Basic Java Questions

1)Print “Welcome in Java” without using semicolons?

What will be the Output of the below code:

2)public class A {

```
public static void main(String[] args) {
byte x=125;
x=x+5;
System.out.println(x);
} }
```

3)public class A {

```
public static void main(String[] args) {
byte x=125;
x=byte(x+5);
System.out.println(x);
} }
```

4) public class A {

```
public static void main(String[] args) {
int a=40;
int b=024;
                int c=0x14;
System.out.println(“a : ”+a+”\t b : ”+b+”\t c”+c);
}
}
```

5)public class A {

```
public static void main(String[] args) {
char y=65536
System.out.println(y);
}
}
```

6) public class A {

```
public static void main(String[] args) {
System.out.println(65==true);
}
}
```

7)public class A {

```
public static void main(String[] args) {
System.out.println(-7%2);
                System.out.println(7%-2);
                System.out.println(-7%-2);
                System.out.println(7/0);
                System.out.println(7.5/0);
                System.out.println(-7.5/0);
                System.out.println(7%0);
                System.out.println(7.5%0);
                System.out.println(0%10);
}
}
```

8) public class A {

```
public static void main(String[] args) {
int x=5,y;
                y= (--x) + (x++) * (++x) / (x--);
System.out.println("x : "+x+"\t y : "+y);
}
}
```

9) public class A {

```
public static void main(String[] args) {
int a=10,b=20,c=30,d=40;
                boolean res= (a<b) || (++c > d);
System.out.println("a : "+a+"\n b : "+b+"\n c : "+c+"\n d : "+d+"\n
result : "+res);
} }
```

10) public class A {

```
public static void main(String[] args) {
long x=1L;
switch(x){
        case 1: System.out.println("matched");
        }
}
}
```

11) public class A {

```
public static void main(String[] args) {
float x=1.1f;
switch(x){
        case 1: System.out.println("matched");
        }
} }
```

12) public class A {

```
public static void main(String[] args) {
int a=10,b=20;
b=(a+b)-(a=b);
        System.out.println("a : "+a+" \n b : "+b);

} }
```

13)Swap without using third variable and without using + , - ,*,/ operator

14) Write a program to count total number of notes in given amount.

15) Write a program for an input year N ,find whether the year is a leap or not.

16) Find the cost of tiling a dining room 20 m long and 15 m wide at the rate of $ 5 per square m.

17) A rectangular garden has dimensions of 30 m by 20 m and is divided in to 4 parts by two pathways that run perpendicular from its sides. One pathway has a width of 3 m and the other, 4 m. Calculate the total usable area of the garden.

18) Find the 28th term and sum of an Arithmetic Progression -21 -18 -15 -12 . . . . .

19)Input format : Input will contain four integer a,b,c,d one each line

Output format : print the greatest of the four integers

Sample Input :

3

4

5

6

Sample Output :

6

Loop

---

1)WAP to print a statement 1000 number of times.
2)WAP to print N natural number.
3)WAP to find out the sum of N natural number.
4)WAP to print table of a number.
5)WAP to find out the factorial of a number.
6)WAP to find out the factors of a number.
7)WAP to check whether entered number is prime or not.
8)WAP to print Fibonacci series.
9)WAP to print N even numbers.
10)WAP to print Even numbers upto N.
11)WAP to print N odd numbers.
12)WAP to print Odd numbers upto N.
13)WAP to print N natural numbers in reverse order
14)WAP to print alphabets in uppercase
15)WAP to print alphabets in lowercase
16)..... -6 -3 0 3 6 9 ..... n terms
17)1 2 4 7 11 16 ..... n terms
18)1 2 2 4 8 32 ..... n terms

19)1 + 1/2 + 1/3 + 1/4 + 1/5 ….. n terms(find out sum)
20)0 7 14 21 28 35 …..
21)1 4 9 16 25 …..
22)1 8 27 64 125 …..
23)1 9 25 49 81 …..
24)0 4 16 36 64 ….. 1 2 3 4 5 6 7 8
25)1 27 125 343 ……
26)0 8 64 216 ……
27)* # * # * # * # * …….
28)1 2 3 4 Hello 6 7 8 9 Hello 11 12 ….
29)1 11 111 1111 11111 ……
30)1+11+111+1111+11111+….
31)9 99 999 9999 99999 …….
32)A b C d E f G h …… n terms
33) 2,5,10,17,26,37…n terms
34)WAP to print Alphabets in reversing order.
35)WAP to check whether entered number is perfect or not
36)WAP to count number of digits
37)WAP to reverse a number
38)WAP to check whether entered number is palindrome or not
39)WAP to check whether entered number is Armstrong or not
40)WAP to check whether entered number is strong or not
41)WAP to count no. Of even and odd digits in a number
42)WAP to find out LCM of a number
43)WAP to find out HCF of a number
44)WAP to convert binary number into decimal number
45)WAP to interchange first and last digit of a number
46)WAP to find out the sum of all the digits of a number
47)WAP to find out the sum of first and last digit of a user entered number
48)WAP to print tables of all the numbers between two entered numbers
49)WAP to find out the factors of all the numbers between two entered numbers
50)WAP to find out all the perfect numbers between two entered numbers

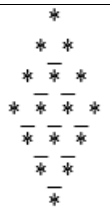
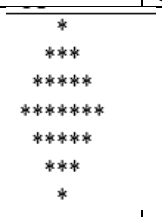
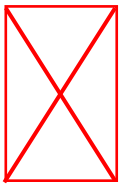
51)WAP to find out all the palindrome numbers between two entered numbers
52)WAP to reverse all the numbers between two entered number
53)WAP to find out all the Armstrong numbers between two entered numbers
54)WAP to print all the strong numbers between two entered numbers
55)WAP to print all the even numbers between two entered numbers
56)WAP to print all the odd numbers between two entered numbers
57)WAP to print factorial of all the numbers between two entered numbers
58)WAP to print all the prime numbers between two entered numbers
59)WAP to convert decimal number into binary number without using array
60)WAP to find out the sum of all integer between 100 and 200 which are divisible by 9
61)WAP to print Square, Cube and Square Root of all numbers from 1 to N
62)WAP to find out all the leap years between two entered years
63) Print series x^1 - x^2 + x ^3- x ^4.....
64) Print series x^1/1 + x^2 /2+ x ^3/3 + x ^4/4.....
65) Print series 1/x^1 +2/ x^2 + 3/x ^3 +4/ x ^4.....
66) Print series x^1/2+ x^2/3 + x ^3/4+ x ^4/5.....
67) Print series 2/x^1 +3/ x^2 + 4/x ^3+5/ x ^4.....
68) Print series x^1/1 + x^2/3 + x ^3/5 +x ^4/7 .....
69) Print series 1/x^1 + 3/x^2 + 5/x ^3 + 7/x ^4.....
70) Print series 1/x^1 + 3/x^2 + 5/x ^3 + 7/x ^4.....
71) Print series 1!/x^1 + 2!/x^2 + 3!/x ^3 + 4!/x ^4.....
72) Print series x^1/1! + x^2/2! + x ^3/3!+ x ^4/4!.....
73) WAP to calculate x^y (x to the power y)
74) WAP to calculate the sum of given series x^1 + x^2 + x ^3.....x^n
75) WAP to calculate the sum of series 1/1! + 2/2! + 3/3!.....n\n!
76) WAP in java to display the n terms of harmonic series and their sum 1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms
77) Calculate the sum of following series 1/x + 1/x^2 + 1/x^3.....1/x^n

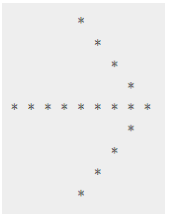
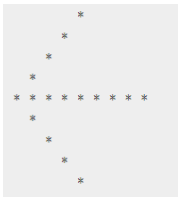
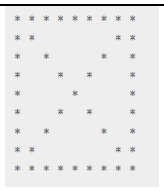
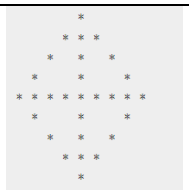
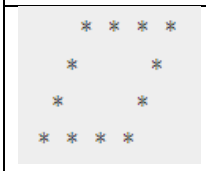
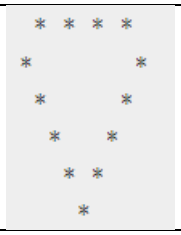
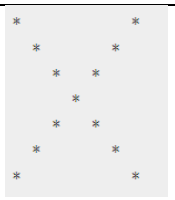
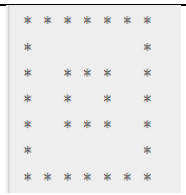
Pattern Questions

1	2	3	4	5	6	7

*****
 	
	*
	*
	*
	*
	*
		*
		 *
		  *
		  *
		  *
			*****
			*****
			*****
			*****
			*****
				12345
				12345
				12345
				12345
				12345
					11111
					22222
					33333
					44444
					55555
						1
						00
						111
						0000
						11111

8	9	10	11	12	13	14
* ** *** **** *****	1 12 123 1234 12345	1 22 333 4444 55555	A AB ABC ABCD ABCDE	a ab abc abcd abcde	1 01 101 0101 10101	1 23 456 78910
15	16	17	18	19	20	21
A BB CCC DDDD EEEEE	a bc def ghij klmno	* ## *** #### *****	1 10 101 1010 10101	* * * * * * * * * * * *	1 12 1 3 1 4 12345	1 22 3 3 4 4 55555
22	23	24	25	26	27	28
A AB A C A D ABCDE	a bc d f g j klmno	* ** *@* *@@* * * * * *	5 54 543 5432 54321	* *# *#* *#*# *#*#*	1 10 1 1 1 0 10101	1 123 12345 1234567 123456789
29	30	31	32	33	34	35
1 222 33333 4444444 555555555	***** **** *** ** *	12345 1234 123 12 1	55555 4444 333 22 1	ABCDE ABCD ABC AB A	EEEEE DDDD CCC BB A	***** * * * * ** *
36	37	38	39	40	41	42
ABCDE A D A C AB A	***** #### *** ## *	55555 4 4 3 3 22 1	123456 54321 1234 321 12 1	* ** **** ******* **********	A BCD EFGHI JKLMNOP	54321 5432 543 54 5
43	44	45	46	47	48	49
1 12 123 1234 12345	1 22 333 4444 55555	5 44 333 2222 11111	A AB ABC ABCD ABCDE	1 11 1*1 1**1 11111	A AB A_C A__D ABCDE	1 10 101 1010 10101
50	51	52	53	54	55	56
12345 1234 123 12 1	55555 4444 333 22 1	12345 1__4 1_3 12 1	55555 4__4 3_3 22 1	ABCDE A__D A_C AB A	ABCDE ABCD ABC AB A	11111 2222 333 44 5
57	58	59	60	61	62	63
* * * * * * * * * * * * * * *	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	A A B A B C A B C D A B C D E	X X X X__X X____X X X X X X	* *** ***** ******* *********	1 123 12345 1234567 123456789	A ABC ABCDE ABCDEEF ABCDEFGHI
64	65	66	67	68	69	70
* *_*	1 1 1	1 1*1	A B B	# *#*	********* *******	* * * * * * * * *

__ *____* *********	1 2 1 1 3 3 1 1 4 6 4 1	1***1 1*****1 111111111	C C D D EEEEEEEEE	**#** ***#*** ****#****	***** *** *	* * * * * *
71	72	73	74	75	76	77
123456789 1234567 12345 123 1	A B C D E A B C D A B C A B A	5 5 5 5 5 4 4 4 4 3 3 3 2 2 1	123456789 1 7 1 5 1 3 1	123456789 1+++++7 1+++5 1+3 1	x xx xxx xxxx xxx xx x	1 12 123 1234 123 12 1
78	79	80	81	82	83	84
1 12 123 1234 123 12 1	1 1 2 1 3 1 4 1 3 1 2 1					1 212 32123 4321234 543212345
85	86	87	88	89	90	91
* * ** ** *** *** **** **** ***** *****	******** *** *** ** ** * *	***** **** *** ** * ** *** **** *****	***** **** *** ** * ** *** **** *****	********** **** **** *** *** ** ** * * * * ** ** *** *** **** **** **********	1 2 3 4 1 2 3 4 5 4 3 2 1 4 3 2 1	##### #### ### ## # ## ### #### #####
92	93	94	95	96	97	98
1 101 10101 1010101 101010101 10101010101	1 333 333 55555 55555 7777777 7777777	ABCDE A E A E A E ABCDE	AAAAA B B C C D D EEEEE	abcde a e a e a e abcde	aaaaa b b c c d d eeeee	**** * * * * ****
99	100	101	102	103	104	105
1 2 11 3 10 12 4 9 13 18 5 8 14 17 19 6 7 15 16 20 21	123 456 789	1 2 3 4 14 15 16 5 13 20 17 6 12 19 18 7 11 10 9 8	J Ja Jav Java	1 2 3 4 5 6 7 8 9 10	1 2 3 4 4 3 2 1 1 2 3 * * 3 2 1 1 2 * * * * 2 1 1 * * * * * * 1	1 2 6 3 7 10 4 8 11 13 5 9 12 14 15
106	107	108	109	110	111	112
_ _ _ _ 1 _ _ _ 3 2 _ _ 5 4 3 _ 7 6 5 4 9 8 7 6 5	I IN IND INDI INDIA INDIA INDI IND IN I	1 3 5 7 9 11 13 15 17 19	1 AB 123 ABCD 12345	Print any random number pyramid as: 4572 572 72 2	1 23 4 56 7 89 10	*000000 0*00000 00*0000 000*000 0000*00 00000*0 000000*

113	114	115	116	117	118	119
Print equilateral triangle number as: 5 454 34543 2345432 123454321	A 1 BB 22 CCC 333	Print continue character number pyramid as: 1 A B 2 3 4 C D E F 5 6 7 8 9	Even-odd number star pyramid as: 1 *2 1*3 *2*4 1*3*5	Write a program to print the following number design/triangle: 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3	1 a 21 ba 321 cba 4321 dcba 54321 edcba	Print positive-negative number triangle as: 9 8 6 7 5 3 4 2 0 -2 1 -1 -3 -5 -7
120	121	122	123	124	125	126
Print the continuous vertical number pyramid as: 1 2 7 3 8 13 4 9 14 19 5 10 15 20 25	Square number pyramid program: 1 1 4 9 1 4 9 16 25 1 4 9 16 25 36 49	Print continue character number pyramid as: 1 A B 2 3 4 C D E F 5 6 7 8 9				
127	128	129	130	131		
						

Q132)Print pattern of numbers from 1 to n as shown bleow.Each of the numbers is separated by a single space.

Input format : The input will contain a single integer n

Output format : Print pattern

Sample Input

3

Sample Output

3 3 3 3 3

3 2 2 2 3

3 2 1 2 3

3 2 2 2 3

3 3 3 3 3

Q133) Given a non-negative integer **N**. The task is to check if N is a power of **2**. More formally, check if **N** can be expressed as $\mathbf{2^x}$ for some integer **x**. Return **true** if N is power of 2 else return **false**.

Example 1:

Input:
N = 8
Output:
YES
Explanation:
8 is equal to 2 raised to 3 ($2^3 = 8$).

Example 2:

Input:
N = 98
Output:
NO
Explanation:
98 cannot be obtained by any power of 2.

Q134) Given two integers n and r, find ${}^nC_r$. Since the answer may be very large, calculate the answer modulo 10^9+7.

Example 1:

Input: n = 3, r = 2
Output: 3
Explaination: ${}^3C_2 = 3$.

Example 2:

Input: n = 2, r = 4
Output: 0
Explaination: r is greater than n.

Q135) Given a number and its reverse. Find that number raised to the power of its own reverse.
Note: As answers can be very large, print the result modulo $10^9 + 7$.

Example 1:

Input:
N = 2, R = 2
Output: 4
Explanation: The reverse of 2 is 2 and after raising power of 2 by 2 we get 4 which gives remainder as 4 when divided by 1000000007.
Example 2:

Input:
N = 12, R = 21
Output: 864354781
Explanation: The reverse of 12 is 21and 12^{21} when divided by 1000000007 gives remainder as 864354781.

Q136)Lucky numbers are subset of integers. Rather than going into much theory, let us see the process of arriving at lucky numbers,

Take the set of integers

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,……

First, delete every second number, we get following reduced set.

1, 3, 5, 7, 9, 11, 13, 15, 17, 19,…….

Now, delete every third number, we get

1, 3, 7, 9, 13, 15, 19,……..

Continue this process indefinitely……

Any number that does NOT get deleted due to above process is called “lucky”.

You are given a number N, you need to tell whether the number is lucky or not. If the number is lucky return 1 otherwise 0.**Example 1:**

Input:

N = 5
Output: 0
Explanation: 5 is not a lucky number as it gets deleted in the second iteration.

Example 2:

Input:
N = 19
Output: 1
Explanation: 19 is a lucky number because it does not get deleted throughout the process.

Q137) Given a number **N**, the task is to find the largest prime factor of that number.

Example 1:

Input:
N = 5
Output:
5
Explanation:
5 has 1 prime factor i.e 5 only.

Example 2:

Input:
N = 24
Output:
3
Explanation:
24 has 2 prime factors 2 and 3 in which 3 is greater.

Q138) Given a number N.Find the minimum number of operations required to reach **N** starting from **0**. You have 2 operations available:

- Double the number
- Add one to the number

Example 1:

Input:
N = 8
Output: 4
Explanation:
0 + 1 = 1 --> 1 + 1 = 2 --> 2 * 2 = 4 --> 4 * 2 = 8.
Example 2:

Input:
N = 7
Output: 5
Explanation:
0 + 1 = 1 --> 1 + 1 = 2 --> 1 + 2 = 3 --> 3 * 2 = 6 --> 6 + 1 = 7.

Q139) Given two integers **N** and **M**. The problem is to find the number closest to **N** and divisible by **M**. If there are more than one such number, then output the one having **maximum absolute value**.

Example 1:

Input:
N = 13 , **M =** 4
Output:
12
Explanation:
12 is the Closest Number to
13 which is divisible by 4.

