrays
i	int	For loop variable.
b	int	Output variable - merged arrays.
j	int	For loop variable.
m	int	to store the size of arrays
n	int	to store the number of arrays.

OUTPUT

```
Enter No of Arrays:
2
```



```
Enter Array 1 size:
3
Enter Aray 1 elements:
14
26
39
Enter Array 2 size:
5
Enter Array 2 elements:
72
96
54
32
27
Merged Array:
14 26 39 72 96 54 32 27
```

ASSIGNMENT NUMBER – 12

TOPIC – Print Twin Prime Numbers.

QUESTION

Write a program in Java to print Twin prime numbers within a range.

PROGRAMMING

```
import java.util.*;
public class TwinPrimeNo
{
public static void printTwinPrime (int n)
{
  // create a boolean array "prime [0...n]"
  // and initialize all entries in it as
  // true. A value in prime [i] will
  // finally be false if i is Not a
  // prime, else true.
boolean prime [] = new boolean [n+1];
for (int i=0; i<=n; i++)
  prime [i] = true;
for (int p=2; p*p<=n; p++) {

    // If prime [p] is not changed, then it is a prime.
if (prime [p] == true) {
    // update all multiples of p.
for (int i=p*2; i<=n; i+=p)
```

```
prime[i] = false ; } }
    // to check for twin prime numbers.
    // display the twin prime.
for(int i=2 ; i<=n-2; i++) {
if(prime[i]== true && prime[i+2]==true)
    //display the result.
System.out.print("("+i+","+(i+2)+")");
} }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	to enter the range of twin prime numbers.
i	int	for loop variable.
p	int	to find the prime numbers

OUTPUT

20
(3,5)(5,7)(11,13)(17,19).

15
(3,5)(5,7)(11,13)

ASSIGNMENT NUMBER – 13

TOPIC – Checking IMEI Number.

QUESTION

Write a program in Java to check for valid IMEI number.

PROGRAMMING

```
import java.util.*;
public class IMEI
{
public static void main ( ) {
Scanner sc = new Scanner (System.in);
System.out.println("Please enter 15 digit IMEI no:");
String input = sc.nextLine ( );
int c = getCheckDigit (input.substring (0,14));
int d = Integer.valueOf(input.substring (14));
    // c = computed check digit.
    // d = check digit in source.
if (c == d) {
System.out.println("It is valid IMEI number.");
} else {
System.out.println("It is not a valid IMEI number.");
System.out.println("Check digit computed:" + c);
} s.close ( ); }
//Returns check digit for 14 digit IMEI prefix.
```

```
public static int getCheckDigit (string imeiPrefix){
 int sum=0;
 for (int i=13; i>0; i=i-1) {
 String sDigit= imeiPrefix.substring (i, i+1);
 int digit = Integer.valueOf (sDigit);
 if (i%2==0) {
     sum = sum+digit;
 } else {
     sum = sum + sumOfDigits (digit*2);
     } }
 Sum = sum *9;
 return sum % 10;//Return check Digit.
}
//Calculate sum of digits for a number.
public static int sumOfDigits (int number){
int sum=0;
while (number >0) {
     sum+= number %10;
     number = number/10; }
return sum; } }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
input	string	to enter input

c	int	15 digit IMEI number. to store the computed check digit.
d	int	to store the check digit in source.
sum	int	to store the sum of first 14 digits of IMEI
i	int	for loop variable.
sDigit	string	to store the digits of IMEI number
digit	int	convert from string type to int type.

OUTPUT

```
Please enter 15 digit IMEI no:
914859533683732
It is not a valid IMEI number.
Check digit computed: 0

Please enter 15 digit IMEI no:
420154203437618
It is valid IMEI number.
```

ASSIGNMENT NUMBER – 14

TOPIC – Magic Square Matrix.

QUESTION

Write a program in Java to print the magic square matrix.

