Authpur National Model Higher Secondary School

Name: - Srinjay Das Gupta

Class- XI- Science

Roll No.-

Subject- Computer Science.

|  |  |  |
| --- | --- | --- |
| Sl.No. | Topic | Page No. |
| 1. | Introduction | 3 |
| 2. | Half-Yearly Assignments | 4-27 |
| 3. | Annual Assignments | 28-63 |
| 4. | Bibliography | 64 |

***INDEX***

*Introduction*

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to *byte code* that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open source software and used by most developers including the Eclipse IDE and is the default JVM for almost all Linux distributions.

--x--

**HALF-YEARLY ASSIGNMENTS**

**Program: 1.** Write a program to find the digital root of a number.

[Digital root of a number is the single digit that results from the continuous summation of the digits of a

Number and the numbers resulting from each summation .E.g. consider the number 378,

Sum of its digits = 3+7+8 = 18, 1+8 =9

So Digital root of 378 = 9]

**ALGORITHM:**

1. Start
2. Declare variables n, n1, s and d
3. Read n from user.
4. Initialize variables s=0 , n1= n
5. Repeat the following steps until false
   1. If n1 is not equal to 0, find the remainder of division of n1 by 10
   2. Add the remainders to s.
   3. If s<10, move to Step 6
   4. Else overwrite n1=s and s=0 and continue the steps.
6. Print s as the Digital Root of n.
7. End

**Source Code:**

import java.util.\*;

class digital\_Root

{

int digitalRoot(int n)

{

int n1,s=0,d;

n1=n;

while(true)

{

while(n1!=0)

{

d=n1%10;

s=s+d; //Finding the sum of the digits

n1/=10;

}

if(s<10) //Digital root is a one digit no. thus s<10

break;

else

{

n1=s;

s=0;

continue;

}

}

return s;//Returning s as Digital root

}

public static void main(String ars[])

{

int n,dr;

Scanner in=new Scanner(System.in);

System.out.println("Enter a no.");

n=in.nextInt();//Reading n from user

digital\_Root ob=new digital\_Root();//calling digital\_Root()

dr=ob.digitalRoot(n);

System.out.println("The Digital Root of "+n+" is "+dr);

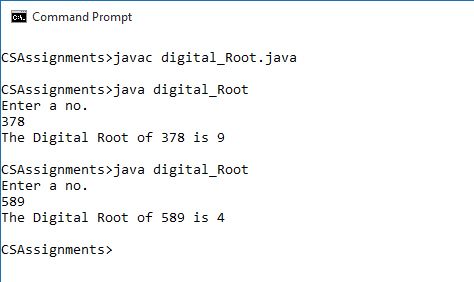
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| n | int | Accept value from user and a parameter for digital\_Root() |
| n1 | int | Temporary variable to store the value of n |
| s | int | Stores the sum of the digits of n1 |
| d | int | Stores the digits of n1 |
| dr | int | Calls digital\_Root and stores the value returned |

**Input / Output Screen**

****

**Program 2:** Write a program to print all the Prime Palindrome numbers in the given range.

**ALGORITHM:**

1. Start
2. Declare variables start, end, i, j and c.
3. Read start and end from user
4. Run loop from i=start till i=end.
5. Check for prime number
   1. Run a loop from j=1 to j=i
   2. If i%j=1 increment c by 1
   3. Outside the loop check if c=2, then return 1 and overwrite c=0
   4. Else return 0 and overwrite c=0
6. Check for palindrome.
   1. Declare variables n, r and d.
   2. Initialise n=i and r=0
   3. Repeat the following processes until n is 0.
      1. Find the remainder of the division of n by 10 and store in d
      2. Multiply d by 10 and add it to r. store this in r
      3. Store the quotient of the division of n by 10 in n
   4. If r is equal to i return 1
   5. Else return 0
7. If i is Prime as well as palindrome print ii.
8. End

**Source code:**

import java.util.\*;

class primePalindrome

{

int start,end;

primePalindrome(int a,int b)

{

start=a;end=b;//Initialising data members by parameterised constructor

}

int isPrime(int i)//Checking Prime no.

{

int j,c=0;

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

{

if(i%j==0)

c++;

}

if(c==2)

return 1;

else

return 0;

}

int isPalin(int i)// Checking for palindrome no.

{

int n,j,r=0,d;

n=i;

while(n!=0)

{

d=n%10;

r=r\*10+d;

n/=10;

}

if(r==i)

return 1;

else

return 0;

}

void generate()//Printing Prime-palindrome no.s

{

int i;

System.out.println("The Prime palindrome no.s b/w "+start+" and "+end+" are ");

for(i=start;i<=end;i++)

{

if(isPrime(i)==1&&isPalin(i)==1)

System.out.println(i+" ");

}

}

public static void main(String ars[])

{

int a,b;

System.out.println("Enter the start and end points");

Scanner in=new Scanner(System.in);

a=in.nextInt();

b=in.nextInt();

primePalindrome ob=new primePalindrome(a,b);

ob.generate();

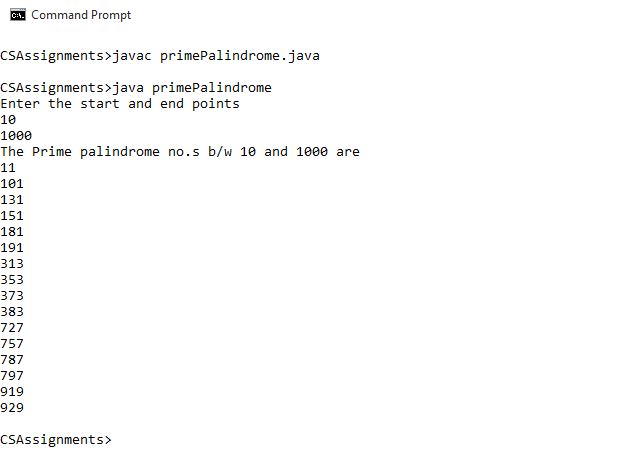
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| start | int | To store the starting point |
| end | int | To store the ending point |
| i | int | Loop variable and parameter for isPalin() and isPrime() |
| j | int | Loop variable |
| a | int | Parameter for constructor |
| b | int | Parameter for constructor |
| r | int | Stores the reverse of the no. |
| d | int | Store the digits of |
| c | int | Counts the factors |