Q140) Given the first 2 terms A1 and A2 of an Arithmetic Series.Find the N^{th} term of the series.

Example 1:

Input:
A1=2
A2=3
N=4
Output:
5
Explanation:

The series is 2,3,4,5,6....
Thus,4th term is 5.

Q141) The task is to find the smallest number with given sum of digits as **S** and number of digits as **D.**

Example 1:

Input:
S = 9
D = 2
Output:
18
Explanation:
18 is the smallest number
possible with sum = 9
and total digits = 2.

Example 2:

Input:
S = 20
D = 3
Output:
299
Explanation:
299 is the smallest number
possible with sum = 20
and total digits = 3.

Q142) Write a program to calculate ${}_nP_r$. **${}_Np^r$** represents n permutation r and value of ${}_nP_r$ is (n!) / (n-r)!.

Example 1:

Input: n = 2, r = 1
Output: 2
Explaination: 2!/(2-1)! = 2!/1! = (2*1)/1 = 2.

Example 2:

Input: n = 3, r = 3
Output: 6
Explaination: 3!/(3-3)! = 3!/0! = 6/1 = 6.

Q143) Given a number **N**. Your task is to check whether it is fascinating or not.
Fascinating Number: When a number(should contain 3 digits or more) is multiplied by 2 and 3, and when both these products are concatenated with the original number, then it results in all digits from 1 to 9 present exactly once.

Example 1:

Input:
N = 192
Output: Fascinating
Explanation: After multiplication with 2
and 3, and concatenating with original
number, number will become 192384576
which contains all digits from 1 to 9.
Example 2:

Input:
N = 853
Output: Not Fascinating
Explanation: It's not a fascinating
number.

Array

Q.1 Find a peak element which is not smaller than its neighbours ?

An element is called a peak element if its value is not smaller than the value of its adjacent elements(if they exists). Given an array **arr[]** of size **N**, **Return the index of any one of its peak elements**.

Note: The generated output will always be 1 if the index that you return is correct. Otherwise output will be 0. Also, **arr[]** will be in ascending order before the peak element, and in descending order after that.

Example 1:

Input:

N = 3

arr[] = {1,2,3}

Possible Answer: 2

Generated Output: 1

Explanation: index 2 is 3.

It is the peak element as it is

greater than its neighbour 2.

If 2 is returned then the generated output will be 1 else 0.

Example 2:

Input:

N = 3

arr[] = {3,4,2}

Possible Answer: 1

Output: 1

Explanation: 4 (at index 1) is the

peak element as it is greater than

it's neighbor elements 3 and 2.

If 1 is returned then the generated output will be 1 else 0.

Your Task:

You don't have to read input or print anything. Complete the

function **peakElement**() which takes the array arr[] and its size N as input parameters and return the index of any one of its peak elements.

Q.2 Find the minimum and maximum element in an array ?

Given an array **A** of size **N** of integers. Your task is to find the **minimum and maximum** elements in the array.

Example 1:

Input:

N = 6

A[] = {3, 2, 1, 56, 10000, 167}

Output: 1 10000
Explanation: minimum and maximum elements of array are 1 and 10000.

Example 2:

Input:

N = 5

A[] = {1, 345, 234, 21, 56789}

Output: 1 56789
Explanation: minimum and maximum element of array are 1 and 56789.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **getMinMax()** which takes the array **A[]** and its size **N** as inputs and returns the **minimum and maximum** element of the array

Q.3 Write a program to reverse the array ?

You are given a string s. You need to reverse the string.

Example 1:

Input:

s = Geeks

Output: skeeG

Example 2:

Input:

s = for

Output: rof

Your Task:

You only need to complete the function **reverseWord()** that takes s as parameter and returns the reversed string

Q.4 Write a program to sort the given array

Given a random set of numbers, Print them in sorted order.

Example 1:

Input:

N = 4

arr[] = {1, 5, 3, 2}

Output: {1, 2, 3, 5}

Explanation: After sorting array will

be like {1, 2, 3, 5}.

Example 2:

Input:

N = 2

arr[] = {3, 1}

Output: {1, 3}

Explanation: After sorting array will

be like {1, 3}.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **sortArr()** which takes the list of integers and the size N as inputs and returns the modified list

Q .5 Find the Kth largest and Kth smallest number in an array

Given an array **arr[]** and an integer **K** where K is smaller than size of array, the task is to find the **K^{th} smallest** element in the given array. It is given that all array elements are distinct.

Note :- **l** and **r** denotes the **starting** and **ending** index of the array.

Example 1:

Input:

N = 6

arr[] = 7 10 4 3 20 15

K = 3
L=0 R=5

Output : 7

Explanation :

3rd smallest element in the given

array is 7.

Example 2:

Input:

N = 5

arr[] = 7 10 4 20 15

K = 4 L=0 R=4

Output : 15

Explanation :

4th smallest element in the given

array is 15.

Your Task:
You don't have to read input or print anything. Your task is to complete the function **kthSmallest()** which takes the array **arr[]**, integers **l** and **r** denoting the **starting** and **ending** index of the array and an integer **K** as input and returns the $\mathbf{K^{th}}$ smallest element.

Q.6 Find the occurrence of an integer in the array

Given a vector of **N** positive integers and an integer **X**. The task is to find the **frequency** of X in vector.

Example 1:

Input:

N = 5

vector = {1, 1, 1, 1, 1}

X = 1

Output: 5

Explanation: Frequency of 1 is 5.

Your Task:
Your task is to complete the function **findFrequency**() which should count the frequency of X and return it.

Q.7 Sort the array of 0s, 1s, and 2s

Given an array of size **N** containing only 0s, 1s, and 2s; sort the array in ascending order.

Example 1:

Input: N = 5

arr[]= {0 2 1 2 0}

Output: 0 0 1 2 2

Explanation: 0s 1s and 2s are segregated into ascending order.

Example 2:

Input:

N = 3

arr[] = {0 1 0}

Output: 0 0 1

Explanation: 0s 1s and 2s are segregated into ascending order.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **sort012()** that takes an array **arr** and **N** as input parameters and sorts the array in-place.

Q.8 Subarray with given Sum

Given an unsorted array **A** of size **N** that contains only non negative integers, find a continuous sub-array that adds to a given number **S** and return the left and right index(**1-based indexing**) of that subarray.

In case of multiple subarrays, return the subarray indexes which come first on moving from left to right.

Note:- You have to return an ArrayList consisting of two elements left and right. In case no such subarray exists return an array consisting of element **-1**.

Example 1:

Input: N = 5, S = 12

A[] = {1,2,3,7,5}

Output: 2 4

Explanation: The sum of elements from 2nd position to 4th position is 12.

Example 2:

Input: N = 10, S = 15

A[] = {1,2,3,4,5,6,7,8,9,10}

Output: 1 5

Explanation: The sum of elements from 1st position to 5^{th} position is 15.

Your Task:
You don't need to read input or print anything. The task is to complete the function **subarraySum**() which takes arr, N, and S as input parameters and returns an **ArrayList** containing the **starting** and **ending** positions of the first such occurring subarray from the left where sum equals to S. The **two indexes in the array** should be according to **1-based indexing**. If no such subarray is found, return an array consisting of only one element that is -1.

Q.9 Move all the negative elements to one side of the array

Given an unsorted array **arr[]** of size **N** having both negative and positive integers. The task is place all negative element at the end of array without changing the order of positive element and negative element.

Example 1:

Input : N = 8

arr[] = {1, -1, 3, 2, -7, -5, 11, 6 }

Output : 1 3 2 11 6 -1 -7 -5

Example 2:

Input : N=8

arr[] = {-5, 7, -3, -4, 9, 10, -1, 11}

Output : 7 9 10 11 -5 -3 -4 -1

Your Task:
You don't need to read input or print anything. Your task is to complete the function **segregateElements()** which takes the array **arr[]** and its size **N** as inputs and **store** the answer in the array **arr[]** itself.

Q.10 Find the Union and Intersection of the two sorted arrays

Given two arrays **a[]** and **b[]** of size **n** and **m** respectively. The task is to find the number of elements in the union between these two arrays.

Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of element should be printed in the union.

Note : Elements are not necessarily distinct.

Example 1:

Input:

5 3

1 2 3 4 5

1 2 3

Output: 5

Explanation: 1, 2, 3, 4 and 5 are the elements which comes in the union set of both arrays. So count is 5.

Example 2:

Input:

6 2

85 25 1 32 54 6

85 2

Output: 7

Explanation: 85, 25, 1, 32, 54, 6, and 2 are the elements which comes in the union set of both arrays. So count is 7.

Your Task:
Complete **doUnion** function that takes **a, n, b, m as parameters and returns** the count of union elements of the two arrays. The **printing** is done by the **driver** code.

Q.11 Write a program to cyclically rotate an array by one

Given an array, rotate the array by one position in clock-wise direction.

Example 1:

Input:N = 5

A[] = {1, 2, 3, 4, 5}

Output: 5 1 2 3 4

Example 2:

Input: N = 8

A[] = {9, 8, 7, 6, 4, 2, 1, 3}

Output: 3 9 8 7 6 4 2 1

Your Task:
You don't need to read input or print anything. Your task is to complete the function **rotate()** which takes the array **A[]** and its size **N** as inputs and modify the array in place.

Q.12 Find the missing integer

Given an array of size **N-1** such that it only contains distinct integers in the range of **1 to N**. Find the missing element.

Example 1:

Input: N = 5

A[] = {1,2,3,5}

Output: 4

Example 2:

Input: N = 10

A[] = {6,1,2,8,3,4,7,10,5}

Output: 9

Your Task :

You don't need to read input or print anything. Complete the function **MissingNumber()** that takes array and N as input parameters and returns the value of the missing number.

Q.13 Count Pairs with the given sum

Given an array of **N** integers, and an integer **K**, find the number of pairs of elements in the array whose sum is equal to **K**.

Example 1:

Input: N = 4, K = 6

arr[] = {1, 5, 7, 1}

Output: 2

Explanation:

arr[0] + arr[1] = 1 + 5 = 6

and arr[1] + arr[3] = 5 + 1 = 6.

Example 2:

Input: N = 4, K = 2

arr[] = {1, 1, 1, 1}

Output: 6

Explanation: Each 1 will produce sum 2 with any 1.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **getPairsCount()** which takes **arr[], n** and **k** as input parameters and returns the number of pairs that have sum K.

Q.14 Find duplicates in an array

Given an array **a** of size **N** which contains elements from **0** to **N-1**, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing **[-1]**.

Note: The extra space is only for the array to be returned. Try and perform all operations within the provided array.

Example 1:

Input:

N = 4

a[] = {0,3,1,2}

Output:
-1

Explanation:
There is no repeating element in the array. Therefore output is -1.

Example 2:

Input:

N = 5

a[] = {2,3,1,2,3}

Output:
2 3

Explanation:
2 and 3 occur more than once in the given array.

Your Task:
Complete the function **duplicates()** which takes array a[] and n as input as parameters and returns a list of elements that occur more than once in the given array in a sorted manner.

Q.15 Sort an Array using the Quicksort algorithm

Quick Sort is a Divide and Conquer algorithm. It picks an element as a pivot and partitions the given array around the picked pivot.
Given an array arr[], its starting position is low (the index of the array) and its ending position is high(the index of the array).

Note: The **low** and **high** are inclusive.

Implement the partition() and quickSort() functions to sort the array.

Example 1:

Input:

N = 5

arr[] = { 4, 1, 3, 9, 7}

Output:

1 3 4 7 9

Example 2:

Input:

N = 9

arr[] = { 2, 1, 6, 10, 4, 1, 3, 9, 7}

Output:

1 1 2 3 4 6 7 9 10

Your Task:

You don't need to read input or print anything. Your task is to complete the functions **partition()** and **quickSort()** which takes the array arr[], low and high as input parameters and partitions the array. Consider the last element as the pivot such that all the elements less than(or equal to) the pivot lie before it and the elements greater than it lie after the pivot.

Q.16 Find common elements in three sorted arrays

Given three arrays sorted in increasing order. Find the elements that are common in all three arrays.

Note: can you take care of the duplicates without using any additional Data Structure?

Example 1:

Input:

n1 = 6; A = {1, 5, 10, 20, 40, 80}

n2 = 5; B = {6, 7, 20, 80, 100}

n3 = 8; C = {3, 4, 15, 20, 30, 70, 80, 120}

Output: 20 80

Explanation: 20 and 80 are the only

common elements in A, B and C.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **commonElements()** which take the 3 arrays A[], B[], C[] and their respective sizes n1, n2 and n3 as inputs and returns an array containing the common element present in all the 3 arrays in sorted order.
If there are no such elements return an empty array. In this case the output will be printed as -1.

Q.17 Find the first repeating element in an array of integers

Given an array arr[] of size n, find the first repeating element. The element should occur more than once and the index of its first occurrence should be the smallest.

Note:- The position you return should be according to 1-based indexing.

Example 1:

Input:

n = 7

arr[] = {1, 5, 3, 4, 3, 5, 6}

Output: 2

Explanation:

5 is appearing twice and

its first appearence is at index 2

which is less than 3 whose first

occuring index is 3.

Example 2:

Input:

n = 4

arr[] = {1, 2, 3, 4}

Output: -1

Explanation:

All elements appear only once so

answer is -1.

Your Task:
You don't need to read input or print anything. Complete the function **firstRepeated()** which takes arr and n as input parameters and returns the position of the first repeating element. If there is no such element, return -1.

Q.18 Find the first non-repeating element in a given array of integers

Find the first non-repeating element in a given array **arr** of **N** integers.
Note: Array consists of only positive and negative integers and **not zero**.

Example 1:

Input : arr[] = {-1, 2, -1, 3, 2}

Output : 3

Explanation:

-1 and 2 are repeating whereas 3 is

the only number occuring once.

Hence, the output is 3.

Example 2:

Input : arr[] = {1, 1, 1}

Output : 0

Your Task:
This is a function problem. The input is already taken care of by the driver code. You only need to complete the function **firstNonRepeating()** that takes an array **(arr)**, sizeOfArray **(n)**, and **returns** the first non-repeating element. The driver code takes care of the printing.

Q.19 Subarrays with equal 1s and 0s

Given an array containing 0s and 1s. Find the number of subarrays having equal number of 0s and 1s.

Example 1:

Input:

n = 7

A[] = {1,0,0,1,0,1,1}

Output: 8

Explanation: The index range for the 8

sub-arrays are: (0, 1), (2, 3), (0, 3), (3, 4),

(4, 5) ,(2, 5), (0, 5), (1, 6)

Example 2:

Input:

n = 5

A[] = {1,1,1,1,0}

Output: 1

Explanation: The index range for the

subarray is (3,4).

Your Task:
You don't need to read input or print anything. Your task is to complete the function **countSubarrWithEqualZeroAndOne()** which takes the array arr[] and the size of the array as inputs and returns the number of subarrays with equal number of 0s and 1s.

Q.20 Rearrange the array in alternating positive and negative items

Given an unsorted array **Arr** of **N** positive and negative numbers. Your task is to create an array of alternate positive and negative numbers without changing the relative order of positive and negative numbers.
Note: Array should start with a positive number and 0 (zero) should be considered a positive element.

Example 1:

Input:

N = 9

Arr[] = {9, 4, -2, -1, 5, 0, -5, -3, 2}

Output:

9 -2 4 -1 5 -5 0 -3 2

Explanation : Positive elements : 9,4,5,0,2

Negative elements : -2,-1,-5,-3

As we need to maintain the relative order of

postive elements and negative elements we will pick

each element from the positive and negative and will

store them. If any of the positive and negative numbers

are completed. we will continue with the remaining signed

elements.The output is 9,-2,4,-1,5,-5,0,-3,2.

Example 2:

Input:

N = 10

Arr[] = {-5, -2, 5, 2, 4, 7, 1, 8, 0, -8}

Output:

5 -5 2 -2 4 -8 7 1 8 0

Explanation : Positive elements : 5,2,4,7,1,8,0

Negative elements : -5,-2,-8

As we need to maintain the relative order of

postive elements and negative elements we will pick

each element from the positive and negative and will

store them. If any of the positive and negative numbers

are completed. we will continue with the remaining signed

elements.The output is 5,-5,2,-2,4,-8,7,1,8,0.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **rearrange()** which takes the array of integers **arr[]** and **n** as parameters. You need to modify the array itself.

Q.21 Find if there is any subarray with a sum equal to zero

Given an array of integers. Find if there is a **subarray** (of size at-least one) with **0 sum**. You just need to return true/false depending upon whether there is a subarray present with 0-sum or not. Printing will be taken care by the driver code.

Example 1:

Input:

n = 5

arr = {4,2,-3,1,6}

Output:

Yes

Explanation:

2, -3, 1 is the subarray with sum 0.

Example 2:

Input:

n = 5

arr = {4,2,0,1,6}

Output:

Yes

Explanation:

0 is one of the element in the array so there exist a subarray with sum 0.

Your Task:
You only need to complete the function **subArrayExists()** that takes **array** and **n** as **parameters** and **returns** true or false.

Q.22 Find the Largest sum contiguous Subarray

Given an array **Arr[]** of **N** integers. Find the contiguous sub-array(containing at least one number) which has the maximum sum and return its sum.

Example 1:

Input:

N = 5

Arr[] = {1,2,3,-2,5}

Output:

9

Explanation:

Max subarray sum is 9

of elements (1, 2, 3, -2, 5) which

is a contiguous subarray.

Example 2:

Input:

N = 4

Arr[] = {-1,-2,-3,-4}

Output:

-1

Explanation:

Max subarray sum is -1

of element (-1)

Your Task:

You don't need to read input or print anything. The task is to complete the function **maxSubarraySum**() which takes Arr[] and N as input parameters and returns the sum of subarray with maximum sum.

Q.23 Find the factorial of a large number

Given an integer N, find its factorial. **return a list of integers** denoting the digits that make up the factorial of **N**.