PROGRAMMING

```
import java.util.*;
public class magicsquare {
// Function to generate odd sized magic squares.
public static void generatesquare () {
Scanner sc = new Scanner (System.in);
System.out.println ("Enter the size of matrix:");
int n = sc.nextInt ();
int [][] magicSquare = new int [n][n];
// Initialize position for 1.
int i = n/2;
int j = n-1;
// One by one put all values in magic square.
for (int num=1; num<=n*n;) {
  if (i==-1 && j==n) // 3rd Condition.
  { j=n-2;
    i=0; }
else {
  if (j==n)
```

```
        j=0;
    if(i<0)
    i=n-1;
    }
    //2nd Condition.
    if(magicSquare[i][j]!=0){
    j-=2;
    i++;
    continue;}
    else                    //Set number
      magicSquare[i][j]=num++;
    j++;        // 1st Condition.
    i--;}
    System.out.println("the magic square for:"+n+":");
    System.out.println("Sum of each row or column"+
    n*(n*n+1)/2+":");
    for(i=0;i<n;i++){
      for(j=0;j<n;j++)
        System.out.print(magicSquare[i][j]+" ");
    System.out.println();}} //Display Output
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	to store the size of

magicSquare	int	magic square matrix. to store the elements of matrix.
i	int	for loop variable — index of the matrix.
j	int	for loop variable — index of matrix.
num	int	for loop variable — total no of elements.

OUTPUT

```
Enter the size of matrix:
5
The magic square for 5:
Sum of each row or column 65:
9  3  22  16  15
2  21  20  4  8
25  19  13  7  1
18  12  6  5  24
11  10  4  23  17
```

ASSIGNMENT NUMBER- 15

TOPIC- Keith Number.

QUESTION

Write a program in Java to input a number and check whether it is Keith number or not.

PROGRAMMING

```
import java.util.*;
public class keithnumber {
public static void main () {
Scanner sc = new Scanner (System.in);
System.out.println ("Enter a number:");
int n = sc.nextInt ();
int n1 = n ; // stores the actual value.
String s = Integer.toString (n) ; // to convert to string.
int d = s.length () ; // to store the length of string s.
int arr [] = new int [n];
int i, sum ;
for (i=d-1; i>=0; i--) {
arr [i] = n1 % 10 ; // to store each digit.
 n1 = n1 / 10 ; }
i = d ; sum = 0 ;
while (sum < n)
{
  sum = 0 ;
```

```
for (int j=1; j<=d; j++) {
sum = sum + arr [i-j] ; }//calculates the sum.
arr [i] = sum ;
i++; }
if (sum==n) //checks whether both are same or not
  System.out.println(" Keith number.");
else
  System.out.println("Not a Keith number");
} }
```

VARIABLE DESCRIPTION.

VARIABLE	DATA TYPE	DESCRIPTION
n	int	to store the number entered from the user.
n1	int	to store the actual value.
s	string	to convert into string type.
d	int	to store the length of the string.
arr	int	array type - to store each digit.
sum	int	to store the sum of digits.

i	int	for loop variable.
j	int	for loop variable.

OUTPUT

```
Enter a number :
75
Keith Number.

Enter a number :
128
Not a Keith number.
```

ASSIGNMENT NUMBER - 16

TOPIC - Display calendar of any month of any year.

QUESTION

Write a program in Java to display calendar of any month of any year.

PROGRAMMING

```
import java.util.*;
public class calendar {
public static void main() {
Scanner sc = new Scanner (System.in);
System.out.print("Enter the year(e.g., 2001):");
int y = sc.nextInt();
System.out.print("Enter month in number:");
int m = sc.nextInt();
//Print Calendar for the month of the year.
if (m<1 || m>12 || y<1980)
  System.out.println("Wrong input!");
else
  printMonth(year, month); }
public static void printMonth(int y, int m) {
printMonthTitle(year, month);
printMonthBody(year, month); }
public static void printMonthTitle(int y, int m) {
System.out.println("          " + getMonthName(m) +
```