**Input / Output Screen**

****

**Program 3:** Write a program to take a binary number and convert it into Decimal number.

**ALGORITHM:**

1. Start
2. Declare variables dec\_out, dec1, dec2, bin\_s, bin\_in, bin\_frac, pi, i, p, ch
3. Initialise dec1=0.0 and dec2=0.0
4. Read bin\_s from user
5. Find the index of ‘.’ In bin\_s and store in ‘pi’
6. Separate the integral and the fractional part of bin\_s into bin\_in and bin\_frac respectively
7. Store the length of bin\_in in p
8. Declare a temporary variable k and initialise it to 0
9. Run a loop from i=(p-1) until i<=1 and repeat the following
   1. Store each character of bin\_in in ch
   2. Check if the integral value of ch is>1
   3. If true terminate the program
   4. Else declare a temporary variable temp to store the integral value of c
   5. Increment the value of dec1 by (temp\*2^k)
   6. Increment k by 1
10. Repeat from Step 7 for bin\_frac but in Step 9 e dec2 will be incremented
11. Concatenate dec1 and dec2 in dec\_out
12. Return dec\_out
13. End

**Source Code:**

import java.util.\*;

class binarytodecimal

{

double converter(double bin)

{

double dec\_out,dec1=0.0,dec2=0.0;String bin\_s,bin\_in,bin\_frac;int pi,i,p;char ch;

bin\_s=Double.toString(bin);

pi=bin\_s.indexOf(".");

bin\_in=bin\_s.substring(0,pi);

bin\_frac=bin\_s.substring(pi+1);

p=bin\_in.length();

int k=0;

for(i=(p-1);i>=0;i--)

{

ch=bin\_in.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>1)

{

System.out.println("Invalid Binary No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec1=dec1+(temp\*Math.pow(2,k));

k++;

}

p=bin\_frac.length();

for(i=0;i<p;i++)

{

ch=bin\_frac.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>1)

{

System.out.println("Invalid Binary No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec2=dec2+(temp\*Math.pow(2,-(i+1)));

}

dec\_out=dec1+dec2;

return dec\_out;

}

public static void main(String ars[])

{

double bin;

Scanner in=new Scanner(System.in);

System.out.println("Enter a binary no. (fractions included)");

bin=in.nextDouble();

binarytodecimal ob=new binarytodecimal();

System.out.println("The decimal equivalent of "+bin+" is "+(ob.converter(bin)));

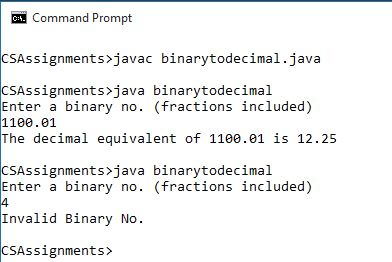
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Function** |
| dec\_out | double | Store the decimal equivalent |
| dec1 | double | Store the integral decimal |
| dec2 | double | Store the fractional decimal |
| bin\_s | String | Input from user |
| bin\_in | String | Integral part of bin\_s |
| bin\_frac | String | Fractional part of bin\_s |
| pi | int | Index of ‘.’ in bin\_s |
| i | int | Loop variable |
| p | int | Stores the lengths of bin\_in and bin\_frac |
| k | int | Stores the power of 2 |
| temp | int | Store the integral value of ch |
| ch | char | Store each of charac ter of bin\_in and bin\_frac |

**Input/ Output Screen**



**Program 4:** Write a program to accept an Octal number and convert it into Decimal number.

**ALGORITHM:**

1. Start
2. Declare variables dec\_out, dec1, dec2, oct\_s, oct\_in, oct\_frac, pi, i, p, ch
3. Initialise dec1=0.0 and dec2=0.0
4. Read oct\_s from user
5. Find the index of ‘.’ In oct\_s and store in ‘pi’
6. Separate the integral and the fractional part of oct\_s into oct\_in and oct\_frac respectively
7. Store the length of oct\_in in p
8. Declare a temporary variable k and initialise it to 0
9. Run a loop from i=(p-1) until i<=1 and repeat the following
   1. Store each character of oct\_in in ch
   2. Check if the integral value of ch is>1
   3. If true terminate the program
   4. Else declare a temporary variable temp to store the integral value of c
   5. Increment the value of dec1 by (temp\*8^k)
   6. Increment k by 1
10. Repeat from Step 7 for oct\_frac but in Step 9 (e) dec2 will be incremented
11. Concatenate dec1 and dec2 in dec\_out
12. Return dec\_out

**Source Code:**

import java.util.\*;

class OctToDec

{

double converter(double oct)

{

double dec\_out,dec1=0.0,dec2=0.0;String oct\_s,oct\_in,oct\_frac;int pi,i,p;char ch;

oct\_s=Double.toString(oct);

pi=oct\_s.indexOf(".");

oct\_in=oct\_s.substring(0,pi);

oct\_frac=oct\_s.substring(pi+1);

p=oct\_in.length();

int k=0;

for(i=(p-1);i>=0;i--)

{

ch=oct\_in.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>7)

{

System.out.println("Invalid Octal No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec1=dec1+(temp\*Math.pow(8,k));

k++;

}

p=oct\_frac.length();

for(i=0;i<p;i++)

{

ch=oct\_frac.charAt(i);

if(Integer.parseInt(String.valueOf(ch))>7)