Example 1:

Input: N = 5

Output: 120

Explanation : 5! = 1*2*3*4*5 = 120

Example 2:

Input: N = 10

Output: 3628800

Explanation :

10! = 1*2*3*4*5*6*7*8*9*10 = 3628800

Your Task:
You don't need to read input or print anything. Complete the function *factorial()* that takes integer **N** as input parameter and returns a **list of integers** denoting the digits that make up the factorial of N.

Q.24 Find Maximum Product Subarray

Given an array **Arr[]** that contains **N** integers (may be **positive**, **negative** or **zero**). Find the product of the maximum product subarray.

Example 1:

Input:

N = 5

Arr[] = {6, -3, -10, 0, 2}

Output: 180

Explanation: Subarray with maximum product

is [6, -3, -10] which gives product as 180.

Example 2:

Input:

N = 6

Arr[] = {2, 3, 4, 5, -1, 0}

Output: 120

Explanation: Subarray with maximum product

is [2, 3, 4, 5] which gives product as 120.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **maxProduct()** which takes the array of integers **arr** and **n** as parameters and returns an integer denoting the answer.
Note: Use 64-bit integer data type to avoid overflow.

Q.25 Find the longest consecutive subsequence

Given an array of positive integers. Find the length of the longest sub-sequence such that elements in the subsequence are consecutive integers,
the **consecutive numbers can be in any order.**

Example 1:

Input:

N = 7

a[] = {2,6,1,9,4,5,3}

Output:

6

Explanation:

The consecutive numbers here

are 1, 2, 3, 4, 5, 6. These 6

numbers form the longest consecutive

subsquence.

Example 2:

Input:

N = 7

a[] = {1,9,3,10,4,20,2}

Output:

4

Explanation:

1, 2, 3, 4 is the longest

consecutive subsequence.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **findLongestConseqSubseq()** which takes the array arr[] and the size of the array as inputs and returns the length of the longest subsequence of consecutive integers.

Q.26 Find the minimum element in a rotated and sorted array

A sorted(in ascending order) array A[ ] with distinct elements is rotated at some unknown point, the task is to find the minimum element in it.

Example 1

Input:

N = 5

arr[] = {4 ,5 ,1 ,2 ,3}

Output: 1

Explanation: 1 is the minimum element in the array.

Example 2

Input:

N = 7

arr[] = {10, 20, 30, 40, 50, 5, 7}

Output: 5

Explanation: Here 5 is the minimum element.

Your Task:

Complete the function **findMin**() which takes an array **arr**[] and **n**, size of the array as input parameters, and returns the minimum element of the array.

Q.27 Max sum in the configuration

Given an array(0-based indexing), you have to find the max sum of i*A[i] where A[i] is the element at index i in the array. The only operation allowed is to rotate(clock-wise or counter clock-wise) the array any number of times.

Example 1:

Input:

N = 4

A[] = {8,3,1,2}

Output: 29

Explanation: Above the configuration

possible by rotating elements are

3 1 2 8 here sum is 3*0+1*1+2*2+8*3 = 29

1 2 8 3 here sum is 1*0+2*1+8*2+3*3 = 27

2 8 3 1 here sum is 2*0+8*1+3*2+1*3 = 17

8 3 1 2 here sum is 8*0+3*1+1*2+2*3 = 11

Here the max sum is 29

Your Task:

Your task is to complete the function **max_sum** which takes two arguments which is the array A [ ] and its size and returns an integer value denoting the required max sum.

Q.28 Minimize the maximum difference between the heights.

Given an array(0-based indexing), you have to find the max sum of i*A[i] where A[i] is the element at index i in the array. The only operation allowed is to rotate(clock-wise or counter clock-wise) the array any number of times.

Example 1:

Input:

N = 4

A[] = {8,3,1,2}

Output: 29

Explanation: Above the configuration

possible by rotating elements are

3 1 2 8 here sum is 3*0+1*1+2*2+8*3 = 29

1 2 8 3 here sum is 1*0+2*1+8*2+3*3 = 27

2 8 3 1 here sum is 2*0+8*1+3*2+1*3 = 17

8 3 1 2 here sum is 8*0+3*1+1*2+2*3 = 11

Here the max sum is 29

Your Task:

Your task is to complete the function **max_sum** which takes two arguments which is the array A [ ] and its size and returns an integer value denoting the required max sum.

Q.29 Minimum number of jumps to reach the end

Given an array of **N** integers **arr[]** where each element represents the **maximum** length of the jump that can be made forward from that element. This means if arr[i] = x, then we can jump any distance y such that y ≤ x.

Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is **0**, then you cannot move through that element.

Note: Return -1 if you can't reach the end of the array.

Example 1:

Input:

N = 11

arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}

Output: 3

Explanation:

First jump from 1st element to 2nd

element with value 3. Now, from here

we jump to 5th element with value 9,

and from here we will jump to the last.

Example 2:

Input : N = 6

arr = {1, 4, 3, 2, 6, 7}

Output: 2

Explanation: First we jump from the 1st to 2nd element and then jump to the last element.

Your task:
You don't need to read input or print anything. Your task is to complete function **minJumps()** which takes the array **arr** and it's size **N** as input parameters and returns the minimum number of jumps. If not possible return -1.

Q.30 Find a triplet that sums to a given value

Given an array arr of size n and an integer X. Find if there's a triplet in the array which sums up to the given integer X.

Example 1:

Input: n = 6, X = 13

arr[] = [1 4 45 6 10 8]

Output: 1

Explanation: The triplet {1, 4, 8} in the array sums up to 13.

Example 2:

Input: n = 5, X = 10

arr[] = [1 2 4 3 6]

Output: 1

Explanation: The triplet {1, 3, 6} in the array sums up to 10.

Your Task:
You don't need to read input or print anything. Your task is to complete the function **find3Numbers()** which takes the array arr[], the size of the array (n) and the sum (X) as inputs and returns True if there exists a triplet in the array arr[] which sums up to X and False otherwise.

Q.31 Smallest positive missing number

You are given an array arr[] of N integers. The task is to find the smallest positive number missing from the array.

Note: Positive number starts from 1.

Example 1:

Input: N = 5

arr[] = {1,2,3,4,5}

Output: 6

Explanation: Smallest positive missing number is 6.

Example 2:

Input: N = 5

arr[] = {0,-10,1,3,-20}

Output: 2

Explanation: Smallest positive missing number is 2.

Your Task:

The task is to complete the function missingNumber() which returns the smallest positive missing number in the array.

Q.32 Find the row with a maximum number of 1’s

Given a boolean 2D array of n x m dimensions where each row is sorted. Find the 0-based index of the first row that has the maximum number of 1's.

Example 1:

Input:

N = 4 , M = 4

Arr[][] = {{0, 1, 1, 1},

{0, 0, 1, 1},

{1, 1, 1, 1},

{0, 0, 0, 0}}

Output: 2

Explanation: Row 2 contains 4 1's (0-based

indexing).

Example 2:

Input:

N = 2, M = 2

Arr[][] = {{0, 0}, {1, 1}}

Output: 1

Explanation: Row 1 contains 2 1's (0-based

indexing).

Your Task:

You don't need to read input or print anything. Your task is to complete the function rowWithMax1s() which takes the array of booleans arr[][], n and m as input parameters and returns the 0-based index of the first row that has the most number of 1s. If no such row exists, return -1.

Q.33 Print the matrix in a Spiral manner

Given a matrix of size r*c. Traverse the matrix in spiral form.

Example 1:

Input: r = 4, c = 4

matrix[][] = {{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12},

{13, 14, 15,16}}

Output:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Example 2:

Input: r = 3, c = 4

matrix[][] = {{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12}}

Output: 1 2 3 4 8 12 11 10 9 5 6 7

Explanation:

Applying same technique as shown above,

output for the 2nd testcase will be

1 2 3 4 8 12 11 10 9 5 6 7.

Your Task:

You dont need to read input or print anything. Complete the function spirallyTraverse() that takes matrix, r and c as input parameters and returns a list of integers denoting the spiral traversal of matrix.

Q.34 Find whether an array is a subset of another array

Given two arrays: a1[0..n-1] of size n and a2[0..m-1] of size m. Task is to check whether a2[] is a subset of a1[] or not. Both the arrays can be sorted or unsorted. There can be duplicate elements.

Example 1:

Input: a1[] = {11, 7, 1, 13, 21, 3, 7, 3}

a2[] = {11, 3, 7, 1, 7}

Output: Yes

Explanation: a2[] is a subset of a1[]

Example 2:

Input: a1[] = {1, 2, 3, 4, 4, 5, 6}

a2[] = {1, 2, 4}

Output: Yes

Explanation: a2[] is a subset of a1[]

Example 3:

Input: a1[] = {10, 5, 2, 23, 19}

a2[] = {19, 5, 3}

Output: No

Explanation: a2[] is not a subset of a1[]

Your Task:

You don't need to read input or print anything. Your task is to complete the function isSubset() which takes the array a1[], a2[], its size n and m as inputs and return "Yes" if arr2 is subset of arr1 else return "No" if arr2 is not subset of arr1.

Q.35 Majority Element

Given an array A of N elements. Find the majority element in the array. A majority element in an array A of size N is an element that appears strictly more than N/2 times in the array.

Example 1:

Input: N = 3

A[] = {1,2,3}

Output:-1

Explanation: Since, each element in {1,2,3} appears only once so there is no majority element.

Example 2:

Input: N = 5

A[] = {3,1,3,3,2}

Output: 3

Explanation: Since, 3 is present more than N/2 times, so it is

the majority element.

Your Task:

The task is to complete the function majorityElement() which returns the majority element in the array. If no majority exists, return -1.

Q.36 Wave Array

Given a sorted array arr[] of distinct integers. Sort the array into a wave-like array(In Place).

In other words, arrange the elements into a sequence such that arr[1] >= arr[2] <= arr[3] >= arr[4] <= arr[5].....

If there are multiple solutions, find the lexicographically smallest one.

Note:The given array is sorted in ascending order, and you don't need to return anything to make changes in the original array itself.

Example 1:

Input: n = 5

arr[] = {1,2,3,4,5}

Output: 2 1 4 3 5

Explanation: Array elements after sorting it in wave form are

2 1 4 3 5.

Example 2:

Input: n = 6

arr[] = {2,4,7,8,9,10}

Output: 4 2 8 7 10 9

Explanation: Array elements after sorting it in wave form are

4 2 8 7 10 9.

Your Task: The task is to complete the function convertToWave(), which converts the given array to a wave array.

Q.37 Wave Array

Given a sorted array arr[] of distinct integers. Sort the array into a wave-like array(In Place).

In other words, arrange the elements into a sequence such that arr[1] >= arr[2] <= arr[3] >= arr[4] <= arr[5].....

If there are multiple solutions, find the lexicographically smallest one.

Note:The given array is sorted in ascending order, and you don't need to return anything to make changes in the original array itself.

Example 1:

Input: n = 5

arr[] = {1,2,3,4,5}

Output: 2 1 4 3 5

Explanation: Array elements after sorting it in wave form are

2 1 4 3 5.

Example 2:

Input: n = 6

arr[] = {2,4,7,8,9,10}

Output: 4 2 8 7 10 9

Explanation: Array elements after sorting it in wave form are

4 2 8 7 10 9.

Your Task:

The task is to complete the function convertToWave(), which converts the given array to a wave array.

Q.38 Maximum Index

Given an array A[] of N positive integers. The task is to find the maximum of j - i subjected to the constraint of A[i] < A[j] and i < j.

Example 1:

Input: N = 2

A[] = {1, 10}

Output: 1

Explanation: A[0]<A[1] so (j-i) is 1-0 = 1.

Example 2:

Input: N = 9

A[] = {34, 8, 10, 3, 2, 80, 30, 33, 1}

Output: 6

Explanation: In the given array A[1] < A[7] satisfying the required condition(A[i] < A[j]) thus giving the maximum difference of j - i which is 6(7-1).

Your Task:

The task is to complete the function maxIndexDiff() which finds and returns maximum index difference. Printing the output will be handled by driver code. Return 0 in case no such index is found.

Q.39 Max sum path in two arrays

Given two sorted arrays A and B of size M and N respectively. Each array may have some elements in common with the other array. Find the maximum sum of a path from the beginning of any array to the end of any of the two arrays.

We can switch from one array to another array only at the common elements. Both the arrays are sorted.

Note: Only one repeated value is considered in the valid path sum.

Example 1:

Input: M = 5, N = 4

A[] = {2,3,7,10,12}

B[] = {1,5,7,8}

Output: 35

Explanation: The path will be 1+5+7+10+12

= 35.

Example 2:

Input: M = 3, N = 3

A[] = {1,2,3}

B[] = {3,4,5}

Output: 15

Explanation: The path will be 1+2+3+4+5=15.

Your Task:

You don't need to read input or print anything. Complete the function max_path_sum() which takes the two arrays A and B along with their sizes M and N as input parameters. It returns the maximum path sum.

Q.40 Find Missing And Repeating

Given an unsorted array Arr of size N of positive integers. One number 'A' from set {1, 2,....,N} is missing and one number 'B' occurs twice in array. Find these two numbers.

Example 1:

Input: N = 2

Arr[] = {2, 2}

Output: 2 1

Explanation: Repeating number is 2 and

smallest positive missing number is 1.

Example 2:

Input: N = 3

Arr[] = {1, 3, 3}

Output: 3 2

Explanation: Repeating number is 3 and

smallest positive missing number is 2.

Your Task:

You don't need to read input or print anything. Your task is to complete the function findTwoElement() which takes the array of integers arr and n as parameters and returns an array of integers of size 2 denoting the answer (The first index contains B and second index contains A.)

Q.41 Chocolate Distribution Problem

Given an array A[ ] of positive integers of size N, where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are M students, the task is to distribute chocolate packets among M students such that :

1. Each student gets exactly one packet.

2. The difference between maximum number of chocolates given to a student and minimum number of chocolates given to a student is minimum.

Example 1:

Input: N = 8, M = 5

A = {3, 4, 1, 9, 56, 7, 9, 12}

Output: 6

Explanation: The minimum difference between maximum chocolates and minimum chocolates is 9 - 3 = 6 by choosing following M packets :{3, 4, 9, 7, 9}.

Example 2:

Input: N = 7, M = 3

A = {7, 3, 2, 4, 9, 12, 56}

Output: 2

Explanation: The minimum difference between maximum chocolates and minimum chocolates is 4 - 2 = 2 by choosing following M packets :{3, 2, 4}.

Your Task:

You don't need to take any input or print anything. Your task is to complete the function findMinDiff() which takes array A[ ], N and M as input parameters and returns the minimum possible difference between maximum number of chocolates given to a student and minimum number of chocolates given to a student.

Q.42 Partition Equal Subset Sum

Given an array arr[] of size N, check if it can be partitioned into two parts such that the sum of elements in both parts is the same.

Example 1:

Input: N = 4

arr = {1, 5, 11, 5}

Output: YES

Explanation:

The two parts are {1, 5, 5} and {11}.

Example 2:

Input: N = 3

arr = {1, 3, 5}

Output: NO

Explanation: This array can never be

partitioned into two such parts.

Your Task:

You do not need to read input or print anything. Your task is to complete the function equalPartition() which takes the value N and the array as input parameters and returns 1 if the partition is possible. Otherwise, returns 0.

Q.43 Smallest Positive integer that can’t be represented as a sum

Given an array of size N, find the smallest positive integer value that is either not presented in the array or cannot be represented as a sum(coz sum means you are adding two or more elements) of some elements from the array.

Example 1:

Input:

N = 6

array[] = {1, 10, 3, 11, 6, 15}

Output:

2

Explanation:

2 is the smallest integer value that cannot

be represented as sum of elements from the array.

Example 2:

Input:

N = 3

array[] = {1, 1, 1}

Output:

4

Explanation:

1 is present in the array.

2 can be created by combining two 1s.

3 can be created by combining three 1s.

4 is the smallest integer value that cannot be

represented as sum of elements from the array.

Your Task:

You don't need to read input or print anything. Complete the function smallestpositive() which takes the array and N as input parameters and returns the smallest positive integer value that cannot be represented as the sum of some elements from the array.

Q.44 Coin Change Problem

Given an integer array coins[ ] of size N representing different denominations of currency and an integer sum, find the number of ways you can make sum by using different combinations from coins[ ].

Note: Assume that you have an infinite supply of each type of coin. And you can use any coin as many times as you want.

Example 1:

Input: N = 3, sum = 4

coins = {1,2,3}

Output: 4

Explanation: Four Possible ways are: {1,1,1,1},{1,1,2},{2,2},{1,3}.

Example 2:

Input: N = 4, Sum = 10

coins = {2,5,3,6}

Output: 5

Explanation: Five Possible ways are: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}.

Your Task:

You don't need to read input or print anything. Your task is to complete the function count() which accepts an array coins its size N and sum as input parameters and returns the number of ways to make change for given sum of money.

Q.45 Longest Alternating subsequence

A sequence {x1, x2, .. xn} is alternating sequence if its elements satisfy one of the following relations :

x1 < x2 > x3 < x4 > x5..... or x1 >x2 < x3 > x4 < x5.....

Your task is to find the longest such sequence.

Example 1:

Input: nums = {1,5,4}

Output: 3

Explanation: The entire sequenece is a alternating sequence.

Example 2:

Input: nums = {1,17,5,10,13,15,10,5,16,8}

Output: 7

Explanation: There are several sub sequences that achieve this length. One is {1,17,10,13,10,16,8}.

Your Task: You don't need to read or print anyhting. Your task is to complete the function AlternatingaMaxLength() which takes the sequence nums as input parameter and returns the maximum length of alternating sequence.

Q.46 Sorting refers to arranging data in a particular format. Sort an array of integers in ascending order. One of the algorithm is selection sort. Use below explanation of selection sort to do this.

INITIAL ARRAY :

2 3 1 45 15

First iteration : Compare every element after first element with first element and if it is larger then swap. In first iteration, 2 is larger than 1. So, swap it.