```
" " + year);
System.out.println("-------------------");
System.out.println("Sun Mon Tue Wed Thurs Fri Sat");
}
public static String getMonthName(int m){
String monthName = null;
switch (m) {
case 1: monthName = "January"; break;
case 2: monthName = "February"; break;
case 3: monthName = "March"; break;
case 4: monthName = "April"; break;
case 5: monthName = "May"; break;
case 6: monthName = "June"; break;
case 7: monthName = "July"; break;
case 8: monthName = "August"; break;
case 9: monthName = "September"; break;
case 10: monthName = "October"; break;
case 11: monthName = "November"; break;
case 12: monthName = "December"; }
return monthName; }// to store the name of month.
public static void printMonthBody(int y, int m) {
int sDay = getStartDay(y,m);
int nDay = getNumberOfDaysInMonth(y,m);
int i=0;
for(i=0; i<sDay; i++)
  System.out.print("  ");
```



```
for(i=1; i<=nDay; i++) {
if(i<10)
  System.out.print("  "+i);
else
  System.out.print(" "+i);
if((i+sDay)%7==0)
  System.out.println();
} System.out.println(); }
public static int getStartDay(int y, int m) {
int sDay1800 = 3; // Return the start Day.
int totalNumberOfDays = getTotalNumberOfDays(y,m);
return (totalNumberOfDays + sDay1800)%7; }
public static int getTotalNumberOfDays(int y, int m) {
int total = 0;
for(int i=1800; i<y; i++)// total days from 1800.
if(isLeapYear(i))
  total = total + 366;
else
  total = total + 365;
for(int i=1; i<m; i++)
  total = total + getNumberOfDaysInMonth(y,i);
return total; }
public static int getNumberOfDaysInMonth(int y, int m)
{ if(m==1||m==3||m==5||m==7||m==8||m==10
  ||m==12)
return 31;
```

```
if (m==4||m==6||m==9||m==11)
    return 30;
if (m==2) return isLeapYear(y)? 29:28;
return 0; // If month is incorrect.
} public static boolean isLeapYear(int y){
return y%400==0||(y%4==0&&y%100!=0);
} } //Determine if it is a leap year.
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
y	int	to store the year.
m	int	to store the month
i	int	for loop variable.
sDay	int	starting Day
nDay	int	number of days in the month.
total	int	total no of days. leap year or not

OUTPUT

```
Enter full year (e.g., 2001): 2012
Enter month in number: 12
December 2012
```

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

ASSIGNMENT NUMBER – 17

TOPIC – Factorial using Recursion.

QUESTION

Write a program to calculate factorial using Recursion

PROGRAMMING

```
import java.util.*;
public class factorial {
public static void main ( ) {
Scanner sc = new Scanner (System.in);//Scanner object
System.out.println("Enter the number: ");
int n = sc.nextInt ( );// Store entered value
int factorial = fact (n); // called user defined function
System.out.println ("Factorial of entered no": +
 factorial); }
public static int fact (int n) {
int output;
if (n==1) { //Checks whether the number is 1.
 return 1; }
//Recursion: Function calling itself
output = fact (n-1)*n;
return output; }}
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
n	int	to store the no entered by user.
factorial	int	to store the result of factorial of n.
fact	int	to calculate the factorial of value n.
output	int	to store the output of factorial.

OUTPUT

Enter the number:
5
Factorial of entered no: 120.

Enter the number:
10
Factorial of entered no:
3628800.

ASSIGNMENT NUMBER - 18

TOPIC - Linar Search.

QUESTION

Write a program in Java to search an element in an array using Linear Search.

PROGRAMMING