{

System.out.println("Invalid Octal No.");

System.exit(0);

}

int temp=Integer.parseInt(String.valueOf(ch));

dec2=dec2+(temp\*Math.pow(8,-(i+1)));

}

dec\_out=dec1+dec2;

return dec\_out;

}

public static void main(String ars[])

{

double oct;

Scanner in=new Scanner(System.in);

System.out.println("Enter a octal no. (fractions included)");

oct=in.nextDouble();

OctToDec ob=new OctToDec();

System.out.println("The decimal equivalent of "+oct+" is "+(ob.converter(oct)));

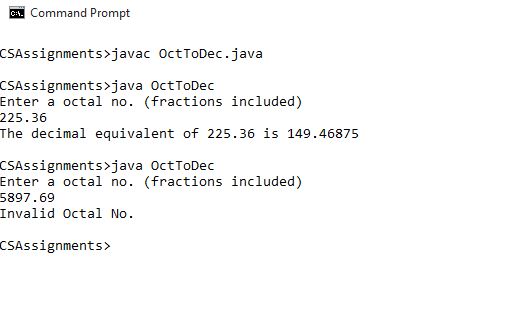
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Function** |
| dec\_out | double | Store the decimal equivalent |
| dec1 | double | Store the integral decimal |
| dec2 | double | Store the fractional decimal |
| oct\_s | String | Input from user |
| oct\_in | String | Integral part of oct\_s |
| oct\_frac | String | Fractional part of oct\_s |
| pi | int | Index of ‘.’ in oct\_s |
| i | int | Loop variable |
| p | int | Stores the lengths of oct\_in and oct\_frac |
| k | int | Stores the power of 8 |
| temp | int | Store the integral value of ch |
| ch | char | Store each of charac ter of oct\_in and oct\_frac |

**Input/ Output Screen**

****

**Program 5:** Write a program to accept a decimal number and convert it into binary number.

**ALGORITHM:**

1. Declare variables dec, d, bin, dec\_in, dec\_frac and i
2. Initialise bin to null (i.e.,””)
3. Read dec from user
4. Store the integral value of dec in dec\_in
5. Store the fractional value of dec in dec\_frac
6. Execute the following steps until dec\_in=0
   1. Store the remainder of the division of dec\_in by 2 in d
   2. Append d to bin
   3. Divide dec\_in by 2
7. Execute the following steps from i=0 till i=5
   1. Multiply dec\_frac by 2
   2. Declare a variable frac\_bit and store the integral part of dec\_frac(multiplied by 2) in frac\_bit
   3. Check if frac\_bit is 1
   4. If true append 1 to bin
      1. Subtract ii from dec\_frac
      2. Append ‘1’ to bin
   5. Else append 0 to bin
8. Return bin

**Source Code:**

import java.util.\*;

class decToBin

{

String converter(double dec)

{

String bin=""; int dec\_in,d;double dec\_frac;

dec\_in=(int)dec;

dec\_frac=dec-dec\_in;

while(dec\_in!=0)

{

d=dec\_in%2;

bin=Integer.toString(d)+bin;

dec\_in/=2;

}

bin=bin+".";

for(int i=0;i<5;i++)

{

dec\_frac\*=2;

int fract\_bit = (int) dec\_frac;

if (fract\_bit == 1)

{

dec\_frac -= fract\_bit;

bin+= (char)(1 + '0');

}

else

{

bin += (char)(0 + '0');

}

}

return bin;

}

public static void main(String ars[])

{

Scanner in=new Scanner (System.in);

double dec;String bin;

System.out.println("Enter the decimal no.");

dec=in.nextDouble();

decToBin ob=new decToBin();

bin=ob.converter(dec);

System.out.println("The binary equivalent of "+dec+" is "+bin);

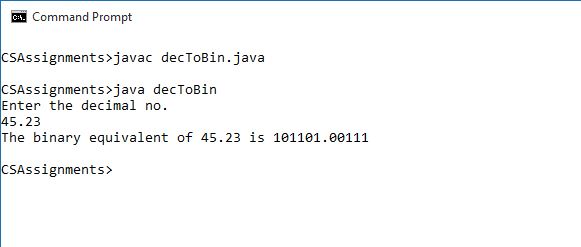
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| bin | String | Store the Binary equivalent |
| dec |  | Accept decimal number from User |
| d | int | Remainder of division of dec\_in and dec\_frac by 2 |
| dec\_in | int | Store the integral part of dec |
| dec\_frac | double | Store the fractional part of dec |
| frac\_bit | int | Store each bit of the fractional part of dec |

**Input/ Output Screen**



**Program 6:** Write a program to check whether a given number is fascinating number or not.

[A fascinating number is one which when multiplied by 2 and 3 and then the results are concatenated with the original number, the new number contains all the digits from 1 to 9 exactly once. E.g. – 192

192 × 2 = 384

192 × 3 = 576

After concatenating, (192) + (384) + (576) = 192384576, contains all the digits 1 to 9 exactly once.

So 192 is a Fascinating number.]

**ALGORITHM:**

1. Start
2. Declare variables n, n2, n3, i, j, f, N.
3. Initialise f to 1.
4. Read n from user.
5. Check if n is a 3 digit number or not
6. If true multiply n by 2 and store in n2
7. Multiply n by 3 and store in n3.
8. Concatenate n, n2 and n3 in N.
9. Run a loop from i= ‘1’ till i= ‘9’
   1. Initialise c to 0.
   2. Run another loop from j=0 till j=N.length().
      1. Store each character of N in ch
      2. If ch=i increment c
   3. If c is greater than 1 or equal to 0 break the loop and overwrite f to 0
10. Check if f is 1
11. If true n is a fascinating number.
12. End.