1 3 2 45 15

Second iteration : Compare every element after second element with second element and if it is larger then swap. In second iteration, 3 is larger than 2. So, swap it.

1 2 3 45 15

Third iteration : Nothing will swap as 3 is smaller than every element after it.

1 2 3 45 15

Fourth iteration : Compare every element after fourth element with fourth element and if it is larger then swap. In fourth iteration, 45 is larger than 15. So, swap it.

1 2 3 15 45

Q.47 Input any number. Find the sum of the digits of the number using a recursive function.

Q.48 If the input array is [10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60], your program should be able to find that the subarray lies between the indexes 3 and 8.

Q.49 ake an array of 10 elements. Split it into middle and store the elements in two dfferent arrays. E.g.-

INITIAL array :

58 24 13 15 63 9 8 81 1 78

After spliting :

58 24 13 15 63

9 8 81 1 78

Q.50 Write a Java program to find common elements between two arrays (string values).

Q.51 Write a Java program to find common elements between two integer arrays.

Q.52 Write a Java program to remove duplicate elements from an array.

Q.53 Write a Java program to find the second largest element in an array.

Q.54 Write a Java program to add two matrices of the same size.

Q.55 Write a Java program to convert an ArrayList to an array.

Q.56 Write a Java program to find all pairs of elements in an array whose sum is equal to a specified number.

Q.57 Write a Java program to find a missing number in an array.

Q.58 Write a Java program to find common elements in three sorted (in non-decreasing order) arrays.

Q.59 Write a Java program to move all 0's to the end of an array. Maintain the relative order of the other (non-zero) array elements.

Q.60 Write a Java program to find the number of even and odd integers in a given array of integers.

Q.61 Write a Java program to check if an array of integers is without 0 and -1.

Q.62 Write a Java program to check if the sum of all the 10's in the array is exactly 30. Return false if the condition does not satisfy, otherwise true.

Q.63 Write a Java program to remove duplicate elements from a given array and return the updated array length.

Sample array: [20, 20, 30, 40, 50, 50, 50]

After removing the duplicate elements the program should return 4 as the new length of the array.

Q.64 Write a Java program to find the sum of the two elements of a given array equal to a given integer.

Sample array: [1,2,4,5,6]

Target value: 6

Q.65 Write a Java program that returns the missing letter from an array of increasing letters (upper or lower). Assume there will always be one omission from the array.

Example:

Original array of elements:

[p, r, s, t]

Missing letter in the said array: q

Q.66 Write a Java program that checks whether an array is negative dominant or not. If the array is negative dominant return true otherwise false.

Example:

Original array of numbers:

[1, -2, -5, -4, 3, -6]

Check Negative Dominance in the said array!true

Q.67 Write a Java program that checks whether an array of integers alternates between positive and negative values.

Example:

Original array: [1, -2, 5, -4, 3, -6]

Check the said array of integers alternates between positive and negative values!true

Q.68 Write a Java program to calculate the largest gap between sorted elements of an array of integers.

Example:

Original array: [23, -2, 45, 38, 12, 4, 6]

Largest gap between sorted elements of the said array: 15

Q.69 Write a Java program to find all triplets equal to a given sum in an unsorted array of integers.

Example:

Input : nums = { 1, 6, 3, 0, 8, 4, 1, 7 }

Output: Triplets of sum 7

(0 1 6)

(0 3 4)

Q.70 Write a Java program to sort a given array of distinct integers where all its numbers are sorted except two numbers.

Example:

Input : nums1 = { 3, 5, 6, 9, 8, 7 }

nums2 = { 5, 0, 1, 2, 3, 4, -2 }

Output: After sorting new array becomes: [3, 5, 6, 7, 8, 9]

After sorting new array becomes: [-2, 0, 1, 2, 3, 4, 5]

Q.71 Write a Java program to replace every element with the next greatest element (from the right side) in a given array of integers.

There is no element next to the last element, therefore replace it with -1.

Q.72 Write a Java program to check if a given array contains a subarray with 0 sum.

Example:

Input :

nums1= { 1, 2, -2, 3, 4, 5, 6 }

nums2 = { 1, 2, 3, 4, 5, 6 }

nums3 = { 1, 2, -3, 4, 5, 6 }

Output:

Does the said array contain a subarray with 0 sum: true

Does the said array contain a subarray with 0 sum: false

Does the said array contain a subarray with 0 sum: true

Q.73 Write a Java program to check if a sub-array is formed by consecutive integers from a given array of integers.

Example:

Input :

nums = { 2, 5, 0, 2, 1, 4, 3, 6, 1, 0 }

Output:

The largest sub-array is [1, 7]

Elements of the sub-array: 5 0 2 1 4 3 6

Q.74 Given two sorted arrays A and B of size p and q, write a Java program to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.

Example:

Input :

int[] A = { 1, 5, 6, 7, 8, 10 }

int[] B = { 2, 4, 9 }

Output: Sorted Arrays:

A: [1, 2, 4, 5, 6, 7]

B: [8, 9, 10]

Q.75 Write a Java program to find the maximum product of two integers in a given array of integers.

Example:

Input :

nums = { 2, 3, 5, 7, -7, 5, 8, -5 }

Output:

Pair is (7, 8), Maximum Product: 56

Q.76 Write a Java program to rearrange a given array of unique elements such that every second element of the array is greater than its left and right elements.

Example:

Input :

nums= { 1, 2, 4, 9, 5, 3, 8, 7, 10, 12, 14 }

Output:

Array with every second element is greater than its left and right elements:

[1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12]

Q.77 Write a Java program to replace each element of the array with the product of every other element in a given array of integers.

Example:

Input :

nums1 = { 1, 2, 3, 4, 5, 6, 7}

nums2 = {0, 1, 2, 3, 4, 5, 6, 7}

Output:

Array with product of every other element:

[5040, 2520, 1680, 1260, 1008, 840, 720]

Array with product of every other element:

[5040, 0, 0, 0, 0, 0, 0, 0]

Q.78 Write a Java program to find the Longest Bitonic Subarray in a given array.

A bitonic subarray is a subarray of a given array where elements are first sorted in increasing order, then in decreasing order. A strictly increasing or strictly decreasing subarray is also accepted as bitonic subarray.

Example:

Input :

nums = { 4, 5, 9, 5, 6, 10, 11, 9, 6, 4, 5 }

Output:

The longest bitonic subarray is [3,9]

Elements of the said sub-array: 5 6 10 11 9 6 4

The length of longest bitonic subarray is 7

Q.79 Write a Java program to find the maximum difference between two elements in a given array of integers such that the smaller element appears before the larger element.

Example:

Input : nums = { 2, 3, 1, 7, 9, 5, 11, 3, 5 }

Output: The maximum difference between two elements of the said array elements

10

Q.80 Write a Java program to find a contiguous subarray within a given array of integers with the largest sum.

In computer science, the maximum sum subarray problem is the task of finding a contiguous subarray with the largest sum, within a given one-dimensional array A[1...n] of numbers. Formally, the task is to find indices and with, such that the sum is as large as possible.

Example:

Input : int[] A = {1, 2, -3, -4, 0, 6, 7, 8, 9}

Output: The largest sum of contiguous sub-array: 30The

Q.81 Write a Java program to find the subarray with the largest sum in a given circular array of integers.

Example:

Input :

nums1 = { 2, 1, -5, 4, -3, 1, -3, 4, -1 }

nums2 = { 1, -2, 3, 0, 7, 8, 1, 2, -3 }

Output:

The sum of subarray with the largest sum is 6

The sum of subarray with the largest sum is 21

Q.82 Write a Java program to create an array of its anti-diagonals from a given square matrix.

Example:

Input :

1 2

3 4

Output:

[

[1],

[2, 3],

[4]

]

Q.83 Write a Java program to get the majority element from an array of integers containing duplicates.

Majority element: A majority element is an element that appears more than n/2 times where n is the array size.

Q.84 Write a Java program to print all the LEADERS in the array.

Note: An element is leader if it is greater than all the elements to its right side.

Q.85 Write a Java program to print all the LEADERS in the array.

Note: An element is leader if it is greater than all the elements to its right side.

Q.86 Write a Java program to find the two elements in a given array of positive and negative numbers such that their sum is close to zero.

Q.87 Write a Java program to find the smallest and second smallest elements of a given array.

Q.88 Write a Java program to find all combinations of four elements of an array whose sum is equal to a given value.

Q.89 Write a Java program to count the number of possible triangles from a given unsorted array of positive integers.

Note: The triangle inequality states that the sum of the lengths of any two sides of a triangle must be greater than or equal to the length of the third side.

Q.90 Write a Java program to cyclically rotate a given array clockwise by one.

Q.91 Write a Java program to check whether there is a pair with a specified sum in a given sorted and rotated array.

Q.92 Write a Java program to find the rotation count in a given rotated sorted array of integers.

Q.93 Write a Java program to arrange the elements of an array of integers so that all negative integers appear before all positive integers.

Q.94 Write a Java program to arrange the elements of an array of integers so that all positive integers appear before all negative integers.

Q.95 Write a Java program to sort an array of positive integers from an array. In the sorted array the value of the first element should be maximum, the second value should be a minimum, third should be the second maximum, the fourth should be the second minimum and so on.

Q.96 Write a Java program to find all combinations of four elements of an array whose sum is equal to a given value.

Q.97 Write a Java program to test two arrays' equality.

Q.98 Write a Java program to find all combinations of four elements of an array whose sum is equal to a given value.

Q.99 Write a Java program to separate even and odd numbers from a given array of integers. Put all even numbers first, and then odd numbers.

Q.100 Write a Java program to shuffle a given array of integers.

Example:

Input : nums = { 1, 2, 3, 4, 5, 6 }

Output: Shuffle Array: [4, 2, 6, 5, 1, 3]

Q. 101

Stickler the thief wants to loot money from a society having n houses in a single line. He is a weird person and follows a certain rule when looting the houses. According to the rule, he will never loot two consecutive houses. At the same time, he wants to maximize the amount he loots. The thief knows which house has what amount of money but is unable to come up with an optimal looting strategy. He asks for your help to find the maximum money he can get if he strictly follows the rule. ith house has a[i] amount of money present in it.

Example 1:

Input:

n = 5

a[] = {6,5,5,7,4}

Output:

15

Explanation:

Maximum amount he can get by looting 1st, 3rd and 5th house. Which is 6+5+4=15.

Example 2:

Input:

n = 3

a[] = {1,5,3}

Output:

5

Explanation:

Loot only 2nd house and get maximum amount of 5.

Your Task:

Complete the functionFindMaxSum() which takes an array arr[] and n as input which returns the maximum money he can get following the rules.

Q. 102

Given an array arr of n elements that is first strictly increasing and then maybe strictly decreasing, find the maximum element in the array.

Note: If the array is increasing then just print the last element will be the maximum value.

Example 1:

Input:

n = 9

arr[] = {1,15,25,45,42,21,17,12,11}

Output: 45

Explanation: Maximum element is 45.

Example 2:

Input:

n = 5

arr[] = {1, 45, 47, 50, 5}

Output: 50

Explanation: Maximum element is 50.

Your Task:

You don't need to read input or print anything. Your task is to complete the function findMaximum() which takes the array arr[], and n as parameters and returns an integer denoting the answer.

Q. 103

Given a sorted array with possibly duplicate elements. The task is to find indexes of first and last occurrences of an element X in the given array.

Note: If the element is not present in the array return {-1,-1} as pair.

Example 1:

Input:

N = 9

v[] = {1, 3, 5, 5, 5, 5, 67, 123, 125}

X = 5

Output:

2 5

Explanation:

Index of first occurrence of 5 is 2

and index of last occurrence of 5 is 5.

Example 2:

Input:

N = 9

v[] = {1, 3, 5, 5, 5, 5, 7, 123, 125}

X = 7

Output:

6 6

Your Task:

You don't need to read input or print anything. Your task is to complete the function indexes() which takes the array v[] and an integer X as inputs and returns the first and last occurrence of the element X. If the element is not present in the array return {-1,-1} as pair.

Can you solve the problem in expected time complexity?

Q. 104 Nearly sorted

Given an array of n elements, where each element is at most k away from its target position, you need to sort the array optimally.

Example 1:

Input:

n = 7, k = 3

arr[] = {6,5,3,2,8,10,9}

Output: 2 3 5 6 8 9 10

Explanation: The sorted array will be

2 3 5 6 8 9 10

Example 2:

Input:

n = 5, k = 2

arr[] = {3,1,4,2,5}

Output: 1 2 3 4 5

Note: DO NOT use STL sort() function for this question.

Your Task:

You are required to complete the method nearlySorted() which takes 3 arguments and returns the sorted array.

Q. 105 Matrix interchange

Working with 2D arrays is quite important. Here we will do swapping of column in a 2D array. You are given a matrix M or r rows and c columns. You need to swap the first column with the last column.

Example:

Input:

3 4

1 2 3 4

4 3 2 1

6 7 8 9

Output:

4 2 3 1

1 3 2 4

9 7 8 6

Your Task:

Since this is a function problem, you don't need to take any input. Just complete the provided function interchange(int, int , int) that take matrix, rows and columns as parameters.

Q.106 Key Pair

Given an array Arr of N positive integers and another number X. Determine whether or not there exist two elements in Arr whose sum is exactly X.

Example 1:

Input:

N = 6, X = 16

Arr[] = {1, 4, 45, 6, 10, 8}

Output: Yes

Explanation: Arr[3] + Arr[4] = 6 + 10 = 16

Example 2:

Input:

N = 5, X = 10

Arr[] = {1, 2, 4, 3, 6}

Output: Yes

Explanation: Arr[2] + Arr[4] = 4 + 6 = 10

Your Task:

You don't need to read input or print anything. Your task is to complete the function hasArrayTwoCandidates() which takes the array of integers arr, n and x as parameters and returns a boolean denoting the answer.

Q.107 Second Largest

Given an array Arr of size N, print second largest distinct element from an array.

Example 1:

Input:

N = 6

Arr[] = {12, 35, 1, 10, 34, 1}

Output: 34

Explanation: The largest element of the

array is 35 and the second largest element

is 34.

Example 2:

Input:

N = 3

Arr[] = {10, 5, 10}

Output: 5

Explanation: The largest element of

the array is 10 and the second

largest element is 5.

Your Task:

You don't need to read input or print anything. Your task is to complete the function print2largest() which takes the array of integers arr and n as parameters and returns an integer denoting the answer. If 2nd largest element doesn't exist then return -1.

Q.108 Intersection of two arrays

Given two arrays a[] and b[] respectively of size n and m, the task is to print the count of elements in the intersection (or common elements) of the two arrays.

For this question, the intersection of two arrays can be defined as the set containing distinct common elements between the two arrays.

Example 1:

Input:

n = 5, m = 3

a[] = {89, 24, 75, 11, 23}

b[] = {89, 2, 4}

Output: 1

Explanation:

89 is the only element

in the intersection of two arrays.

Example 2:

Input:

n = 6, m = 5

a[] = {1, 2, 3, 4, 5, 6}

b[] = {3, 4, 5, 6, 7}

Output: 4

Explanation:

3 4 5 and 6 are the elements

in the intersection of two arrays.

Your Task:

You don't need to read input or print anything. Your task is to complete the function NumberofElementsInIntersection() which takes two integers n and m and their respective arrays a[] and b[] as input. The function should return the count of the number of elements in the intersection.

Q.109 Move all zeros at the end

Given an array arr[] of n positive integers. Push all the zeros of the given array to the right end of the array while maintaining the order of non-zero elements. Do the mentioned change in the array in-place.

Example 1:

Input:

N = 5

Arr[] = {3, 5, 0, 0, 4}

Output: 3 5 4 0 0

Explanation: The non-zero elements

preserve their order while the 0

elements are moved to the right.

Example 2:

Input:

N = 4

Arr[] = {0, 0, 0, 4}

Output: 4 0 0 0

Explanation: 4 is the only non-zero

element and it gets moved to the left.

Your Task:

You don't need to read input or print anything. Complete the function pushZerosToEnd() which takes the array arr[] and its size n as input parameters and modifies arr[] in-place such that all the zeroes are moved to the right.

Q.110 Union of two sorted array

Union of two arrays can be defined as the common and distinct elements in the two arrays.

Given two sorted arrays of size n and m respectively, find their union.

Example 1:

Input:

n = 5, arr1[] = {1, 2, 3, 4, 5}

m = 3, arr2 [] = {1, 2, 3}

Output: 1 2 3 4 5

Explanation: Distinct elements including

both the arrays are: 1 2 3 4 5.

Example 2:

Input:

n = 5, arr1[] = {2, 2, 3, 4, 5}

m = 5, arr2[] = {1, 1, 2, 3, 4}

Output: 1 2 3 4 5

Explanation: Distinct elements including

both the arrays are: 1 2 3 4 5.

Example 3:

Input:

n = 5, arr1[] = {1, 1, 1, 1, 1}

m = 5, arr2[] = {2, 2, 2, 2, 2}

Output: 1 2

Explanation: Distinct elements including

both the arrays are: 1 2.

Your Task:

You do not need to read input or print anything. Complete the function findUnion() that takes two arrays arr1[], arr2[], and their size n and m as input parameters and returns a list containing the union of the two arrays.

String

1.Write a Java program to concatenate Two strings.

2.Write a Java program to get the character at the given index within the String.

3.Write a Java program to count a number of Unicode code points in the specified text range of a String.

4.Write a Java program to compare two strings lexicographically Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions.

5.WAP to count the word whose first letter is a vowel.

6.Write a Java program to compare two strings lexicographically , ignoring case differences.

7.Write a Java program to concatenate a given string to the end of another string.

8.Write a Java program to test if a given string contains the specified sequence of char values.

9.Write a Java program to check whether a given string ends with the contents of another string.

10.Write a Java program to check whether two String objects contain the same data.

11.Write a Java program to get the contents of a given string as a byte array.

12.Write a Java program to get the canonical representation of the string object.

13.Write a Java program to create a character array containing the contents of a string.

14.Write a Java program to convert all the characters in a string to Lowercase.

