Task1a) For 5 inputs, find Maximum.(flowchart+javacode)

Task1b) For 5 inputs, find minimum. (flowchart+javacode)

Task1c) For 5 inputs, find average. (flowchart+javacode)

Task2 ) combine flowcharts &javacodes of finding maximum & minimum from Task1 into 1 flowchart & 1 javacode. Hint: write common steps from both flowchart only once. If any line or any condition (along with yes/no part) is different, then write the different parts one after another.

Task3 ) combine flowcharts &javacodes of Task2 with finding average into 1 flowchart & 1 javacode

Task4) Draw flowchart & Write a Java program which adds all numbers that are multiples of both 7 & 9 up to 600.

Task5) Draw flowchart & Write a Java program which adds all numbers that are multiples of either 7 or 9 up to 600. Ensure that numbers like 63 are added only once in the sum.

Task6) Draw flowchart & Write Java program which adds all numbers that are multiples of either 7 or 9 but not both, up to 600.

Task7) Draw flowchart & Write a Java program which takes a number & tells how many digits are in that number. Example: if user gives 9876, you should print 4.

Hint: keep dividing by ten & count how many times the could be divided.

9876 by 10, is 987, count that got 1 digit

987 by 10, is 98, count that got 1 digit (total 2)

98 by 10, is 9, count that got 1 digit (total 3)

9 by 10, is 0, count that got 1 digit (total 4)

done!

Task8) Draw flowchart & Write a Java program which takes a number & prints the value of 10 to the power that number.You need to use loop because variable in the power isn't allowed.For example: if user gives 3, print 1000.

Hint: Keep multiplying 1 by 10, again & again, 3 (or n) times like sum=sum+n, you need to write,

product = product x 10

1 x 10 = 10

10 x 10 = 100

100 x 10 = 1000

1000 x 10 = 10000

Task9) Draw flowchart & Write a Java program which takes a number & prints the digits from unit place, then tenth, then hundredth, etc.Example: if user gives 32768, then print 8, 6, 7, 2, 3Hint: Taking remainder/modulus of division by 10. After printing the remainder, drop the last digit by dividing by 10. Then start over.

32,768 % 10 = 8

32,768 / 10 = 3, 276

3, 276 % 10 = 6

3, 276 / 10 = 327

327 % 10 = 7

327 / 10 = 32

32 % 10 = 2

32 / 10 = 3

3 % 10 = 3

3 / 10 = 0

Task10) Draw flowchart & Write a Java program which takes a number & prints the digits from left to right.

Example: if user gives 32768, then print 3, 2, 7, 6, 8 Hint: count how many digits calculate 10 to the power that (number of digits) minus 1. Say, 32768 has 5 digits, so you calculate 10 to the power 4 which is 10,000.

Then divide 32,768 by 10,000 & thus you get 3. take remainder of 32,768 by 10,000 & thus you get 2,768

Then divide 10,000 by 10 to get 1,000 Then divide 2,768 by 1,000 & thus you get 2. take remainder of 2,768 by 1,000 & thus you get 768. keep going on until there is no more digits left (zero!). In short:

Loop 1: First count digits, say 5 in this case for 32,768

Loop 2: Then calculate 10 to the power 4 (5-1), that is 10,000.

Loop 3: Then repeat following three steps

32,768 / 10,000 = 3

32,768 % 10,000 = 2,768

10,000/10 = 1,000

2,768 / 1,000 = 2

2,768 % 1,000 = 768

1,000/10 = 100

768 / 100 = 7

768 % 100 = 68

100/10 = 10

68 / 10 = 6

68 % 10 = 8

10/10 = 1

8 / 1 = 8

8 % 1 = 0

1/10 = 0

Task11) Telling words: Draw flowchart & write javacode of a program that takes a number between 0& 9. You have to print that number in words.

Hint: if n == 1, then print "one"

else if n == 2, then print "two"

Task12) Combine Task 10 & 11 into a single flowchart & java program so that it can tell any number in words.

Example: If user gives 932, print nine three two.

Task13) Draw flowchart & write java code of a program that takes a number & prints all numbers up to that number. If the user gives 8, print 1 to 8.

Task14) Draw flowchart & write java code of a program that takes a number & counts how many times that number can be divided by all numbers up to that number (Those numbers are also known as factors)

If the user gives 8, tries to divide 8 by each of 1 to 8 & count how many times it could be divided.