**Source Code:**

import java.util.\*;

class Fascinating

{

void check(int n)

{

int n2,j,n3,f=1;String N;char i;

if(n>=100)

{

n2=n\*2;

n3=n\*3;

N=Integer.toString(n)+""+Integer.toString(n2)+""+Integer.toString(n3);

for(i='1';i<='9';i++)

{

int c=0;

for(j=0;j<N.length();j++)

{

char ch=N.charAt(j);

if(ch==i)

c++;

}

if(c>1||c==0)

{

f=0;break;

}

}

if(f==1)

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

else

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

}

else

System.out.println ("Invalid Input");

}

public static void main(String ars[])

{

int n;

Scanner in =new Scanner(System.in);

System.out.println("Enter a no. of 3 digits ");

n=in.nextInt();

Fascinating ob=new Fascinating();

ob.check(n);

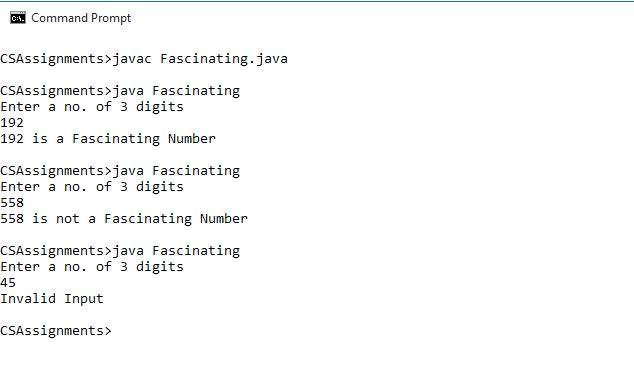
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| n | int | Accepts number from user |
| n2 | int | Store n\*2 |
| n3 | int | Store n\*3 |
| N | String | Store the concatenation of n, n2, n3 |
| i | char | Loop variable |
| j | int | Loop variable |
| f | int | Flag variable |
| ch | char | Store each character of N |

**Input/ Output Screen**



**Annual Assignments**

**Program 7:** Write a program to accept a Binary number and print its 1’s complement and 2’s complement.

**ALGORITHM:**

1. Start
2. Declare variable bin
3. Read bin from User
4. Calculate 1’s Complement
   1. Declare variables one, p, i, c,
   2. Initialise one to null (“”)
   3. Initialise p to the length of bin
   4. Run a loop from i=0 till i=p
      1. Store each character of bin in c
      2. If c is ‘1’ append ‘0’ to one
      3. If c is ‘0’ append ‘1’ to one
   5. Return one as 1’s complement
5. Calculate 2’s Complement
   1. Declare variables two, p, i, last, c
   2. Initialise two to null(“”)
   3. Store the last index of ‘1’ in last
   4. Store the length of bin in p
   5. Append the substring of bin from last to p in two
   6. Overwrite p to the length of the substring of bin till last
   7. Run a loop from i=p-1 till i=0
      1. Store each character of length of the substring of bin till last in c
      2. If c is ‘1’ append two to ‘0’
      3. If c is ‘0’ append two to ‘1’
   8. Return two as 2’s Complement
6. Print the 1’s and 2’s Complement
7. End

**Source Code:**

import java.util.\*;

class ones\_twos

{

String ones(String bin)

{

String one="";int p,i;char c;

p=bin.length();

for(i=(p-1);i>=0;i--)

{

c=bin.charAt(i);

if(Integer.parseInt(String.valueOf(c))>1)

{

System.out.println("Invalid Binary No.");

System.exit(0);

}

}

for(i=0;i<p;i++)

{

c=bin.charAt(i);

if(c=='1')

one=one+"0";

else

one=one+"1";

}

return one;

}

String twos(String bin)

{

String two="";int p,i,last;char c;

last=bin.lastIndexOf("1");

p=bin.length();

two=two+bin.substring(last,p);

p=(bin.substring(0,last)).length();

for(i=p-1;i>=0;i--)

{

c=(bin.substring(0,last)).charAt(i);

if(c=='1')

two="0"+two;

else

two="1"+two;

}

return two;

}

public static void main(String ars[])

{

String bin,one, two;

Scanner in=new Scanner(System.in);

System.out.println("Enter a Binary no.");

bin=in.next();

ones\_twos ob=new ones\_twos();

one=ob.ones(bin);

two=ob.twos(bin);

System.out.println("One's="+one);

System.out.println("Two's="+two);

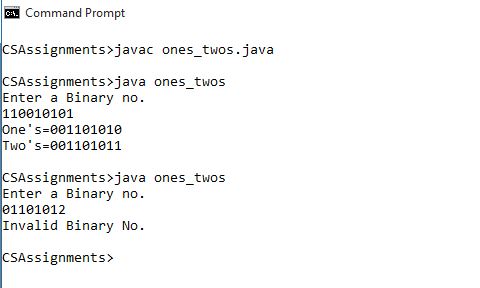
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| bin | String | Accept binary number from user |
| p | int | Store the length of bin |
| i | int | Loop variable |
| one | String | Store the 1’s complement |
| two | String | Store the 2’s complement |
| c | char | Store each character of bin |
| last | int | Store the last index of 1 in bin |

**Input/ Output Screen**

****

**Program 8:** Design a class with the following specification :-

Class name : Mersenne

Data members : num (long type )

Member methods : -

int is Mersenne ( ) : checks whether the number is Mersenne or not .

int is DoubleMersenne( ): checks whether the number is double Mersenne or not.

void genMersenne (): generates Mersenne numbers.

void genDoubleMersenne () : generates Double Mersenne numbers.

Design a constructor to initialize the data members with the value of parameter. Define the main() also to execute the functions properly.