15.Write a Java program to convert all the characters in a string to Uppercase.

16.Write a java program to get the length of a given string.

17.Write a Java program to replace a specified character with another character.

18.Write a Java program to replace each substring of a given string that matches the given regular expression with the given replacement.

19.Write a Java program to check whether a given string starts with the contents of another string.

20. Write a Java program to get a substring of a given string between two specified positions.

21.Write a Java program to trim any leading or trailing whitespace from a given string.

22. Write a Java program to create a new string repeating every character twice of a given string.

23.Write a Java program to return the sum of the digits present in the given string. If there are no digits the sum return is 0.

24.Write a Java program to Count words in Given String.

25.Write a Java program to Swap Two Strings.

26.Write a Java program to Swap Two Strings without a Third String Variable.

27.Write a Java program to Reverse Each Word of a String.

28.Write a Java program How to search a word inside a string?.

29. Write a Java program to find the first non repeating character in a string.

30. Write a Java program to print after removing duplicates from a given string.

31. Write a Java program to find the maximum occurring character in a string.

32.Write a Java program to reverse words in a given string.

33.Write a Java program to find the maximum between two strings.

34.Write a Java program How to check Palindrome String.

35.Write a Java program to check if the letter 'x' is present in the word 'String Exercises'.

36.Write a Java program to Convert Lowercase to Uppercase.

37.Write a Java program to Convert Uppercase to Lowercase.

38.Write a Java program to Count Number of Uppercase and Lowercase letters

39.Write a program to Trim a given string using String.trim() method

40.Write a program to Replace string with another string in java using String.replace() method.

41.Write a program to Convert any type of value to string value using String.valueOf() method.

42.Write a program to Compare the strings using equals(), compareTo() and == operator.

43.Write a program to First alphabet capital of each word in a given string.

44.Write a program to Convert a character array to string.

45.Write a program to String concatenation with primitive data type values.

46.Write a program to Check given strings are Anagram or not.

47.Write a program to separate all tokens (words) using StringTokenizer.

48.WAP to check whether a character is an alphabet or not.

49.WAP to find the ASCII value of a character.

50.WAP to Toggle each character in a string.

51.WAP to count the number of vowels in a String.

52.WAP to Remove the vowels from a String.

53.WAP to Remove all characters from string except alphabets.

54.WAP to remove brackets from an algebraic expression.

55.WAP capitalise the first and last character of each word of a string.

56.WAP to check frequency of characters in a string.

57.WAP to find the total number of alphabets, digits or special characters in a string.

58.WAP to find out the total occurrence of each letter in a string.

Sample Input: “aabbccddd”

Output: a- 2 times

b- 2 times

c- 2 times

d- 3 times

59. WAP for Replacing a particular word with another word in a string.

60.WAP to Count common subsequence in two strings.

Lets understand with the help of an example.

m= “ABC”

n= “AB”

The subsequence of m = A,B,C, AB, BC, AC, ABC and n=A,B,AB

The count of common subsequences is 3 that is (A,B AB)

61.Write a program to check if one string is a rotation of another.

62.Write a program to check if a given string represents a valid number.

63.Write a program to find the length of the longest substring without repeating characters.

64.WAP checks if a String contains only digits.

65.Write a java program to find all possible palindromes of a given String.

Sample Input: “aaabbbacccababacccaabb”.

66. WAP to find words of maximum length in a given String.

Sample Input: “Dear Student,you have need to work hard”

Output: “Student”.

67.WAP to remove the first occurrence of a word from a string.

68.WAP to remove all occurrences of a word in a given string.

69.WAP to search all occurrences of a word in a given string.

Input string: I love programming. I love Java.

Input word to search: love

Output

'love' is found at index: 2
'love' is found at index: 22

70.Write a function to pad a given string with a specific character to a certain length.

Output:

Original String: Hello
Padded String: Hello*****

71.WAP to remove the last occurrence of a character from a string.
Input string : I love programming. I love Java.
Input character to remove : 'I'

Output

String after removing the last 'I' : I love programming. i love Java.

72.WAP to replace all occurrences of a character with another in a string.

73.WAP to replace the first occurrence of a character with another in a string.

74.WAP to replace last occurrence of a character with another in a string

75.WAP to find the first occurrence of a word in a given string.

76.WAP to find the last occurrence of a word in a given string.

77.Write a C program to find the highest frequency character in a string.

78..Write a C program to find the lowest frequency character in a string.

79.WAP to count occurrences of a word in a string.

Input string: I love programming. I love Java.
Input word to search: love

Output

Total occurrences of 'love': 2

80.Check if two strings match where one string contains wildcard characters

Output:

First string with wild characters : Prepins*a Second string without wild characters :: Prepinsta
True

81.WAP to convert a given string to a title case using the title() method.
sentence = "this is a sample sentence"
Output:
"This Is A Sample Sentence"

82.WAP to Check if a given string starts with a specific prefix using the startswith() method.

83.WAP to Check if a given string ends with a specific suffix using the endswith() method.

84.WAP to Find the index of the first occurrence of a substring in a given string using the find() method.

85.WAP to Find the index of the first occurrence of a specific character in a given string using the index() method

86.WAP to check if a given string consists only of alphabetic characters using the isalpha() method.

87.WAP to check if a given string consists only of digits using the isdigit() method.

88.WAP to check if a given string consists only of whitespace characters using the isspace() method.

89.WAP to Remove leading and trailing whitespaces from a given string using the strip() method.

```
padded_string = "   hello   "
stripped_string = padded_string.strip()
print(stripped_string)
```

Output:
"hello"

90.WAP to Split a given sentence into a list of words using the split() method.

91.WAP to Join a list of words into a single string using the join() method.

92.WAP to Check if a given string starts with "Hello" and ends with "World" using both startswith() and endswith() methods.

93.WAP to Swap the case of each character in a given string using the swapcase() method.

mixed_case_string = "Hello World"
Output:
"hELLO wORLD"

94.What will be the Output of the following code:

```
class string_class      {
    public static void main(String args[])        {
          String obj = "hello";
          String obj1 = "world";
            String obj2 = obj;
           obj2 = " world";
            System.out.println(obj + " " + obj2);
 }
}
```

95.What will be the Output of the following code:

```
public class Java
{
   public static void main(String[] args)
   {
      String str1 = "Java";

      String str2 = "Java";

      System.out.println(str1 == str2);

      System.out.println(str1.equals(str2));
```

```
        System.out.println(str1.hashCode() == str2.hashCode());
    }
}
```

96.What will be the Output of the following code:

```
public class Java
{
    public static void main(String[] args)
    {
        String s1 = "ONE";

        s1.concat("TWO");

        s1.concat("THREE");

        System.out.println(s1);
    }
}
```

97.What will be the Output of the following code:

```
public class Java  {
        public static void main(String[] args)      {
            String str1 = "Java J2EE Spring Hibernate SQL";
            String str2 = "Python Java Scala C C++";
System.out.println(str1.contains("HTML") == str2.contains("HTML"));
 }
}
```

98.What will be the Output of the following code:

```
public class Java
{
    public static void main(String[] args)
    {
        String s1 = 1+null"null";
```

```
        System.out.println(s1);
    }
}
```

99.Find the output of the following code.

```
int Integer = 24;
char String  = 'I';
System.out.print(Integer);
System.out.print(String);
```

100.Find the output of the following code.

```
Public class Test{
        Public static void main(String argos[]){
                String str1 = "one";
                String str2 = "two";
                System.out.println(str1.concat(str2));
        }
}
```

101.How many objects will be created in the following?

```
String a = new String("Interviewbit");
String b = new String("Interviewbit");
Strinc c = "Interviewbit";
String d = "Interviewbit";
```

102.Find the output of the following code.

```
 public class Test {
      public static void main(String[] args) {
       String s = new String("5");
       System.out.println(1 + 10 + s + 1 + 10);
```

```
    }
}
```

103.Find the output of the following code.

```
public class Test {

    public static void main(String[] args) {
        String s1 = null;
        System.out.println(s1);
        System.out.println(s1.toString());
    }
}
```

104.Find the output of the following code.

```
public class Test {

    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = new String("hello");

        s2 = s2.intern();
        System.out.println(s1 == s2);
    }
}
```

105.Find the output of the following code.

```
public class Test {

    public static void main(String[] args) {
        String s1 = "abc";
```

```
        StringBuffer s2 = new StringBuffer(s1);
        System.out.println(s1.equals(s2));
    }
}
```

106.Find the output of the following code.

```
public class Main{
    public static void main(String []args){
        String str1 = "Java";
        String str2 = "Java";
        String str3 = new String("Java");

        System.out.println(str1.equals(str2) && str1.equals(str3));
    }
}
```

107.Find the output of the following code.

```
class A {
  public static void main(String args[]){
  String s1 = new String("digitalocean");
                String s2 = new String("DIGITALOCEAN");
                System.out.println(s1 = s2);
  }
 }
```

108.Find the output of the following code.

```
public class JavaStringsQuiz
{
    public static void main(String[] args)
    {
        String str1 = "Java";

        String str2 = "Java";

        System.out.println(str1 == str2);

        System.out.println(str1.equals(str2));

        System.out.println(str1.hashCode() == str2.hashCode());
    }
}
```

109.Find the output of the following code.

```
public class Java
{
    public static void main(String[] args)
    {
        System.out.println("JAVAJ2EE".substring(2, 5).substring(1).charAt(2));
    }
}
```

110. Find the output of the following code.

```
public class Java
{
    public static void main(String[] args)
    {
        System.out.println("Java Concept Of The Day".substring(8, 4));
    }
```

}

Exception Handling

1) What is exception ?

2) Why we handle the exception ?

3) What is error ?

4) What is Compile time (Checked) exception ?

5) What is Runtime (Unchecked) exception ?

6) What is Fully checked exception ?

7) What is the use case of throw keyword ?

8) What is the use case of throws keyword ?

9) What is the use case of finally block ?

10) What is try with resorce ?

11) What is exception prapogation ?

12) Difference between exception and error ?

13) Difference between checked and unchecked exception ?

14) Difference between throw and throws keyword ?

15) Difference between parsally checked and fullychecked exception ?

16) Difference between final keyword finally block and finalize method ?

17) Difference between e.printStackTrace and System.out.println(e) ?

18) Any 5 compile time exception ?

19) Any 5 runtime exception ?

20) Parent of exception class ?

21) Can we create Checked exception ?

22) Can we create Unchecked exception ?

23) How to create checked and unchecked exception ?

24) At what time exception occure ?

25) Which type of code I write inside the try block ?

26) Can I write multiple catch blocks in respect of one try block?

27) Can I write multiple finally block in respect of one try block ?

28) Can I write multiple exception class in one catch block ?

29) Can I write try block without catch or finally ?

30) Can I write catch block without try or finally ?

31) Can I write finally block without try of catch ?

32) Can I give same exception class in one catch block ?

33) Can I change the sequence of try catch and finally block ?

34) Can I propogate multiple exception class ?

35) Can I throw the object ?

36) Can I throw the throable ?

37) Can I throw the exception ?

38) Can I throw the error ?

39) Can I catch the object ?

40) Can I catch the throable ?

41) Can I catch the exception ?

42) Can I catch the error ?

43) Can I throw the same exception class ?

44) Can I throw the exception at the time of method overriding ?

45) Can I change the exception at the time of method overriding ?

46) Can I throw the checked and unchecked exception ?

47) In which case finally block will not execute ?

48) At what time I use throw keyword ?

49) At what time I use throws keyword ?

50) Which type of exception I throw ?

51) Which type of exception is occure at runtime ?

52) Which type of exception is occure at compiletime ?

53) At what time java compiler throw the classCastException ?

54) At what time java compiler throw the nullPointerException ?

55) At what time java compiler throw the entruptedException?

56) At what time catch block execute ?

57) What is the advantage of exception handling ?

58) Type of error ?

59) Without try catch block can I catch the exception ?

60) How many type of exception are there ?

61) Who is responsible to check the exception ?

62) How to create customize exception ?

63) What is wrong with the following code? Why it is showing compilation error?

```
1   public class JavaExceptionHandlingQuiz
2   {
3       public static void main(String[] args)
4       {
5           try
6           {
7               System.out.println("Try Block");
8           }
9
10          System.out.println("-----");
11
12          catch (Exception e)
13          {
14              System.out.println("Catch Block");
15          }
16      }
17  }
```

64) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        int i = 1;

        try
        {
            i++;
        }
        catch (Exception e)
        {
            i++;
        }
        finally
        {
```

```
            i++;
        }
        System.out.println(i);
    }
}
```

65) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);

            int i = 100 / 0;

            System.out.println(2);
        }
        catch (Exception e)
        {
            System.out.println(3);
        }
    }
}
```

66) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);
        }
```

```
        catch (Exception e)
        {
            System.out.println(2);
        }

        System.out.println(3);

        finally
        {
            System.out.println(4);
        }
    }
}
```

67) Catch block takes one argument of type java.lang.Object. True OR False?

68) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        System.out.println(1);

        try
        {
            System.out.println(2);

            int i = Integer.parseInt("ABC");

            System.out.println(3);
        }
        catch (Exception e)
        {
            System.out.println(4);
```

```
        }
        finally
        {
            System.out.println(5);
        }

        System.out.println(6);
    }
}
```

69) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);

            int i = Integer.parseInt("ABC");

            System.out.println(2);
        }
        catch (NumberFormatException e)
        {
            System.out.println(3);
        }
        catch (Exception e)
        {
            System.out.println(4);
        }
    }
}
```

70) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            int[] a = {1, 2, 3, 4};

            int i = a[4];
        }
        catch (NumberFormatException e)
        {
            System.out.println(1);
        }
        catch (NullPointerException e)
        {
            System.out.println(2);
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println(3);
        }
    }
}
```

71) What will be the outcome of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            String s = null;

            int i = s.length();
```

```
        }
        catch (Exception e)
        {
            System.out.println(1);
        }
        catch (NullPointerException e)
        {
            System.out.println(2);
        }
    }
}
```

72) Pipe (|) operator is introduced in Java 7 to catch the multiple exceptions using single catch block. Yes OR No?

73) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        String[] s = {"abc", "123", null, "xyz"};

        for (int i = 0; i < 6; i++)
        {
            try
            {
                int a = s[i].length() + Integer.parseInt(s[i]);
            }
            catch (NumberFormatException e)
            {
                System.out.println(1);
            }
            catch (NullPointerException e)
            {
```

```
            System.out.println(2);
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println(3);
        }
    }
  }
}
```

74) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            try
            {
                try
                {
                    String s = args[1];
                }
                catch (NullPointerException e)
                {
                    System.out.println(1);
                }
            }
            catch (ArrayIndexOutOfBoundsException e)
            {
                System.out.println(2);
            }
        }
        catch (Exception e)
        {
```

```
            System.out.println(3);
        }
    }
}
```

75) Does the following code compile without errors?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("try Block");
        }
        finally
        {
            System.out.println("finally Block");
        }
    }
}
```

76) Does the following code compile without errors?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("try Block");
        }
    }
}
```

77) What is wrong with the following code? Why it is showing compilation error?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println(1);
        }
        catch (Exception | NumberFormatException | NullPointerException e)
        {
            System.out.println(2);
        }
    }
}
```

78) java.lang.Throwable is the super class for all type of errors and exceptions in Java. True OR False?

79) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            main(args);
        }
        catch (NumberFormatException | NullPointerException e)
        {
            System.out.println(1);
        }
        catch (Exception | Error e)
        {
            System.out.println(2);
        }
    }
```

```
}
```

80) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static int anyMethod()
    {
        try
        {
            return 10;
        }
        catch (Exception e)
        {
            return 20;
        }
        finally
        {
            return 30;
        }
    }

    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

81) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static int anyMethod()
    {
        int i = 1;

        try
        {
```

```
            i = i/0;

            return ++i;
        }
        catch (Exception e)
        {
            return ++i;
        }
        finally
        {
            return ++i;
        }
    }

    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

82) In the following program, finally block is executed or not?

```
public class JavaExceptionHandlingQuiz
{
    static void anyMethod()
    {
        try
        {
            return;
        }
        catch (Exception e)
        {
            return;
        }
        finally
        {
            System.out.println("finally Block");
```

```
        }
    }

    public static void main(String[] args)
    {
        anyMethod();
    }
}
```

83) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static String anyMethod()
    {
        String s = "ZERO";

        try
        {
            s = s + "ONE";

            return s;
        }
        catch (Exception e)
        {
            s = s + "TWO";

            return s;
        }
        finally
        {
            s = s + "THREE";

            return s;
        }
    }
```

```
public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

84) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static String anyMethod()
    {
        String s = "ONE";

        try
        {
            s = s + "TWO";

            return s;
        }
        catch (Exception e)
        {
            s = s + "THREE";

            return s;
        }
        finally
        {
            s = s + "FOUR";
        }
    }

    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
```

```
}
```

85) Does the following code compile without errors?

```
public class JavaExceptionHandlingQuiz
{
    static void anyMethod()
    {
        try
        {
            System.out.println("Try Block");
        }
        catch (Exception e)
        {
            System.out.println("Catch Block");
        }
        finally
        {
            return;
        }

        System.out.println("Any Statements");
    }

    public static void main(String[] args)
    {
        anyMethod();
    }
}
```

86) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static int anyMethod()
    {
        int i = 1;
```

```
        try
        {
            i = i + 1;

            return i;
        }
        catch (Exception e)
        {
            i = i + 2;
        }
        finally
        {
            i = i + 3;
        }

        return i;
    }

    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

87) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static int anyMethod()
    {
        int i = 1;

        try
        {
            i = i / 0;
```

```
            return i;
        }
        catch (Exception e)
        {
            i = i + 1;

            return i;
        }
        finally
        {
            i = i + 2;

            System.out.println(i);
        }
    }
    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

88) What will be the outcome of the following program?

```
public class JavaExceptionHandlingQuiz
{
    static int anyMethod()
    {
        int i = 1;

        try
        {
            i = i + 1;

            return i;
        }
        catch (Exception e)
```

```
        {
            i = i + 2;
        }
        finally
        {
            i = i + 3;
        }

        System.out.println(i);
    }

    public static void main(String[] args)
    {
        System.out.println(anyMethod());
    }
}
```

89) Checked exceptions are checked at compile time. True OR False?