For example: If user enters 8,

try to divide 8 by 1, its divisible (increase count to 1)

try to divide 8 by 2, its divisible (increase count to 2)

try to divide 8 by 3, its NOT divisible

try to divide 8 by 4, its divisible (increase count to 3)

try to divide 8 by 5, its NOT divisible

try to divide 8 by 6, its NOT divisible

try to divide 8 by 7, its NOT divisible

try to divide 8 by 8, its divisible (increase count to 4)

Now print the count which is 4 in this case.

Task15) If a number is NOT divisible any number other than 1and itself, then it is called prime number.

For example, 13 is a prime number because it is NOTdivisible by any number other than 1 & 13 (itself).

Take one number from the user & tell if it is prime number or not.Hint: Use the technique from Task14 & count factors of the input. Factors are those numbers between 1& n that can divide the number, n. If there are more than two factors (1 and n), then the number, n is not prime because it was divisible by other numbers.

Task16) Modify Task15, instead of counting factors, print sum of factors.

Task17) Modify Task15, calculate sum of factors less than the number itself. If the sum equals to the number, then

print that the number is a perfect number.Example: User enters n = 6. Factors of 6 are 1, 2, 3.

Sum of those factors 1+2+3=6 which is same as the number 6(n).

So, print that 6 is a perfect number.

Example2: If user enters 8. Factors of 8 are 1, 2, 4.

1+2+4=7 (NOT equal to 8).

So, print that 8 is NOT a perfect number.

Task18) Ask user for a range. Count how many numbers are prime and how many numbers are perfect numbers between that range.For example, between 2& 6 there are 3 prime numbers (2, 3, 5) and 1 perfect number (6).

Sample Input:

2 6

Sample Output:

Between 2 & 6,

Found 3 prime numbers

Found 1 perfect number.

Task19) Assume that we have the following array contains marks of 5 students.

int[] marks = new int[] {10, 30, 20, 50, 40};Find the max, min, average mark.

Output:

Highest mark is 50

Lowest mark is 10

Average mark is 30

Task20) Assume that we have the following array contains marks.int[] marks = new int[] {10, 30, 20, 50, 40};

Find how many students are better than average.Output:

2 students are better than average.

They received following marks

50

40

Hint: First calculate average marks. Then loop through the marksarray & count how many marks are greater than average.Print this count. Then again, loop through the marks & counthow many elements were greater than average marks.

Task21) Assume that we have the following array.int[] a = new int[] {10, 30, 20, 50, 40};

Find the largest number & print it along with its location.Output:

Largest number is 50

Largest number was found at location 3

Task22) Assume that we have the following array.int[] a = new int[] {10, 30, 20, 50, 40};

Modify Task21. Swap the first number with largest number.

Your output should be 50, 30, 20, 10, 40. Hint:

Find largestNumber&largestNumberLocation.

backup = a[0];

a[0]=largestNumber;

a[largestNumberLocation]=backup;

Task23) Assume that we have the following array.int[] a = new int[] {50, 30, 20, 10, 40};

Modify Task22.

Find largest number among all numbers between 2nd& last number (30, 20, 10, 40).

Swap the 2nd number with this largest number.

Thus you just put 2nd largest in 2nd position.

Task24) Combine ideas from Task 22 & 23.

Ask the user for a number n, & then find nth highest number

Hint:

Repeat the idea (from task 22,23) n times.

Then you will find nth highest number in nth position in array.

Task 25) Modify Task 24 to sort / arrange all numbers in the array.

Your output should be 50, 40, 30, 20, 10

Task 26) Find median among some numbers.

Ask user how many numbers & input those numbers.

Definition of Median:

For example, if a < b < c, then the median of the list {a, b, c} is b,

and, if a < b < c < d, then the median

of the list {a, b, c, d} is the mean of b & c;

i.e., it is (b + c)/2.

Hint:

Sort numbers

if there are odd number of numbers, say 9, then

the median is the middle number, at 5th position

if there are even number of numbers, say 10, then

the median is average of two middle numbers, (5th+6th)/2

Sample Input 1:

5

10 50 40 20 30

Sample Output 1:

30

Sample Input 2:

6

10 50 40 20 30 60

Sample Output 2:

35