**ALGORITHM:**

1. Start
2. Declare variable num
3. Initialise num using a parameterised constructor
4. Check Mersenne number
   1. Declare variable n, i, f
   2. Initialise n and f to 0
   3. Run an infinite loop from i=1
      1. Overwrite n to [(2^i)-1]
      2. If n=num overwrite f to 1 and break the loop
   4. Return f
5. Check Double Mersenne number
   1. Declare variable n, i, f, pwr
   2. Initialise n, f, pwr to 0
   3. Run an infinite loop from i=1
      1. Overwrite pwr to [(2^i)-1]
      2. Overwrite n to [(2^pwr)-1]
      3. If n=num overwrite f to 1 and break the loop
   4. Return f
6. Generate Mersenne number
   1. Initialise num to 0
   2. Run a loop from i=1 to 10
      1. Overwrite num to [(2^i)-1]
      2. Print num
7. Generate double Mersenne
   1. Initialise num to 0\
   2. Declare variable pwr
   3. Initialise pwr to 0
   4. Run a loop from i=1 to 10
      1. Overwrite pwr to [(2^i)-1]
      2. Overwrite num to [(2^pwr)-1]
      3. Print num
8. End

**Source Code:**

import java.util.\*;

class Mersenne

{

long num;

Mersenne(long n)

{

num=n;

}

int isMersenne()

{

long n=0;int i,f=0;

for( i=1;;i++)

{

n=(long)(Math.pow(2,i)-1);

if(n==num){

f=1;

break;

}

}

return f;

}

int isDoubleMersenne()

{

long n=0, pwr=0;int f=0;

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

{

pwr=(long)Math.pow(2,i)-1;

n=(long)(Math.pow(2,pwr)-1);

if(n==num)

{

f=1;

break;

}

}

return f;

}

public void genMersenne(){

num=0;

System.out.println("Generated Mersenne numbers are :");

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

{

num=(long)(Math.pow(2,i)-1);

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

}

System.out.println();

}

public void genDoubleMersenne()

{

num=0;long pwr=0;

System.out.println("Generated double Mersenne Numbers are:");

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

{

pwr=(long)Math.pow(2,i)-1;

num=(long)(Math.pow(2,pwr)-1);

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

}

System.out.println();

}

public static void main(String ars[])

{

int N;

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

Scanner in=new Scanner(System.in);

N=in.nextInt();

Mersenne ob=new Mersenne(N);

if (ob.isMersenne()==1)

System.out.println(N+" is a Mersenne");

else

System.out.println(N+" is not a Mersenne");

if (ob.isDoubleMersenne()==1)

System.out.println(N+" is a Double Mersenne");

else

System.out.println(N+" is not a Double Mersenne");

ob.genMersenne();ob.genDoubleMersenne();

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data type | Function |
| num | long | Accept number from user |
| pwr | long | Exponent for checking and generation of double Mersenne no. |
| i | int | Loop variable |
| f | int | Flag variable |

**Input/ Output Screen**



**Program 9:** Write a program to accept a decimal number and convert it into Octal number.

**ALGORITHM:**

1. Start
2. Declare variable dec , r, n, oct
3. Accept dec from user
4. Initialise oct to null (“”)
5. Run a loop till n is not equal to 0
   1. Store the remainder of the division of n by 8 in r
   2. Append the string value of r to oct
   3. Divide n by 8 and store in n
6. Return oct
7. End

**Source Code:**

import java.util.\*;

class decToOct

{

String converter(int dec)

{

int r,n;

String oct="";

while(dec!=0)

{

r=dec%8;

oct=Integer.toString(r)+oct;

dec=dec/8;

}

return oct;

}

public static void main(String ars[])

{

decToOct ob=new decToOct();

Scanner in=new Scanner(System.in);

int dec;

String oct;

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

dec=in.nextInt();

oct=ob.converter(dec);

System.out.println("Octal="+oct);

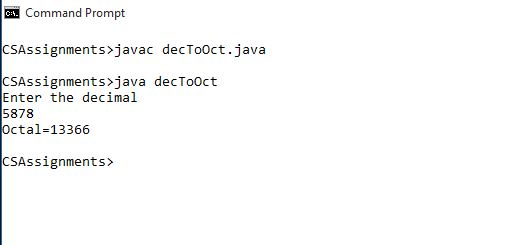
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| dec | int | Accept decimal no from user |
| oct | String | Store the Octal equivalent of dec |
| r | int | Store the remainder of the division of dec by 8 |

**Input / Output Screen**

****

**Program 10:** Write a program to accept a decimal number and convert it into Hexadecimal number.

**ALGORITHM:**

1. Start
2. Declare variable r, n, hex, dec
3. Accept dec from User
4. Initialise n to dec
5. Declare a character array dig[] and initialise it with {‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’}.
6. Execute the following while n>0
   1. Store the remainder of division of n by 16
   2. Append hex to the element in dig[] of index r
   3. Divide n by 16 and store it in n
7. Return hex as the hexadecimal equivalent of n

**Source Code:**

import java.util.\*;

class decToHex

{

String converter(int dec)

{

int r,n;

String hex="";

n=dec;

char dig[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

while(n!=0)

{

r=n%16;

hex=dig[r]+hex;

n=n/16;

}

return hex;

}

public static void main(String ars[])

{

decToHex ob=new decToHex();

Scanner in=new Scanner(System.in);

int dec;

String hex;

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

dec=in.nextInt();

hex=ob.converter(dec);

System.out.println("Hexadecimal="+hex);

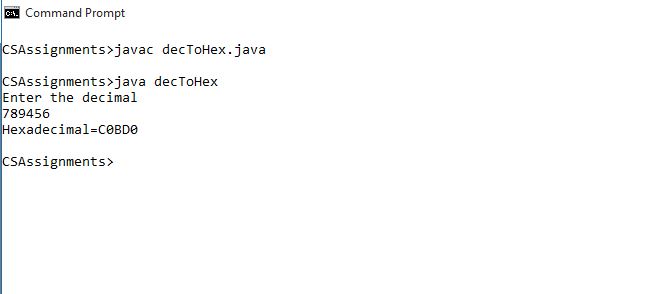
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data type | Function |
| n | int | Stores the number given by the user |
| r | int | Stores the remainder of the division of n by 16 |
| hex | String | Stores the Hexadecimal equivalent of n |
| dec | int | Accept decimal value from User |
| dig | char | Character array to store the digits of the hexadecimal number System |