90) What are the two sub classes of java.lang.Throwable?

91) What will be the outcome of the following code?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            throw new NullPointerException("My_Exception");
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

92) What will be the output of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        try
        {
            try
            {
                NumberFormatException ex = new NumberFormatException();

                throw ex;
            }
            catch (NullPointerException ex)
            {
                System.out.println(1);
            }
        }
        catch (Exception ex)
        {
            System.out.println(2);
        }
        catch (Throwable ex)
        {
            System.out.println(3);
        }
    }
}
```

93) What will be the outcome of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args) throws Exception
```

```
    {
        try
        {
            try
            {
                try
                {
                    throw new Exception();
                }
                catch (Exception ex)
                {
                    throw ex;
                }
            }
            catch (Exception ex)
            {
                throw ex;
            }
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```

94) StackOverflowError occurs only at run time. Yes OR No?

95) Does the following code compile without errors?

```
class SuperClass
{
    void anyMethod()
    {
```

```
        System.out.println("Super Class Method");
    }
}

class SubClass extends SuperClass
{
    @Override
    void anyMethod() throws IOException
    {
        System.out.println("Sub Class Method");
    }
}
```

96) Does the following code compile successfully?

```
class SuperClass
{
    void anyMethod() throws NullPointerException
    {
        System.out.println("Super Class Method");
    }
}

class SubClass extends SuperClass
{
    @Override
void anyMethod() throws ArrayIndexOutOfBoundsException,
NumberFormatException, ClassCastException{

System.out.println("Sub Class Method");

    }
}
```

97) Does the following code compile without errors?

```
class SuperClass
```

```
{
    void anyMethod() throws IOException
    {
        System.out.println("Super Class Method");
    }
}

class SubClass extends SuperClass
{
    @Override
    void anyMethod() throws Exception
    {
        System.out.println("Sub CLass Method");
    }
}
```

98) Does the following code compile successfully?

```
class SuperClass
{
    void anyMethod() throws IOException
    {
        System.out.println("Super Class Method");
    }
}

class SubClass extends SuperClass
{
    @Override
    void anyMethod() throws FileNotFoundException
    {
        System.out.println("Sub CLass Method");
    }
}
```

99) Does the following code compile successfully?

```
class SuperClass
{
    void anyMethod() throws ClassNotFoundException
    {
        System.out.println("Super Class Method");
    }
}

class SubClass extends SuperClass
{
    @Override
    void anyMethod() throws FileNotFoundException
    {
        System.out.println("Sub CLass Method");
    }
}
```

100) InterruptedException is a checked exception. Yes OR No?

101) Does the following code compile successfully?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        Class.forName("AnyClassName");
    }
}
```

102) Unchecked exceptions are also called run time exceptions. True OR False?

103) OutOfMemoryError – is it an exception or an error?

104) What will be the outcome of the following program?

```
public class JavaExceptionHandlingQuiz
{
    public static void main(String[] args)
    {
        FileInputStream fis = new FileInputStream("My_File");
        fis.close();
    }
}
```

105) NullPointerException is a checked exception. True OR False?

106) What happens when you run the below program?

```
import java.io.IOException;
import java.io.FileNotFoundException;

import javax.xml.bind.JAXBException;

public class TestExceptionDemo {

        public static void main(String[] args) {
          try {
              demoException();
          } catch (IOException e) {
              System.out.println(e.getMessage());
          } catch (FileNotFoundException e) {
              System.out.println(e.getMessage());
          } catch (JAXBException e) {
              System.out.println(e.getMessage());
          }
        }

        public static void demoException() throws IOException,
        FileNotFoundException,
```

```
        JAXBException{

        }
}
```

107) What happens when the below program is run?

```
public class TestExceptionDemo {

        public static void main(String[] args) {
                try{
                   demoMethod();
                }catch(NullPointerException e){
                        System.out.println(e.getMessage());
                }catch(Exception e){
                        System.out.println(e.getMessage());
                }

                foobar();
        }

        public static void demoMethod(){

        }

        public static void foobar() throws NullPointerException{

        }
}
```

108) Is it possible to throw an Exception inside a Lambda Expression's body?

109)What is the output of this below program?

```
public class TestClass
{
   public static void main(String[] args)
   {
```

```
        int a = 30;
        int b = 40;
        int c = 10;
        int expression = (a * b)/(a - b + c);
        System.out.println("Result: " +expression);
      }
}
```

110) How to catch multiple exception in a single catch block ?

MULTITHREADING

1) What Is Multi Tasking?

2) What Is Advantage Of Multi Threading?

3) In How Many Ways We Can Define A Thread with example?

4) Among Extending Thread And Implementing Runnable Which Approach Is Recommended ?

5) Difference Between t.start() And t.run()?

6) Explain About Thread Scheduler?

7) If We Are Not Overriding run() What Will Happen?

8) Is It Possible Overloading Of run()?

9) Is It Possible To Override a start() And What Will Happen?

10) Explain Life Cycle Of A Thread?

11) What Is The Importance Of Thread Class start()?

12) After Starting A Thread If We Try To Restart The Same Thread Once Again What Will happened ?

13) Explain Thread Class Constructors?

14) How To Get And Set Name Of A Thread with example?

15) Who Uses Thread Priorities with example ?

16) Default Priority For Main Thread?

17) Once We Create A New Thread What Is Its Priority?

18) How To Get Priority From Thread And Set Priority To A Thread?

19) If We Are Trying To Set Priority Of Thread As 100, What Will Happen?

20) If 2 Threads Having Different Priority Then Which Thread Will Get Chance First For Execution?

21) If 2 Threads Having Same Priority Then Which Thread Will Get Chance First For Execution?

22) What Is yield() And Explain Its Purpose?

23) IsJoin Is Overloaded?

24) Purpose Of sleep() with program?

25) What Is synchronized Key Word? Explain Its Advantages And Disadvantages?

26) What Is Object Lock explain with example?

27) What Is A Class Level Lock explain with example?

28) Difference Between Synchronized Method And Static Synchronized Method?

29) How 2 Threads Will Communicate With Each Other?

30) wait(), notify(), notifyAll() Are Available In Which Class?

31) Why wait(), notify(), notifyAll() Methods Are Defined In Object Instead Of Thread Class?

32) Without Having The Lock Is It Possible To Call wait()?

33) How A Thread Can Interrupt Another Thread?

34) What Is Deadlock? Is It Possible To Resolve Deadlock Situation?

35) How We Can Stop A Thread Explicitly?

36) What Is Daemon Thread? Give An Example Purpose Of Daemon Thread?

37) Explain About ThreadGroup?

38) What Is ThreadLocal?

39) What is the default name of the thread?

40) How can we get thread name?

41) What are the different ways we can change the default name of the thread?

41) Can we change name & priority of thread after it is started?

42) What is the default group?

43) Synchronized keyword either applied to method or object?

44) What is the difference between thread based and process based ?

45) What is starvation with program?

46) How many types of thread in java ?

47) What is the interthread communication ?

48) What is inconsistency problem write a program?

49) What is contact switching ?

50) How will notify the thread ?

51) Can you start a thread twice?

52) What are the two ways of implementing thread in Java with program?

53) Write a program for class lock in multithreading?

54) Write a program for object lock in multithreading?

55) How can we create daemon threads?

56) What are the wait() and sleep() methods write a program?

57) Can you start a thread twice?

Example:

```
public class TestThreadTwice1 extends Thread{

public void run(){

System.out.println(" thread is executing now.....");

}

public static void main(String args[]){

TestThreadTwice1 t1=new TestThreadTwice1();

t1.start();

t1.start();

}

}
```

58) What will happen if we don’t override the thread class run() method?

Example:

```
class MyThread extends Thread {
//don't override run() method
}
public class DontOverrideRun {
public static void main(String[] args) {
System.out.println("Started Main.");
MyThread thread1=new MyThread();
thread1.start();
System.out.println("Ended Main.");
}
}
```

59) Predict the output of the following code:

```
class MyThread extends Thread
```

```
{
public void run()
{
System.out.println("Running");
}
}
class ThreadTest {
public static void main(String args[]) throws InterruptedException
{
Runnable r = new MyThread(); // #1
Thread myThread = new Thread(r); // #2
myThread.start();
}
}
```

60) In the following java program, what is the name of the thread?

```
class multithreaded_programing
{
public static void main(String args[])
{
Thread t = Thread.currentThread();
System.out.println(t);
}
}
```

61) What do you understand about Deadlock situations in Java Multithreading?write a program.

62) How is thread-safety achieved in multithreaded programming?write a program.

63) What is thread priority?write a program.

64) How to Make User Thread to Daemon Thread?Explain with program.

65) Can we Overload run() method? What if we do not override the run() method? Explain with program.

66) Can we Overload run() method? What if we do not override the run() method? Explain with program.

67)What do you understand by inter-communication?Explain with program?

68) Define BlockingQueue.Write aprogram?

69) What do you mean by shutdown hook?Write a program?

70) Define with program ThreadLocal variable in Java?

I/O questions..

1. Write a code to get a file object.
2. Write a code to store data as object .
3. Write a code to store data as characters.
4. Write a code to read object from file.
5. Write a code to read character from file.
6. Write a code to delete a file.
7. Write a code to rename a file.
8. Write a code to create a directory.
9. Write a code to read directory using array or arrayList.
10. Write a code to copy a file in java.
11. Write a code to read character from file.
12. Create a code to generate file url in java.
13. Write a code to read the contents of a file line by line in Java.
14. Write a code to get size for a file in java.
15. Give an example or write a code to explain file permission concept in java.
16. Create a Java program that reads the first 100 bytes from a `RandonAccessFile` and prints the content to the console.
17. Develop a Java program that appends data at specific positions within a `RandonAccessFile`.
18. Write a code to handle exceptions in file handling in Java?
19. Write a code to check for a hidden file in java.
20. Create a Java program that reads an image file, applies a simple transformation (e.g., grayscale conversion), and saves the transformed image to a new file.
21. Develop a Java program that categorizes files in a directory based on their sizes (small, medium, large) and displays a summary.
22. Create a Java program that performs multiple file operations (e.g., copy, delete) as a single transaction, rolling back all changes if any operation fails.
23. Implement a Java program that creates a backup of a specified directory, including all its files and subdirectories, preserving the directory structure.
24. Implement a Java program to copy a large file from one location to another and display the progress percentage during the copy operation.
25. Write a Java program that searches for a specific keyword in all text files within a given directory and its subdirectories.
26. Implement a Java program that merges the content of multiple text files into a single file.
27. Write a Java program that retrieves and displays metadata (such as size, last modified time, and permissions) for a given file.
28. Develop a Java program that calculates the compression ratio of a compressed file compared to its original size.

29. Implement a Java program that creates a backup of a specified directory, including all its files and subdirectories, preserving the directory structure.
30. Write a Java program that reads a text file containing names and sorts them alphabetically, then saves the sorted names to a new file.
31. Implement a Java program that uses multiple threads to process different sections of a large file concurrently, performing a specific operation on each section.
32. Create a singleton class and serialize its instance. Ensure that deserialization returns the same instance.

    Additional Requirements:
    -Implement proper handling to ensure only one instance is created.
33. Create a class with transient fields. Implement serialization and deserialization, ensuring that transient fields are properly handled.

    Additional Requirements:
    -Use the transient keyword for specific fields.
    -Provide default values for transient fields during deserialization.
34. Create a program that serializes and deserializes a collection of objects (e.g., a list of Person objects). Save the serialized data to a file and then load it back.

    Additional Requirements:

    Use a collection class (e.g., ArrayList) to store multiple objects.
    Handle potential exceptions properly.
35. Modify the Person class from the previous question to implement custom serialization. Encrypt sensitive information (e.g., address) during serialization and decrypt during deserialization.

    Additional Requirements:

    Implement custom writeObject and readObject methods.
    Use a symmetric encryption algorithm (e.g., AES) for encryption and decryption.
36. Create a Person class with attributes such as name, age, and address. Implement serialization and deserialization methods to save and load Person objects to and from a file.

    Additional Requirements:

    Use the Serializable interface.
    Handle potential exceptions properly.

Collection

1. Write a Java program that takes a paragraph of text as input and prints the frequency of each unique word. Treat uppercase and lowercase characters as the same and ignore punctuation. Use a suitable collection from the Java Collections Framework to store and manipulate the data.

Example:

Input:

"Java is a powerful and versatile programming language. Java is used for web development, mobile applications, and enterprise systems."

Output:

java: 2

is: 2

a: 1

powerful: 1

and: 2

versatile: 1

programming: 1

language: 1

used: 1

for: 1

web: 1

development: 1

mobile: 1

applications: 1

enterprise: 1

systems: 1

Constraints:

- The input paragraph will not be empty.
- Words are separated by spaces.
- Punctuation marks (.,;:'" etc.) should be ignored.

Explanation:

- In the given example, the word "Java" appears twice, "is" appears twice, and the other words appear once each.

Hints:

- Use a HashMap or TreeMap to store the word frequencies.
- Consider using toLowerCase() to handle case-insensitivity.
- Use String.split(" ") to split the paragraph into words.

2. Write a Java program that takes an array of integers as input and prints the count of unique elements in the array. Use a suitable collection from the Java Collections Framework.
3. Write a Java program that takes an array of strings as input and prints the first duplicate element. Use a suitable collection from the Java Collections Framework.
4. Implement a basic stack in Java using the Java Collections Framework. The stack should support push, pop, and peek operations.
5. Write a Java program that takes two arrays of integers as input and prints the union and intersection of the two sets. Use suitable collections from the Java Collections Framework.
6. Implement a program that takes a string as input and prints the frequency of each character in the string using a **`HashMap`**.
7. Implement a custom iterator in Java that iterates over even numbers in an ArrayList of integers. The iterator should skip odd numbers.
8. Write a Java program that uses an **`Enumeration`** to iterate over elements in a **`Vector`** and print each element.
9. Write a Java program that uses a **`ListIterator`** to traverse a LinkedList in both forward and backward directions, printing each element.
10. Write a Java program that uses a **`ListIterator`** to traverse an **`ArrayList`** of strings. Remove all elements that start with the letter "A" while iterating.
11. Write a Java program that merges two sorted lists into a new sorted list using **`ListIterator`**. Assume the input lists are already sorted.
12. Write a Java program that uses an iterator to filter out elements greater than a specified threshold from a collection.

13. Implement a program to demonstrate the usage of ArrayList.
    Write a method to reverse a LinkedList.
14. Create a program to remove duplicates from an ArrayList.
    Implement a stack using the ArrayList class.

15. Write a program to find the intersection of two lists.

16. Develop a method to check if a list is palindrome or not.

17. Implement a program to shuffle elements in an ArrayList.

18. Create a method to find the second largest element in a list.

19. Write a program to remove all occurrences of a specific element from an ArrayList.

20. Implement a program to sort a list of custom objects.

21. Create a program to demonstrate the usage of HashSet.

22. Write a method to find the union of two sets.

23. Implement a program to check if two sets are equal.

24. Develop a method to remove all elements from a set.

25. Write a program to find the common elements between two sets.

26. Create a method to find the difference between two sets.

27. Implement a program to remove null values from a HashSet.

28. Write a program to convert a set to an array.

29. Develop a method to check if a set is a subset of another set.

30. Create a program to remove elements from a HashSet while iterating.
31. Implement a program to demonstrate the usage of HashMap.

32. Write a method to check if a key exists in a HashMap.
33. Create a program to find the intersection of two maps.
34. Develop a method to get all keys from a HashMap.
35. Write a program to sort a HashMap by keys.
36. Implement a program to merge two maps.
37. Create a method to get the most occurring element in a HashMap.
38. Write a program to find the difference between two maps.
39. Develop a program to check if two maps have the same values.
40. Implement a program to iterate over a HashMap using entrySet.
41. Implement a program to reverse a List using Collections.reverse().
42. Write a method to shuffle elements in a List using Collections.shuffle().
43. Develop a program to rotate elements in a List using Collections.rotate().
44. Create a method to find the frequency of an element in a List using Collections.frequency().
45. Implement a program to create an immutable list using Collections.unmodifiableList().
46. Write a program to find the minimum element in a Collection using Collections.min().
47. Develop a method to create a synchronized list using Collections.synchronizedList().
48. Create a program to reverse a List using Collections.swap().
49. Write a program to find the median of a List using Collections.sort() and Collections.binarySearch().
50. Implement a program to fill elements in a List using Collections.fill().
51. Write a program to sort a List of custom objects using Comparable.

52. Implement a method to sort a List of custom objects using Comparator.

53. Create a program to perform binary search on a sorted List.

54. Develop a method to find the index of the first occurrence of an element in a sorted List.

55. Write a program to shuffle and then sort a List using Collections.shuffle() and Collections.sort().

56. Implement a program to reverse order the elements in a List after sorting.

57. Create a method to perform a case-insensitive search in a sorted List.

58. Write a program to sort a List of strings by their lengths.
59. Implement a program to find the minimum and maximum elements in a List using Collections.min() and Collections.max().
60. Write a Java program to serialize and deserialize a binary tree.
61. Wap for generic class in java.
62. Create a method calculateAverage that takes an array of Integer objects and calculates the average of the values. The method should return the result as a Double object.
63. Create a method findMaxValue that takes an array of Double objects and finds and returns the maximum value.
64. Implement a method convertToIntArray that takes an array of Integer objects and converts it to a primitive int array. Return the resulting primitive array.
65. Write a main method to demonstrate the functionality of the above methods. Create sample arrays, call the methods, and print the results.

JDBC Questions---

1. Write Java code to establish a connection to a MySQL database using JDBC. Handle potential exceptions appropriately.
2. Write Java code to execute a simple SQL query (e.g., SELECT statement) using JDBC. Print the results.
3. Use a `PreparedStatement` to insert data into a database table using JDBC. Handle potential exceptions.

