1. Create a program that reads the length and width of a farmer’s field from the user in feet.

Display the area of the field in acres. Hint: There are 43,560 square feet in an acre.

Sample Input 1000 1000

Sample Output 22.95684113865932 acres

**Solution:**

import java.util.Scanner;

class Kaviya

{

public static void main(String[] args)

{

Scanner a=new Scanner(System.in);

int len=a.nextInt();

int wid=a.nextInt();

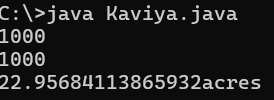
double area=len\*wid;

double acre=area/43560;

System.out.println(acre+"acres");

}

};

**Output:**

2. An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each

gizmo weighs 112 grams. Write a program that reads the number of widgets and the number of

gizmos from the user. Then your program should compute and display the total weight of the

parts.

Sample Input: 10 20

Sample Output: The total weight of all these widgets and gizmos is 2990 grams.

**Solution:**

import java.util.Scanner;

class Product

{

public static void main(String[] args)

{

Scanner a=new Scanner(System.in);

int w=a.nextInt();

int g=a.nextInt();

int ans=w\*75;

int ans1=g\*112;

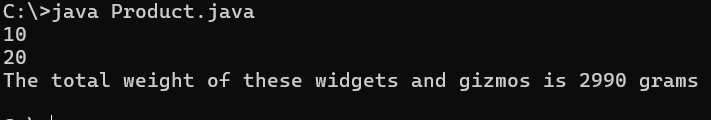
int answer=ans+ans1;

System.out.println("The total weight of these widgets and gizmos is "+ answer +" grams");

}

};

**Output:**



3. In many jurisdictions, a small deposit is added to drink containers to encourage people to recycle

them. In one particular jurisdiction, drink containers holding one liter or less have a $0.10

deposit and drink containers holding more than one liter have a $0.25 deposit. Write a program

that reads the number of containers of each size(less and more) from the user. Your program

should continue by computing and displaying the refund that will be received for returning

those containers. Format the output so that it includes a dollar sign and always displays exactly

two decimal places.

Sample Input 10 20

Sample Output Your total refund will be $6.00.

**Solution:**

import java.util.Scanner;

class Deposit

{

public static void main(String[] args)

{

Scanner a=new Scanner(System.in);

int b=a.nextInt();

int c=a.nextInt();

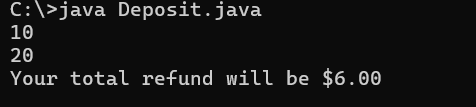
double refund=(b\*0.10)+(c\*0.25);

System.out.printf("Your total refund will be $%.2f%n",refund);

}

};

**Output:**



4. Write a program to find whether the given input number is Odd. If the given number is odd,

the program should return 2 else It should return 1.

**Solution:**

import java.util.Scanner;

class Odd

{

public static void main(String[] args)

{

Scanner o=new Scanner(System.in);

int a=o.nextInt();

if(a%2==0)

{

System.out.println("1");

}

else

{

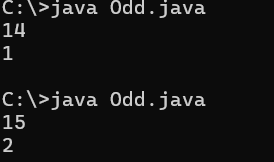
System.out.println("2");

}

}

};

**Output:**



5. Write a program that returns the last digit of the given number. Last digit is being referred to

the least significant digit i.e. the digit in the ones (units) place in the given number. The last

digit should be returned as a positive number.

For example,

if the given number is 197, the last digit is 7. if the given number is -197, the last digit is 7.

**Solution:**

import java.util.Scanner;

class Odd

{

public static void main(String[] args)

{

Scanner o=new Scanner(System.in);

int a=o.nextInt();

if(a<0)

{

System.out.println((-1\*a)%10);

}

else

{

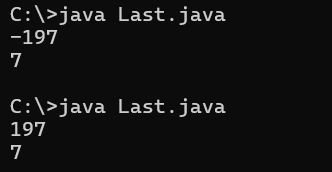
System.out.println(a%10);

}

}

};

**Output:**



6. Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the slim of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

if the input numbers are -267 and -154, the sum of last two digits should be 11.

**Solution:**

import java.util.Scanner;

class Lastnumber

{

public static void main(String[] args)

{

Scanner o=new Scanner(System.in);

int a=o.nextInt();

int b=o.nextInt();

int c,d;

if(a>0 && b>0)

{

c=a%10;

d=b%10;

System.out.println(c+d);

}

else if(a<0 && b>0)

{

c=(-1\*a)%10;

d=b%10;

System.out.println(c+d);

}

else if(a>0 && b<0)

{

c=a%10;

d=(-1\*b)%10;

System.out.println(c+d);

}

else if(a<0 && b<0)

{

c=(-1\*a)%10;

d=(-1\*b)%10;

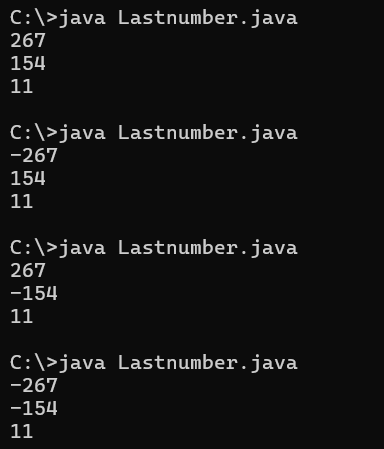
System.out.println(c+d);

}

}

};

**Output:**

****

7. Complete the program to convert days into years, month and days. (Ignoring leap year and

considering 1 month is 30 days)

Input 375 Output YEARS: 1 MONTH: 0 DAYS: 10

Input 200 Output YEARS: 0 MONTH: 6 DAYS: 20

**Solution:**

import java.util.Scanner;

class Convert

{

public static void main(String[] args)

{

Scanner o=new Scanner(System.in);

int n=o.nextInt();

int years,months,days;

years=n/365;

months=n%365/30;

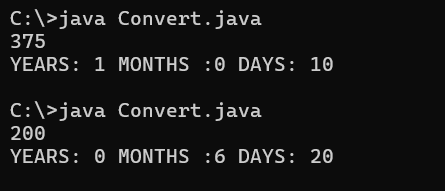
days=n%365%30;

System.out.println("YEARS: " + years + " MONTHS :"+months + " DAYS: " + days);

}

};

**Output:**



8. Write a program that returns the second last digit of the given number. Second last digit is

being referred 10the digit in the tens place in the given number. For example, if the given

number is 197, the second last digit is 9. Note1 - The second last digit should be returned as a

positive number. i.e. if the given number is -197, the second last digit is 9. Note2 - If the given

number is a single digit number, then the second last digit does not exist. In such cases, the

program should return -1. i.e. if the given number is 5, the second last digit should be returned

as 0.

**Solution:**

import java.util.Scanner;

class Tenthplace

{

public static void main(String[] args)

{

Scanner o=new Scanner(System.in);

int n=o.nextInt();

int b,c,d;

if (n>=0 && n<=9)

{

System.out.println("-1");

}

else

{

if(n<0)

{

b=-1\*n;

c=b/10;

d=c%10;

System.out.println(d);

}

if(n>0)

{

c=n/10;

d=c%10;

System.out.println(d);

}

}

}

};

**Output:**