**Input / Output Screen**



**Program 11:** Write a program to accept a Hexadecimal number and convert it into decimal number.

**ALGORITHM:**

1. Start
2. Declare variables hexin, hex, dec, p, i, n, k
3. Read hexin from User
4. Initialise hex to “0123456789ABCDEF” and dec to 0
5. Turn hexin to Uppercase
6. Store the length of hexin in p
7. Initialise k to (p-1)
8. Run a loop from i=0 till i<p and execute the following
   1. Declare a variable ch and store in it each character of hexin
   2. Store in n the index of ch in hex
   3. Increment dec by [n\*(16^k)]
   4. Decrement k by 1
   5. If k<0 break the loop
9. Return dec

**Source Code:**

import java.util.\*;

class HEXtoDEC2

{

int converter(String hexin)

{

String hex="0123456789ABCDEF";

int dec=0,i,p,n,k;

hexin=hexin.toUpperCase();

p=hexin.length();

k=p-1;

for(i=0;i<p;i++)

{

char ch=hexin.charAt(i);

if((int)ch>(int)('F'))

{

System.out.println("Invalid Hexadecimal no.");

System.exit(0);

}

n=hex.indexOf(ch);

dec=dec+(int)(n\*(Math.pow(16,k)));

k--;

if(k<0)

break;

}

return dec;

}

public static void main(String ars[])

{

HEXtoDEC2 ob=new HEXtoDEC2();

String hexin;

int dec;

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

Scanner in=new Scanner(System.in);

hexin=in.next();

hexin=hexin.toUpperCase();

dec=ob.converter(hexin);

System.out.println ("The Decimal value is "+dec);

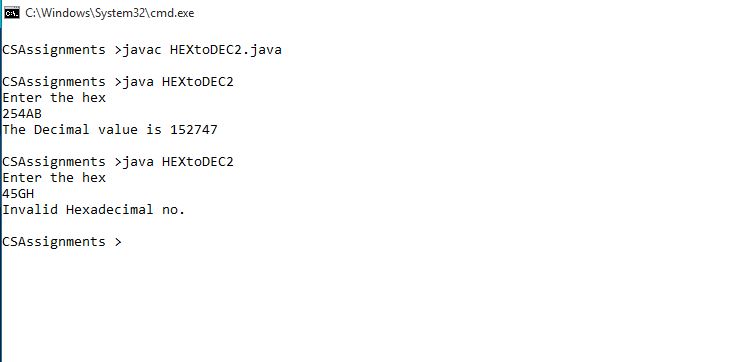
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data type | Function |
| hexin | String | Accept hexadecimal number from User |
| hex | String | Store the digits of hexadecimal number System |
| p | int | Stores the length of hexin |
| i | int | Loop variable |
| n | int | Store the index of each character of hexin wih respect to hex |
| k | int | Stores the power of 16 |
| ch | char | Stores each character of hexin |
| dec | int | Reads decimal value from User |

**Input / Output Screen**



**Program 12:** Write a program to accept a square matrix of size N and find its Saddle point.

**ALGORITHM:**

1. Start
2. Declare variables rowmin, col, i, j, n, p, sp=1
3. Read the size of square matrix from user
4. Declare a 2d array of size n × n
5. Store the length of arr in p
6. Enter the elements of matrix in arr from user
7. Print them in matrix form
8. Run a loop from i=0 till i<p and execute the following
   1. Store the element arr[i][0] in rowmin
   2. Initialise col to 0 and sp to 1
   3. Run a loop from j=1 till j<arr[i].length
      1. Check if arr[i][j]<rowmin
      2. If true overwrite rowmin to arr[i][j] and col to j
   4. Run another loop from j=0 till j<p
      1. Check if arr[j][col]>rowmin
      2. If true overwrite sp to 0 and terminate the loop
   5. Check if sp=1
   6. If true print rowmin as saddle point
9. If sp is 0 print there are no saddle point(s) in the matrix
10. End

**Source code:**

import java.util.\*;

class Saddle\_Point

{

void saddle()

{

int rowmin,col,i,j,n,p,sp=1;

Scanner in=new Scanner(System.in);

System.out.println("Enter the size of square matrix");

n=in.nextInt();

int arr[][]=new int[n][n];

p=arr.length;

System.out.println("Enter the matrix elements");

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

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

arr[i][j]=in.nextInt();

System.out.println("The Matrix entered");

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

{

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

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

System.out.println();

}

for(i=0;i<p;i++)

{

rowmin=arr[i][0];

col=0;sp=1;

for(j=1;j<arr[i].length;j++)

{

if(arr[i][j]<rowmin)

{

rowmin=arr[i][j];

col=j;

}

}

for(j=0;j<p;j++)

{

if(arr[j][col]>rowmin)

{

sp=0;

break;

}

}

if(sp==1)

System.out.println("The Saddle Point is "+rowmin);

}

if(sp==0)

System.out.println("No saddle point");

}

public static void main(String ars[])

{

Saddle\_Point ob=new Saddle\_Point();

ob.saddle();

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| rowmin | int | To store the minimum value of the row |
| col | int | To store the column number of the minimum value of the row |
| i, j | int | Loop variable |
| n | int | To store the size of the matrix |
| p | int | To store the length of the array |
| sp | int | Flag variable to find the saddle point |
| arr | int | 2-D Array to store the matrix entered by the user |

**Input / Output Screen**



**Program 13:** Write a program to accept a sentence which may be terminated by either ‘.’ or ‘?’ or ‘!’ and convert it into

Upper case .Now form a new sentence by arranging words in ascending order of number of characters

Present in each word.