4. Write Java code to perform batch processing using JDBC. Insert multiple records into a database table in a single batch.
5. Implement Java code to demonstrate transaction handling in JDBC. Begin a transaction, execute multiple SQL statements, and commit the transaction. Rollback the transaction if an exception occurs.
6. Implement Java code to retrieve database metadata using JDBC. Print information such as database name, table names, and column details.
7. Write Java code to retrieve and print metadata information (e.g., column names and types) from a **`ResultSet`** using **`ResultSetMetaData`**.
8. Demonstrate the use of a scrollable **`ResultSet`** in Java code. Traverse through the result set in both forward and backward directions.
9. Explain the concepts of result set concurrency and updatability in JDBC. Provide Java code to set and use different concurrency and updatability modes.
10. Write a Java program that demonstrates the usage of three constants from the **`ResultSet`** interface: **`TYPE_SCROLL_INSENSITIVE`**, **`CONCUR_UPDATABLE`**, and **`HOLD_CURSORS_OVER_COMMIT`**. Connect to a database, execute a SELECT query to retrieve data, and use these constants to manipulate the ResultSet in different ways.

JAVASCRIPT

Que 1.) Write a function that takes a sentence as input and returns the longest word in it.

Que 2.) Write a Javascript function to count the number of vowels in a given string.

Que 3.) Write a program that prints the numbers from 1 to 100. But for multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz." For numbers that are multiples of both three and five, print "FizzBuzz."

Que 4.) Write a function that splits an array into parts of a specified size.

Que 5.) Write a function that returns an array containing the common elements between two arrays.

Que 6.) Write a function to validate whether a given string is a valid email address.

Que 7.) Write a function that returns the sum of all prime numbers less than or equal to a given number.

Que 8.) Write a function to calculate the power of a number raised to an exponent (without using ** operator or any function).

Que 9.) Write a function that removes falsy values (false, null, 0, "", undefined, and NaN) from an array.

Que 10.) Write a function that finds the intersection of two arrays.

Que 11.) Write a function to determine if a string has all unique characters.

Que 12.) Write a function to check if one array is a subset of another.

Que 13.) Write a function to count the number of consonants in a string.

Que 14.) Write a function to change the text content of an HTML element.

Que 15.) Write a function to add or remove a CSS class to an HTML element with the help of a single button.

Que 16.) Write a function to set or remove an attribute on an HTML element.

Que 17.) Write a function to count the number of child elements of a given element.

Que 18.) Create a function to change the source of an image element.

Que 19.) Create a function to disable a button.

Que 20.) Write a function that takes a callback as an argument and invokes it.

Que 21.) Create a function that takes two numbers and a callback. The callback should perform an Arithematic operation on the numbers.

Que 22.) Write a function that takes a callback and a delay time. The callback should be executed after the specified delay.

Que 23.) Implement a chain of callbacks where each callback depends on the result of the previous one.

Que 24.) Implement a callback function to handle a button click event.

Que 25.) Write a function that takes a callback and handles errors by passing an error parameter.

Que 26.) Create a function that simulates an asynchronous operation and takes a callback to handle the result.

Que 27.) Write a function that animates an element and takes a callback to execute after the animation is complete.

Que 28.) Create a function that takes a callback and conditionally executes it based on a given condition.

Que 29.) Implement a function that takes a callback and a timeout, throwing an error if the callback takes longer than the specified time.

Que 30.) Write a function that uses promises to replace nested callbacks.

Que 31.) Implement a sequence of asynchronous operations using callbacks without falling into callback hell.

Que 32.) Write a function that performs multiple asynchronous operations in parallel using callbacks.

Que 33.) Write a program with a callback pyramid structure.

Que 34.) Chain multiple promises to perform sequential asynchronous operations.

Que 35.) Convert a set of callback-based functions to use promises instead.

Que 36.) Write a function that returns a simple promise. Resolve the promise after a short timeout and log a success message.

Que 37.) Write a function that returns a promise. Reject the promise after a short timeout and log an error messageQue 38.) Create two functions that return promises. Chain them so that the second promise runs after the first one is resolved. Log the result of the second promise.

Que 39.) Write a function that returns a promise. Reject the promise if a certain condition is met, and handle the rejection by logging an error message.

Que 40.) Create a function that takes a boolean parameter. Return a promise that resolves if the parameter is true and rejects if it's false.

Que 41.) Write a function that returns a promise. Resolve the promise after a short timeout, and reject it if it takes too long. Use Promise.race.

Que 42.) Create two functions that return promises with different delays. Use Promise.all to log the results when both promises are resolved.

Que 43.) Write an async function that returns a promise. Use await within the function to wait for the resolution of another promise.

Que 44.) Write an async function that returns a promise. Inside the function, use try/catch to handle errors and log appropriate messages.

Que 45.) Write a function that returns a promise. Chain multiple promises to simulate sequential execution. Log the final result.

Que 46.) Write a function that uses Promise.race to handle a set of promises with different timeouts.

Que 47.) Implement a function that takes an array of promises and logs their results after they are all settled.

Que 48.) Create a function that returns a promise and resolves or rejects after a specified custom delay.

Que 49.) Write a function that returns a promise. If the promise is rejected, retry it up to a certain number of attempts before giving up.

Que 50.) Implement a function that returns a promise. Chain additional promises with error handling using catch.

Que 51.) Write a function that returns a promise and retries on rejection with a specified timeout between retries.

Que 52.) Write an async function that awaits multiple promises running in parallel with error handling.

Que 53.) Implement a function that returns a promise and resolves multiple times after different delays.

Que 54.) Implement a curried function curryAverage that calculates the average of an array.

Que 55.) Write a simple try-catch block that catches an error and logs a custom error message.

Que 56.) Create a function that throws a custom error, and then use a try-catch block to catch and log the error (Javascript).

Que 57.) Create a function that throws an error and another function that calls the first one. Implement error propagation using try-catch (Javascript).

Que 58.) Define a custom error class and throw an instance of that class. Use a try-catch block to catch and log the custom error (Javascript).

Que 59.) Create an async function that uses await to handle a promise rejection. Log an error message if the promise is rejected.

Que 60.) Explain the difference between innerText and innerHTML with practical example.

Que 61.) Rewrite the following function using arrow function syntax:

```
function add(a, b) {
  return a + b;
}
```

Que 62.) Create a function counter that returns a counter, and each time it is invoked, it increments the counter. Demonstrate the closure property.

Que 63.) Write a function calculate that takes two numbers and a callback function as parameters. The callback function should perform a specific operation (e.g., add, subtract). Use the callback to calculate the result.

Que 64.) Create a function greet that takes a name and a greeting message as parameters. Provide a default greeting message if none is specified.

Que 65.) Define an object calculator with methods for basic arithmetic operations (add, subtract, multiply, divide).

Que 66.) Create a prototype Person with properties name and age. Instantiate two objects using this prototype.

Que 67.) Extend the Person prototype to create a Student prototype with additional properties (e.g., grade, school). Instantiate a Student object.

Que 69.) Write a function printPerson that takes a person object (with properties like name and age) as a parameter and logs their details.

Que 70.) Create a function createCar that takes parameters for make, model, and year, and returns an object representing a car.

Que 71.) Create a function that dynamically adds a new list item to an existing unordered list when a button is clicked.

Que 72.)Build a simple image gallery where clicking on an image displays it in a larger view.

Que 73.) Create a countdown timer that decrements every second and displays the remaining time.

Que 74.) Implement a function to dynamically load content onto a page without refreshing when a button is clicked.

Que 75.) Build a responsive navigation menu that changes its layout based on the screen size.

Que 76.) Develop a color picker that updates the background color of an element as the user selects a color.

Que 77.) Build a dynamic form that adds new input fields when a button is pressed.

Que 78.) Implement a dark mode toggle button that changes the entire page's color scheme.

Que 79.) Create a countdown timer that updates every second using setInterval.

Que 80.) Build a function that prints "Hello" every 3 seconds using setInterval.

Que 81.) Implement a delayed alert using setTimeout.

Que 82.) Create a slideshow that changes images every 5 seconds using setInterval.

Que 83.) Create a function that returns the nth Fibonacci number.

Que 84.) Write a function to filter out even numbers from an array.

Que 85.) What will be the output of the following code :

```
let x = '{ "b": 1, "c": 2 }';
let y = JSON.parse(x);
console.log(typeof y);
```

Que 86.) What will be the output of the following code :

```
let x = Infinity;
console.log(typeof x);
```

Que 87.) What will be the output of the following code :

```
let x = "5";
let y = 2;
console.log(x + y);
console.log(x - y);
```

Que 88.) What will be the output of the following code :

```
let x = [2];
let y = 2;
console.log(x == y);
```

Que 89.) What will be the output of the following code :

```
let x = 10;
let y = 20;
console.log("total = " + x + y);
```

Que 90.) What will be the output of the following code :

```
let x = 5;
let y = 6;
```

```
x += x > y ? x : y;
console.log(x);
```

Que 91.) What will be the output of the following code :

```
setTimeout(() => {
console.log(1);
}, 0);
console.log(2);
```

Que 92.) What will be the output of the following code :

```
console.log(   2 + "2" );
console.log( "2" + "2" );
console.log(   2 - "2" );
console.log( "2" - "2" );
console.log( "A" - "A" );
```

Que 93.) What will be the output of the following code :

```
const elements = [1, 2, 3, 4, 5];
elements.forEach(element => {
  console.log(element);
  if(element == 2){
    return;
  }
});
```

Que 94.) What will be the output of the following code :

```
for (var i = 0; i < 3; i++) {
  setTimeout(function() { console.log(i); }, i*1000);
}
```

Que 95.) What will be the output of the following code :

```
let x = 1;
console.log(x + x++);
```

Que 96.) What will be the output of the following code :

```
let a = [1, 2, 3];
let b = [4, 5, 6];
console.log(a + b);
```

Que 97.) What will be the output of the following code :

```
let x = [1, 2, 3, 4];
let [a, ...b] = x.reverse();
console.log(b);
```

Que 98.) What will be the output of the following code :

```
let x = "b";
let y = "a";
console.log(x + y + + y + y);
```

Que 99.) What will be the output of the following code :

```
let x = 7;
let y = !!x && !!!x;
console.log(y);
```

Que 100.)Given a String with multiple special character :

Ex:- "string1@string2#string3$string4"

you'll have to seperate this and generate output :- "string1 string2 string3 string4" using javascript.

Que 101.) What will be the output of the following code snippet?

```
<script type="text/javascript">
a = 5 + "9";
document.write(a);
</script>
```

Que 102.) What will be the output of the following code snippet?

```
<script type="text/javascript" language="javascript">
  var a = "Scaler";
var result = a.substring(2, 4);
document.write(result);
 </script>
```

Que 103.) What will be the output of the following code snippet?

```
<script type="text/javascript" language="javascript">
var x=12;
var y=8;
var res=eval("x+y");
document.write(res);
 </script>
```

Que 104.) What will be the output of the following code snippet?

```
(function(){
 setTimeout(()=> console.log(1),2000);
 console.log(2);
 setTimeout(()=> console.log(3),0);
 console.log(4);
})();
```

Que 105.) What will be the output of the following code snippet?

```
(function(a){
 return (function(){
   console.log(a);
   a = 6;
 })()
})(21);
```

Que 106.) What will be the output of the following code snippet?

```
function solve(arr, rotations){
 if(rotations == 0) return arr;
 for(let i = 0; i < rotations; i++){
   let element = arr.pop();
   arr.unshift(element);
 }
 return arr;
}
solve([44, 1, 22, 111], 5);
```

Que 107.) What will be the output for the following code snippet?

```
<p id="example"></p>
<script>
function Func()
{
document.getElementById("example").innerHTML=Math.sqrt(81);
```

```
}
</script>
```

Que 108.) What will be the output of the following code snippet? var a = 1;

```
var b = 0;
while (a <= 3)
{
  a++;
  b += a * 2;
  print(b);
}
```

Que 109.) What will be the output of the following code snippet? var a = Math.max();

```
var b = Math.min();
print(a);
print(b);
```

Que 110.) What will be the output of the following code snippet?

```
var a = Math.max() < Math.min();
var b = Math.max() > Math.min();
print(a);
print(b);
```

Que 111.) What will be the output of the following code snippet?

```
 var a = true + true + true * 3;
print(a);
```

Que 112.) What will be the output of the following code snippet?

```
 print(typeof(NaN));
```

Que 113.) What will be the output of the following code snippet? let sum = 0;

```
const a = [1, 2, 3];
a.forEach(getSum);
print(sum);
function getSum(ele) {
  sum += ele;
}
```

Que 114.) What will be the output of the following code snippet?

```
 a = [1, 2, 3, 4, 5];
print(a.slice(2, 4));
```

Que 115.) What will be the output of the following code snippet?

```
 print(parseInt("123Hello"));
print(parseInt("Hello123"));
```

Que 116.) What will be the output of the following code snippet?

```
 var a = "hello";
var sum = 0;
for(var i = 0; i < a.length; i++) {
  sum += (a[i] - 'a');
}
print(sum);
```

Que 117.) What will be the output of the following code snippet?

```
 const set = new Set();
set.add(5);
set.add('Hello');
```

```
set.add({ name: 'Scaler' });
for (let item of set) {
 console.log(item + 6);
}
```

Que 118.) What will be the output of the following code snippet?

```
 const example = ({ a, b, c }) => {
 console.log(a, b, c);
};
example(0, 1, 2);
```

Que 119.) What will be the output of the following code snippet?

```
 let a = [1, 2, 3, 4, 5, 6];
var left = 0, right = 5;
var found = false;
var target = 5;
while(left <= right) {
  var mid = Math.floor((left + right) / 2);
  if(a[mid] == target) {
     found = true;
     break;
  }
  else if(a[mid] < target) {
     left = mid + 1;
  }
  else {
     right = mid - 1;
  }
}
```

```
if(found) {
  print("YES");
}
else {
  print("NO");
}
```

Que 120.) What will be the output of the following code snippet?

```
 let s = "00000001111111";
let l = 0, r = s.length - 1, ans = -1;
while(l <= r) {
  var mid = Math.floor((l + r) / 2);
  if(s[mid] == '1') {
     ans = mid;
     r = mid - 1;
  }
  else {
     l = mid + 1;
  }
}
print(ans);
```

Que 121.) What will be the output of the following code snippet?

```
 let n = 24;
let l = 0, r = 100, ans = n;
while(l <= r) {
  let mid = Math.floor((l + r) / 2);
  if(mid * mid <= n) {
     ans = mid;
```

```
    l = mid + 1;
  }
  else {
    r = mid - 1;
  }
}
print(ans);
```

Que 122.) What will be the output of the following code snippet?

```
 const obj1 = {Name: "Hello", Age: 16};
const obj2 = {Name: "Hello", Age: 16};
print(obj1 === obj2);
```

OOPs

1. Write a Java program to create a class called "Person" with a name and age attribute. Create two instances of the "Person" class, set their attributes using the constructor, and print their name and age.

2. Write a Java program to create a class called "Dog" with a name and breed attribute. Create two instances of the "Dog" class, set their attributes using the constructor and modify the attributes using the setter methods and print the updated values.

3. Write a Java program to create a class called "Rectangle" with width and height attributes. Calculate the area and perimeter of the rectangle.

4. Write a Java program to create a class called "Circle" with a radius attribute. You can access and modify this attribute. Calculate the area and circumference of the circle.

5. Write a Java program to create a class called "Book" with attributes for title, author, and ISBN, and methods to add and remove books from a collection.

6. Write a Java program to create a class called "Employee" with a name, job title, and salary attributes, and methods to calculate and update salary.

7. Write a Java program to create a class called "Bank" with a collection of accounts and methods to add and remove accounts, and to deposit and withdraw money. Also define a class called "Account" to maintain account details of a particular customer.

8. Write a Java program to create class called "TrafficLight" with attributes for color and duration, and methods to change the color and check for red or green.

9. Write a Java program to create a class called "Employee" with a name, salary, and hire date attributes, and a method to calculate years of service.

10. Write a Java program to create a class called "Student" with a name, grade, and courses attributes, and methods to add and remove courses.

11. Write a Java program to create a class called "Library" with a collection of books and methods to add and remove books.

12. Write a Java program to create a class called "Airplane" with a flight number, destination, and departure time attributes, and methods to check flight status and delay.

13. Write a Java program to create a class called "Inventory" with a collection of products and methods to add and remove products, and to check for low inventory.

14. Write a Java program to create a class called "School" with attributes for students, teachers, and classes, and methods to add and remove students and teachers, and to create classes.

15. Write a Java program to create a class called "MusicLibrary" with a collection of songs and methods to add and remove songs, and to play a random song.

16. Write a Java program to create a class called "Shape" with abstract methods for calculating area and perimeter, and subclasses for "Rectangle", "Circle", and "Triangle".

17. Write a Java program to create a class called "Movie" with attributes for title, director, actors, and reviews, and methods for adding and retrieving reviews.

18. Write a Java program to create a class called "Restaurant" with attributes for menu items, prices, and ratings, and methods to add and remove items, and to calculate average rating.

19. Write a Java program to create a class with methods to search for flights and hotels, and to book and cancel reservations.

20. WAP using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor gets invoked after creation of obj1 and obj2

21. Write a program by creating an 'Employee' class having the following methods and print

the final salary.

1 - 'getInfo()' which takes the salary, number of hours of work per day of employee as

parameter

2 - 'AddSal()' which adds 10000 to salary of the employee if it is less than 50000.

3 - 'AddWork()' which adds 5000 to salary of employee if the number of hours of work per day

is more than 6 hours.

22. Create a class Toy having variable toyname ,price,and static variable "madein" .

create no arg and parameterized constructor to initialize class variables .

Define static function static display and non static method display

--display() print class non static variable

--static display print static variable

create another class Electronic Toy which has fields like model no and Toy reference .

create parameterized constructor to initialize both variables.