```
import java.util.*;
public class linearsearch {
public static void main() {
int c, n, i, a[] ; //To capture user input
Scanner sc = new Scanner (System.in);
System.out.println ("Enter number of elements:");
n = sc.nextInt(); // array to store all numbers.
a = new int [n];
System.out.println ("Enter "+n+" integers:");
// loop to store each numbers in array.
for (c=0; c<n; c++)
  a[c] = sc.nextInt();
System.out.println ("Enter the search value:");
i = sc.nextInt(); // enter the searching value.
for (c=0; c<n; c++) {
if (a[c] == i)
{
System.out.println (i+" is present at location "+(c+1));
```

```
/*Item is found so to stop the search & come
out of loop use break statement.*/
break;  } }
if (c==n)
System.out.println(i+" doesn't exist in array.");
} }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
c	int	Counter variable; for loop variable.
n	int	to store the number entered by user.
i	int	To store the value to be searched.
a[]	int	array type - to store the numbers.

OUTPUT

```
Enter number of elements :
6
Enter 6 integers :
22
33
45
1
3
99
Enter the search value :
45
45 is present at location 3

Enter number of elements :
4
Enter 4 integers :
11
22
4
5
Enter the search value :
99
99 doesn't exist in array.
```

ASSIGNMENT NUMBER – 19

TOPIC – Entered String in Alphabetical Order.

QUESTION

Write a program in java to enter a string and display it in alphabetical order.

PROGRAMMING

```
import java.util.*;
public class alphabeticalorder
{ public static void main () {
Scanner sc = new Scanner (System.in);
System.out.println ("Enter the string");
String st = sc.nextLine ();// input entered by user
int l = st.length ();// calculate the lngth of st.
st = st.toUpperCase ();// convert into capital letters
for (int i=65; i<=90; i++) {
  for (int j=0; j<l; j++)  {
char ch = st.charAt (j);
if (i==ch)// checks whether the alphabet is in the string
System.out.print ((char) i+ " ");
  }}
System.out.println ("String in alphabetical order");
  }} //output string in alphabetically.
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
st	String	to store the input entered by user.
l	int	to calculate the length of the string
i	int	for loop variable.
j	int	for loop variable.
ch	char	to store each letters of the string

OUTPUT

Enter the String:
Loyola
ALLOY
String in alphabetical order

ASSIGNMENT NUMBER – 20

TOPIC – Sum of 5x5 array.

QUESTION

Write a program in java to generate sum of all elements of a double dimensional array of 5x5 subscripts.

PROGRAMMING

```
import java.util.*;
public class Sumofarrays {
public static void main () {
Scanner sc = new Scanner (System.in);
System.out.println ("Enter 1st 5x5 array:");
int [][]a= new int [5][5] ; // First array
int [][]b = new int [5][5] ; // Second array
for (int i=1; i<=5; i++) {
 for (int j=1; j<=5; j++) {
a[i][j] = sc.nextInt(); } }
System.out.println ("Enter 2nd 5x5 array:");
for (int i=1; i<=5; i++) {
 for (int j=1; j<=5; j++) {
b[i][j] = sc.nextInt(); } }
int [][] c = new int [5][5]; // Resultant array.
for (i=1; i<=5; i++) {
 for (j=1; j<=5; j++) {
```

```
c[i][j] = a[i][j] + b[i][j]; // Sum of the arrays.
System.out.print(c[i][j] + "  "); } //output
System.out.println(); } }
```

VARIABLE DESCRIPTION

VARIABLE	DATA TYPE	DESCRIPTION
i	int	for loop variable.
j	int	for loop variable.
a[][]	int	Stores the elements of the first array
b[][]	int	Stores the elements of the second array
c[][]	int	stores the resultant array.

OUTPUT

```
Enter 1st 5x5 array:
5
7
11
12
17
19
21
14
16
18
79
29
64
65
68
72
76
67
52
58
49
32
36
89
92
Enter 2nd 5x5 array
72
76
59
42
```

62
34
81
79
24
25
101
38
29
66
61
33
45
98
93
46
78
77
62
65
56

77	83	70	54	79
53	102	93	40	43
180	67	93	131	129
105	121	165	145	104
127	109	98	154	148

CONCLUSION