Sample input-1: It is a String program.

Sample output-1: IT IS A STRING PROGRAM.

Converted string: AN IT IS A STRING PROGRAM.

Sample input-2: it is amazing %

Sample output -2: INVALID INPUT

**ALGORITHM:**

1. Start
2. Declare variables i, j, k, l, m, w, temp, sin
3. Initialise k to 0, m to 0, w to null (“”)
4. Read sin from User
5. Check if sin ends with ‘.’ or ‘!’ or ‘?’
6. If true execute the following
   1. Change sin to uppercase
   2. Print sin
   3. Make a substring of sin till it reaches ‘.’ or ‘!’ or ‘?’
   4. Add a white space at the end of sin
   5. Store the length of sin in l
   6. Run a loop from i=0 till i<l
      1. Store each character in ch
      2. If ch is a white space increment k by 1
   7. Declare a string array sar[] of length k
   8. Run a loop from i=0 till i<l
      1. Store each character of sin in ch
      2. If ch is not a white space append ch to w
      3. Else
         1. Add w to the mth element of sar
         2. Increment m by 1
         3. Overwrite w to null (“”)
   9. Run a loop from i=0 till i<(k-1)
      1. Run another loop from j=0 till j<(k-1-i)
      2. If jth element of sar is a ‘.’ or ‘!’ or ‘?’ skip the iteration
      3. If the length of the jth element of sar is greater than (j+1)th term of sar
         1. Initialise temp to sar[j]
         2. Overwrite sar[j] to sar[j+1]
         3. Overwrite sar[j+1] to temp
   10. Print sar[] as the arranged sentence
7. Else Print “invalid input”
8. End

**Source Code:**

import java.util.\*;

class Sentence

{

void Sent(String sin)

{

int i,j,k=0,l,m=0;

String w="",temp;

if(sin.endsWith(".")||sin.endsWith("!")||sin.endsWith("?"))

{

sin=sin.toUpperCase();

System.out.println(sin);

if(sin.endsWith("."))

sin=sin.substring(0,(sin.indexOf(".")));

else if(sin.endsWith("!"))

sin=sin.substring(0,(sin.indexOf("!")));

else

sin=sin.substring(0,(sin.indexOf("?")));

sin=sin+" ";

l=sin.length();

for(i=0;i<l;i++)

{

char ch=sin.charAt(i);

if(ch==' ')

k++;

}

String sar[]=new String[k];

for(i=0;i<l;i++)

{

char ch=sin.charAt(i);

if(ch!=' ')

w=w+ch;

else

{

sar[m]=w;

w="";

m++;

}

}

for(i=0;i<(k-1);i++)

{

for(j=0;j<(k-1-i);j++)

{

if(sar[j].equals(" . ") || sar[j].equals(" ! ") || sar[j].equals(" ? "))

continue;

if(sar[j].length()>sar[j+1].length())

{

temp=sar[j];

sar[j]=sar[j+1];

sar[j+1]=temp;

}

}

}

System.out.println("The arranged sentence=");

for(i=0;i<k;i++)

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

}

else

System.out.println("Invali dinput");

}

publicstaticvoidmain(Stringars[])

{

String sin;

Scanner in=new Scanner(System.in);

System.out.println("Enterasentence");

sin=in.nextLine();

Sentence ob=new Sentence();

ob.Sent(sin);

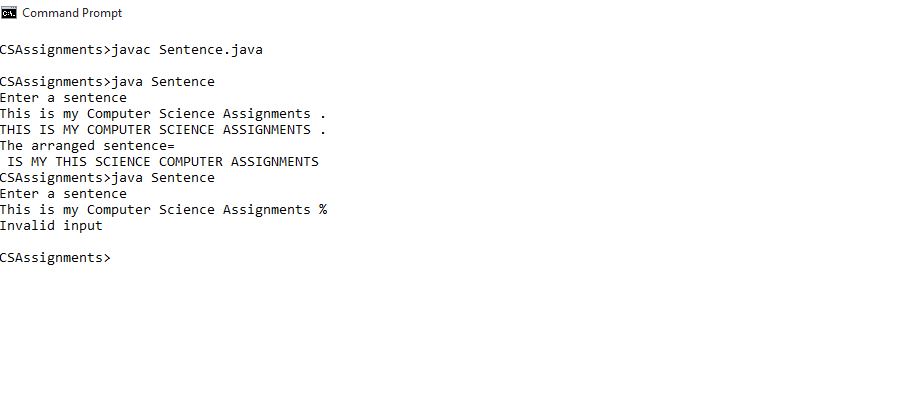
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| i,j | int | Loop variables |
| k | int | Counter for the size of array sar[] |
| l | int | Store the length of the sentence |
| m | int | Acts as index for the array sar[] |
| w | String | Stores each word of sin |
| temp | String | Temporary variable for swapping the words |
| sar | String | String array to store the words of sin and to arrange in ascending order |
| sin | String | Reads the sentence from user |

**Input / Output Screen**



**Program 14:** Write a program to accept a sentence may be terminated by either ‘.’ or ‘?’.

a. Now find the words beginning and ending with a vowel.

b. Place the words which begin end with vowel at the beginning followed by the remaining words as

they occur in the sentence .

Sample Input -1: YOU MUST AIM TO BE A BETTER PERSON TOMORROW THAN YOU ARE TODAY .

Sample output-1 : NUMBER OF WORDS BEGIN AND END WITH A VOWEL = 2

A ARE YOU MUST AIM TO BE BETTER PERSON TOMORROW THAN YOU TODAY .