In Electronic Class define print All function that print all values of same class as well as call

toy class static and non static functions .

23. Create class Product (pid, price, quantity) with parameterized constructor. Create a main

function in different class (say XYZ) and perform following task:

a. Accept five product information from user and store in an arrayb. Find Pid of product with highest price.

c. Create method (with array of product's object as argument) in XYZ class to calculate and return

total amount spent on all products. (amount spent on single product=price of product * quantity of

product) .

24. Create a class Employee with(empNo ,salary and totalSalary)) with following features.

a. Only parameterized constructor;

b. totalSalary always represent total of all the salaries of all employees created.

c. empNo should be auto incremented.

d. display total employees and totalSalary using class method.

25. Create a class Person with properties (name and age) with following features.

a. Default age of person should be 18;

b. A person object can be initialized with name and age;

c. Method to display name and age of person

26. Create a class to calculate Area of circle with one data member to store the radius and another

to store area value.

Create method members

1. init - to input radius from user

2. calc - to calculate area

3. display- to display area 31.

27. Create a class Math Operation with two data member X and Y to store the operand and third

data member R to store result of operation. Create method members ● init - to input X and Y from

user ● add - to add X and Y and store in R ● multiply - to multiply X and Y and store in R ● power - to

calculate X Y and store in R ● display- to display Result R .

28. Create an Object having two properties and four behaviours

(add,substract,multiply,divide) to perform desired operation.object must have setter and getter and keep all properties private.

29. Create a class to calculate Area of circle with one data member to store the radius and another to store area value. Create method members

init - to input radius from user

calc - to calculate area

display- to display area

30. Create a class MathOperation with two data member X and Y to store the operand and third data member R to store result of operation.Create method members

init - to input X and Y from user

add - to add X and Y and store in R

multiply - to multiply X and Y and store in R

power - to calculate X Y and store in R

display- to display Result R

31. Create a class MathOperation containing method ‘multiply’ to calculate multiplication of following arguments.

two integers

three float

all elements of array

one double and one integer

32. Create a class Person with properties (name and age) with following features.

Default age of person should be 18.

A person object can be initialized with name and age.

Method to display name and age of person.

33. Create a class Employee with(empNo ,salary and totalSalary) with following features.

Only parameterized constructor;

totalSalary always represent total of all the salaries of all employees created.

empNo should be auto incremented.

display total employees and totalSalary using class method.

34. Create class Product (pid, price, quantity) with parameterized constructor.Create a main function in different class (say XYZ) and perform following task:

Accept five product information from user and store in an array

Find Pid of product with highest price.

Create method (with array of product's object as argument) in XYZ class to calculate and return total amount spent on all products. (amount spent on single product=price of product * quantity of product).

35. Make list of Students having name, roll no., age, score. Write a program to accept 10 students record and arrange the Students based on the score group [0-50], [50-65],[65-80],[80-100].

36.Create class Tile to store the edge length of a square tile , and create another class Floor to store length and width of a rectangular floor. Add method totalTiles(Tile t) in Floor class with Tile as argument to calculate the whole number of tiles needed to cover the floor completely.

37. Create class OneBHK with instance variable roomArea , hallArea and price.

Create default and parameterized constructor.

Method show(): to print OneBHK data member information.

38. Create another class TwoBHK which has all the properties and behaviour of OneBHK and a new instance variable room2Area.

Create default and parameterized constructor.

Method show(): to print all data member information.

Write main function in another class(Say XYZ) and store three TwoBHK flat's.information and print information using show method. Also print total amount of all flats.

39. Create three classes Faculty (facultyid, salary), FullTimeFaculty (basic, allowance) inherits class Faculty, PartTimeFaculty (hour, rate) inherits class Faculty.

Create a method for accepting input in FullTimeFaculty and PartTimeFaculty, but salary should not be accepted. salary is calculated on the basis of (basic+allowance) for FullTimeFaculty and (hour*rate) for PartTimeFaculty. Also create method in above classes to display faculty data.

Create another class(say XYZ) for main method and store 2 fulltime and 2 parttime faculty information. Also print their details.

40. Create a class Student with two members : rollno and percentage.

Create default and parameterized constructors. Create method show() to display information.

Create another class CollegeStudent inherits Student class. Add a new membersemester to it. Create default and parameterized constructors. Also override show() method.

Create another class SchoolStudent inherits Student class. Add a new memberclassname(eg 12 th ,10 th etc.) to it. Create default and parameterized constructors. Also override show() method.

Create a class(say XYZ) with main method that carries out the operation of the project :

has array to store objects of any class(Student or CollegeStudent, SchoolStudent).

create two CollegeStudent and three SchoolStudent record objects and store them inside the array.

display all record from the array.

search record on the basic of rollno and check given rollno is of SchoolStudent or of CollegeStudent.

count how many students are having A grade,if for A grade percentage >75.

41. Create an Abstract class Processor with int member variable data and method showData to display data value.

Create abstract method process() to define processing of member data.

Create a class Factorial using abstract class Processor to calculate and print factorial of a number by overriding the process method.

Create a class Circle using abstract class Processor to calculate and print area of a circle by overriding the process method Ask user to enter choice (factorial or circle area). Also ask data to work upon; Use Processor class reference to achieve this mechanism.

42. Create Interface Taxable with members salesTax=7% and incomeTax=10.5%. Create abstract method calcTax().

Create class Employee(empId,name,salary) and implement Taxable to calculate incomeTax on yearly salary.

Create class Product(pid,price,quantity) and implement Taxable to calculate salesTax on unit price of product.

Create class for main method(Say XYZ), accept employee information and a product information from user and print income tax and sales tax respectively.

43. Explain the importance of toString() and equals() method of the Object class and override them on class Employee (empId,name,salary).

Create class for main method(say XYZ),and accept five employees information and store in an array. Also ensure if entered empId already exist or not (use equals method).

Display all employee info using toString() method.

44. Create a program that helps banks to maintain records. It should havefollowing facilities.

Anybody can create current or saving account with following initial information:account number, name, balance, and branch.

display account detail for a particular accounts.

display total money deposited in bank.

allow deposit and withdrawal in an account .

for saving account opening balance and minimum balance must be 5000.

for current account opening balance and minimum balance must be 1000.

can not withdraw the amount from the account that makes balance less than the minimum balance.

Que 1.) Design a BankAccount class with encapsulated attributes for account number, balance, and owner name. Provide methods for deposit and withdrawal.

Que 2.) Create a UserAuthentication class with encapsulated username and password attributes. Implement a method for changing the password securely.

Que 3.) Design a LibraryBook class with encapsulated attributes for title, author, and availability status. Implement methods for checking out and returning books.

Que 4.) Create an Employee class with encapsulated attributes for employee details. Provide methods for updating employee information securely.

Que 5.) Design a StudentRecord class with encapsulated attributes for student details and grades. Implement methods for adding and updating grades.

Que 6.) Create a SocialMediaProfile class with encapsulated attributes for user information. Implement methods for posting updates and changing profile settings.

Que 7.) Design an Email class with encapsulated attributes for sender, receiver, subject, and content. Provide methods for sending and receiving emails securely.

Que 8.) Create a RestaurantMenu class with encapsulated attributes for menu items and prices. Implement methods for adding and updating menu items.

Que 9.) Design a MovieTicket class with encapsulated attributes for movie details and booking status. Provide methods for reserving and canceling tickets.

Que 10.) Create an InventoryItem class with encapsulated attributes for item details and quantity. Implement methods for adding and removing items from the inventory.

Que 11.) Design a TravelReservation class with encapsulated attributes for passenger details and reservation status. Implement methods for booking and canceling reservations.

Que 12.) Create a CarRental class with encapsulated attributes for car details and availability. Implement methods for renting and returning cars.

Que 13.) Design a MusicPlaylist class with encapsulated attributes for songs and play status. Provide methods for adding, removing, and shuffling songs.

Que 14.) Design a GameCharacter class with encapsulated attributes for health and abilities. Provide methods for performing actions in the game.

Que 15.) Create an EventScheduler class with encapsulated attributes for events and schedules. Implement methods for adding, updating, and canceling events.

==========

Que 16.) Design an interface Shape with methods like calculateArea() and draw(). Implement classes for different shapes such as circle, rectangle, and triangle.

Que 17.) Create an interface DatabaseConnection with methods like connect() and disconnect(). Implement classes for different databases (MySQL, Oracle) that implement this interface.

Que 18.) Design an interface Payable with a method calculateSalary(). Implement classes for different types of employees (salaried, hourly) that implement this interface.

Que 19.) Create an interface MessagingService with methods like sendMessage() and receiveMessage(). Implement classes for different messaging platforms (SMS, Email) that implement this interface.

Que 20.) Design an interface Rentable with methods like rent() and returnVehicle(). Implement classes for different types of vehicles (car, bike) that implement this interface.

Que 21.) Create an interface Sortable with a method sort(). Implement classes for different sorting algorithms (Bubble Sort, Quick Sort) that implement this interface.

Que 22.) Design an interface PaymentMethod with methods like processPayment() and refund(). Implement classes for different payment methods (credit card, PayPal) that implement this interface.

Que 23.) Create an interface Compressible with methods like compress() and decompress(). Implement classes for different compression algorithms (ZIP, GZIP) that implement this interface.

Que 24.) Create an abstract class Animal with abstract methods like eat() and makeSound(). Implement concrete classes for different types of animals (Lion, Dog) that extend this abstract class.

Que 25.) Design an interface BankAccount with methods like deposit(), withdraw(), and getBalance(). Implement classes for different account types (Savings, Checking) that implement this interface.

Que 26.) Design an abstract class Vehicle with abstract methods like startEngine() and stopEngine(). Implement concrete classes for different types of vehicles (Car, Motorcycle) that extend this abstract class.

Que 27.) Write a lambda expression to filter a list of integers and return only the even numbers.

Que 28.) Implement a lambda expression to sort a list of strings in ascending order based on their length.

Que 29.) Create a lambda expression for performing basic calculator operations (addition, subtraction, multiplication, division) on two numbers.

Que 30.) Define a functional interface called MathOperation with a method that takes two integers and returns an integer result. Implement this interface using a lambda expression to calculate the product of two numbers.

==========

Que 31.) Create a polymorphic method to calculate the area of different shapes (circle, rectangle, triangle).

Que 32.) Implement a polymorphic method for various animals to make different sounds.

Que 33.) Design a polymorphic method to calculate the bonus for employees based on their roles.

Que 34.) Create a polymorphic method to display information about different transportation modes (car, bicycle, airplane).

Que 35.) Create a polymorphic method to process transactions for different types of bank accounts.

Que 36.) Design a polymorphic method to handle notifications in a messaging system (email, SMS, push notification).

Que 37.) Implement a polymorphic method to calculate the total salary for employees in a company with different roles.

Que 38.) Design a polymorphic method to draw various shapes (circle, square, triangle) on a canvas.

Que 39.) Implement a polymorphic method to perform mathematical operations (addition, subtraction, multiplication) on different data types (int, double).

Que 40.) Design a hierarchy of shapes (circle, rectangle, triangle) with a common base class. Implement polymorphic methods for calculating the area of each shape.

Que 41.) Create a class hierarchy for vehicles (car, bike) with a common base class. Implement inheritance for common attributes like make, model, and methods like start().

Que 42.) Design a base class Employee and subclasses for different types (Manager, Developer). Implement inheritance for common attributes like name, id, and methods like calculateSalary().

Que 43.) Create a class hierarchy for animals (mammals, reptiles) with a common base class. Implement inheritance for common methods like eat() and attributes like species.

Que 44.) Design a base class Student and subclasses for different levels (Undergraduate, Graduate). Implement inheritance for common attributes like name, id, and methods like study().

Que 45.) Create a base class MediaPlayer and subclasses for different media types (Audio, Video). Implement inheritance for common methods like play() and attributes like duration.

Que 46.) Create a base class LibraryItem and subclasses for different types (Book, DVD). Implement inheritance for common attributes like title, author, and methods like checkOut().

Que 47.) Create a class hierarchy for geographical locations (city, country) with a common base class. Implement inheritance for common attributes like name and methods like displayInfo().

Que 48.) Design a base class MenuItem and subclasses for different categories (Appetizer, MainCourse). Implement inheritance for common attributes like name, price, and methods like displayDetails().

Que 49.) Create a base class Flight and subclasses for different classes (Economy, Business). Implement inheritance for common methods like bookSeat() and attributes like availableSeats.

Que 50.) Design a base class Product and subclasses for different categories (Electronics, Clothing). Implement inheritance for common attributes like name, price, and methods like addToCart().

Que 51.) Design a base class Course and subclasses for different types (Elective, Core). Implement inheritance for common methods like register() and attributes like instructor.

Que 52.) Extend the shape hierarchy by introducing a 3D shape category (sphere, cube). Implement inheritance and polymorphism for calculating the volume and surface area of each 3D shape.

Que 53.) Introduce multiple levels of hierarchy for employees (CEO, Manager, Developer, Intern). Implement inheritance and polymorphism to handle hierarchical structures and calculate salaries considering different roles and levels.

Que 54.) Create a base class Employee with attributes like name and salary. Implement subclasses for different roles (Manager, Developer) with role-specific methods.

Que 55.) Design a class hierarchy for library items (Book, DVD) with a common base class. Include attributes like title and author. Implement methods for checking in and checking out items.

Que 56.) Design a hierarchy for geometric shapes (circle, rectangle, triangle) with a common base class. Include methods for calculating area and perimeter.

Que 57.) Create a class hierarchy for menu items (Appetizer, MainCourse) with a common base class. Include attributes like name and price. Implement methods for displaying details.

Que 58.) Design a class hierarchy for messages (Email, SMS) with a common base class. Include attributes like sender and receiver. Implement methods for sending and receiving messages.

Que 59.) Create a class hierarchy for bank accounts (Savings, Checking) with a common base class. Include attributes like accountNumber and methods for deposit and withdrawal.

Que 60.)Design a class hierarchy for shapes (line, circle, square) with a common base class. Include methods for drawing each shape.

Que 61.) Design a class hierarchy for products (Electronics, Clothing) with a common base class. Include attributes like name and price. Implement methods for adding to cart.

Que 62.) Create a Rectangle class with attributes length and width. Implement a method to calculate the area.

Que 63.)Create a Person class with attributes name and age. Implement a parameterized constructor and display method.

Que 64.) Create a BankAccount class with attributes accountNumber and balance. Implement a parameterized constructor and methods for deposit and withdrawal.

Que 65.) Create a Student class with attributes name, rollNumber, and grade. Implement a parameterized constructor and a method to display student information.

Que 66.) Create a Circle class with attribute radius. Implement a parameterized constructor and methods for calculating the circumference and area.

Que 67.) Create a Car class with attributes make, model, and year. Implement a parameterized constructor and a method to display car details.

Que 68.) Create a PersonWeight class with attributes name and weight. Implement a parameterized constructor and a method to display the person's weight category.

Que 69.) Create an EmployeeManagementSystem class that uses an array to store employee objects. Implement methods to add employees, display all employees, and find an employee by ID.

Que 70.) Create a BankCustomer class with attributes customerId and balance. Implement a parameterized constructor and methods for deposit and withdrawal, ensuring balance constraints.

Que 71.)Create a Product class with attributes productId, productName, and price. Implement a parameterized constructor and a method to display product details.

Que 72.) Create a TemperatureConverter class with attributes celsius and fahrenheit. Implement a parameterized constructor and methods to convert between Celsius and Fahrenheit.

Que 73.) Create a MobilePhone class with attributes brand, model, and price. Implement a parameterized constructor and a method to display mobile phone details.

Que 74.) Create a Student class with attributes name, id, and an array for storing grades. Implement a program that uses an array of Student objects to represent a grade book. Include methods to add grades, calculate average grades, and display student information.

Que 75.) Create a BankAccount class with attributes accountNumber and balance. Implement a program that uses an array of BankAccount objects to represent a bank's account management system. Include methods to deposit, withdraw, and display account details.

Que 76.) Create an Employee class with attributes id, name, and salary. Implement a program that uses an array of Employee objects to represent an employee database. Include methods to add employees, display all employees, and find an employee by ID.

Que 77.) Create a Car class with attributes make, model, and rentalPrice. Implement a program that uses an array of Car objects to represent a car rental system. Include methods to rent a car, return a car, and display available cars.

Que 78.) Create a Product class with attributes productId, productName, and quantityInStock. Implement a program that uses an array of Product objects to represent a product inventory management system. Include methods to add products, update quantities, and display product details.

Que 79.) Create a SchoolClass class with attributes className and an array for storing student objects. Implement a program that uses an array of SchoolClass objects to represent a school class system. Include methods to add students, display class details, and calculate average class grades.

Que 80.) Create a Room class with attributes roomNumber, type, and availability. Implement a program that uses an array of Room objects to represent a hotel reservation system. Include methods to reserve a room, cancel a reservation, and display available rooms.

Que 81.) Create a Movie class with attributes title, genre, and rating. Implement a program that uses an array of Movie objects to represent a movie database. Include methods to add movies, display movie details, and search for movies by genre.

Que 82.) Create an array of Employee objects. Input details for each employee (id, name, salary) and display the employee with the highest salary.

Que 83.)Write a method to swap two Person objects.

Que 84.) Create a method that takes a Circle object as a parameter and calculates its area.

Que 85.) Implement a method that takes a ShoppingCart object and a Product object as parameters to add the product to the cart.

Que 86.) Create a method to transfer funds between two BankAccount objects.

Que 87.) Design a base class Employee with a method calculateSalary(). Create subclasses Manager and Developer with overridden methods.

Que 88.) Write a method that takes a Product object and updates its quantity based on new stock.

Que 89.) Implement a method that takes an array of Movie objects and displays information about each movie.

Que 90.) Design a Car class with attributes make, model, and year. Implement a method to display car details.

Que 91.) Create classes for an employee payroll system. Design classes like Employee, Manager, and Intern. Implement methods for calculating salaries, handling promotions, and displaying employee details.