Sample input -2: WHO YOU ARE @

Sample output-2: INVALID INPUT

**ALGORITHM:**

1. Start
2. Declare variables sa, w, i, l, k, m, s, ch
3. Read s from User
4. Initialise sa and w to null (“”) ; and k to 0
5. If s ends with ‘.’ or ‘!’
   1. Append a white space at the end of s
   2. Change s to uppercase
   3. Make a substring of s till ‘.’
   4. Store the length of s in l
   5. Run a loop from i=0 till i<l and execute the following
      1. Store each character of s in ch
      2. If ch is not a white space append ch to w
      3. Else execute the following
         1. If w ends and starts with a vowel then increment k by 1 and append a white space and w to sa
         2. Overwrite w to null (“”)
      4. Else append sa, a white space, w and another whitespace to sa
      5. Overwrite w to null (“”)
   6. Print k as the number of words beginning and ending with vowels
6. Else print “Invalid Input”
7. Print sa as the arranged sentence
8. End

**Source Code:**

import java.util.\*;

class Sentence\_vowel

{

void Sent\_Vow(String s)

{

String sa="",w="";int i,l,k=0;

if(s.endsWith(".")|| s.endsWith("?")||s.endsWith("!"))

{

s=s+" ";

s=s.toUpperCase();

s=s.substring(0,(s.indexOf(".")));

l=s.length();

for(i=0;i<l;i++)

{

char ch=s.charAt(i);

if(ch!=' ')

w=w+ch;

else

{

if((w.startsWith("A")||w.startsWith("E")||w.startsWith("I")||w.startsWith("O")||w.startsWith("U"))&&(w.endsWith("A")||w.endsWith("E")||w.endsWith("I")||w.endsWith("O")||w.endsWith("U")))

{

k++;

sa=sa+" "+w;

w="";

}

else

{

sa=sa+" "+w+" ";

w="";

}

}

}

System.out.println("The no. of words which start and end with a vowel are "+ k);

}

else

System.out.println("Invalid input");

System.out.println(sa);

}

public static void main(String ars[])

{

Scanner in=new Scanner(System.in);

System.out.println("Enter a Sentence or terminating with '.' or '!' ");

String s;

s=in.nextLine();

Sentence\_vowel ob=new Sentence\_vowel();

ob.Sent\_Vow(s);

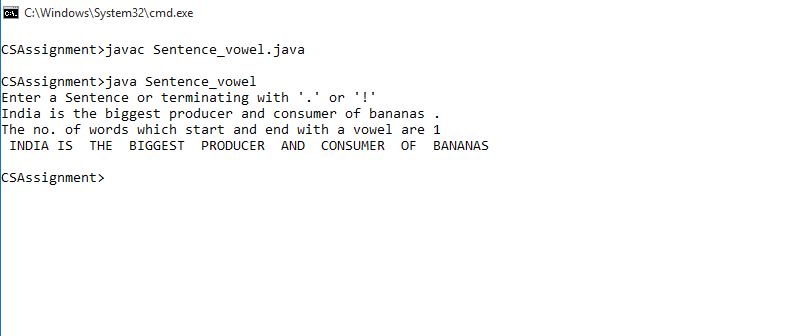
}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data type | Function |
| s | String | Accept sentence from User |
| sa | String | Stores the Arranged sentence |
| w | String | Hold each word of s |
| i | int | Loop variable |
| l | int | Stores the length of s |
| k | int | Store the number of words beginning and ending with vowels |
| ch | char | Stores each character of s |

**Input / Output Screen**



**Program 15:** Write a program to declare a square matrix of size M X M (M must be greater than 2 and less than 7). Rotate the matrix 90° anti clockwise. Display both the original and converted matrix.

**ALGORITHM:**

1. Declare variables len ,n, n2, i, j, N
2. Read the size of matrix from the User in N
3. Declare a 2-D array mat[][] of size N×N
4. Read mat from User
5. Print mat in matrix form
6. Store the length of mat in len
7. Run a loop from i=0 till i<len
   1. Run another loop from j=0till j<i
      1. Initialise n to mat[i][j]
      2. Overwrite mat[i][j] to mat[j][i]
      3. Overwrite m[j][i] to n
8. Run a loop from i=0 till i<(len/2)
   1. Run a loop from j=0 till j<len
      1. Initialise n2 to mat[i][j]
      2. Overwrite mat[i][j] to mat[len-i-1][j]
      3. Overwrite mat[len-i-1][j] to n2
9. Print the rotated matrix
10. End

**Source Code:**

import java.util.Scanner;

class Rotate

{

public static void rotate(int mat[][]) {

int len = mat.length;

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

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

int n = mat[i][j];

mat[i][j] = mat[j][i];

mat[j][i] = n;

}

}

for (int i = 0; i < len / 2; i++) {

for (int j = 0; j < len; j++) {

int n2 = mat[i][j];

mat[i][j] = mat[len - i - 1][j];

mat[len - i - 1][j] = n2;

}

}

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

for (int j = 0; j < len; i++) {

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

}

System.out.println();

}

}

public static void main(String ars[]) {

System.out.println("Enter the size of the square matrix");

Scanner in = new Scanner(System.in);

int N=in.nextInt();

int mat[][] = new int[N][N];

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

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

for (int j = 0; j < N; ++j) {

mat[i][j] = in.nextInt();

}

}

System.out.println("The entered matrix is");

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

for (int j = 0; j< N; j++) {

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

}

System.out.println();

}

System.out.println("The rotated matrix");

rotate(mat);

}

}

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Function |
| i, j | int | Loop variable |
| n | int | Temporary variable for swapping |
| n2 | int | Temporary variable for swapping |
| mat | int | Store the matrix entered by the User |
| N | int | The size of the square matrix |

**Input / Output Screen**



**BIBLIOGRAPHY**

I have done this assignment with the help of the book:

Name: **COMPUTER SCIENCE with JAVA**

Author: **SUMITA ARORA**

Publication: **DHANPAT RAI & CO. (Pvt.) Ltd.**

---X---