Assignment 1

# Q1. Write a program that takes three integer command-line arguments and prints equal if all three are equal, and not equal otherwise.

**public** **class** Ass1Q1 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println("Enter 3 nos.");

Scanner obj= **new** Scanner(System.***in***);

**int** a,b,c;

a=obj.nextInt(); b=obj.nextInt(); c=obj.nextInt();

**if**(a==b && b==c && a==c)

System.***out***.println("Equal");

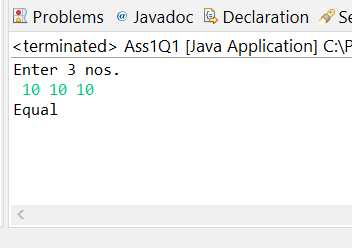
**else**

System.***out***.println("Unequal");

}

}

## OUTPUT:



# Q2. Write a Java program to sort a given binary array in linear times. b\_nums[] = { 0, 1, 1, 0, 1, 1, 0, 1, 0, 0 } Output: After sorting: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1].

**import** java.util.Scanner;

**public** **class** Ass1Q2 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//Declaring variables and entering size of array.

**int** n; **int** []nums;

System.***out***.println("Enter the size of array");

Scanner obj= **new** Scanner(System.***in***);

n=obj.nextInt();

//Inputting the array

nums=**new** **int**[n];

System.***out***.println("Enter the binary elements of array");

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

{

nums[i]=obj.nextInt();

}

//Sorting the array

**int** []res=*binSort*(nums);

//Print the sorted array

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

{

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

}

}

**public** **static** **int**[] binSort(**int** []nums)

{

**int** ind=0;

**for**(**int** i=0;i<nums.length;i++)

{

**if**(nums[i]==0)

{

**int** temp=nums[i];

nums[i]=nums[ind];

nums[ind]=temp;

ind++;

}

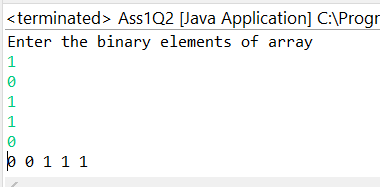
}

**return** nums;

}

}

## OUTPUT:



# Q3. Write a program that takes an integer command-line argument n and prints all the positive powers of 2 less than or equal to n. Make sure that your program works properly for all values of n.

**import** java.util.Scanner;

**public** **class** Ass1Q3 {

**public** **static** **void** main(String []args)

{

Scanner obj=**new** Scanner(System.***in***);

System.***out***.println("Enter an integer");

**int** n=obj.nextInt();

n=Math.*abs*(n);

//Print n powers of 2

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

{

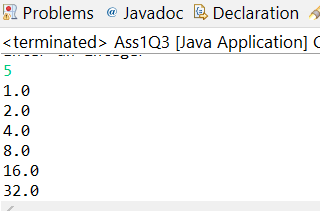
System.***out***.println(Math.*pow*(2, i));

}

}

}

## OUTPUT:



# Q4. What is the difference between ++i and i++.

|  |  |
| --- | --- |
| **I++**   1. It is a post-fix operation. Meaning it will be evaluated after the value of I is used in expression. 2. Example: i=10; i++ + 5   Output: 15 | **++i**   1. It is a pre-fix operation. Meaning it will be evaluated before the value of I is used in expression. 2. Example: i=10; ++i + 5   Output: 16 |

# Q5. In a for loop, and can be statements more complicated than declaring, initializing, and updating a loop-control variable. How can I take advantage of this ability?

The test condition might have compound relation. This helps in simultaneously updating multiple variables and in situations when we have a relationship between variables which have to also be changed in some way.

Example:

for(int i=0,j=0; i<10 && j<10;i++, j+=i)

{

Statements;}

# Q6. Can I use a double variable as a loop-control variable in a for loop?

Yes.

Example:

for(int i=0,j=0; i<10 && j<10;i++, j+=i)

{

Statements;}

# Q7. Are there cases where I must use a for loop but not a while, or vice versa?

For loops must be used in scenarios where the exact number of repetitions is known. It is a convenient and compact way of looping when the conditions are simple.

For when the exact number of repetitions is not known or the testing conditions are very complex while loops are used.

# Q8. Write a Java program to sort an array of given integers using the Bubble sorting Algorithm and Heap sort Algorithm.

**public** **class** Ass1Q8 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** arr[] = {12, 11, 13, 5, 6, 7};

**int** n = arr.length;

Ass1Q8 ob = **new** Ass1Q8();

ob.sort(arr);

System.***out***.println("Heap sort's output is: ");

ob.*printArray*(arr);

System.***out***.println("Bubble sort:");

ob.*printArray*(arr);

}

**public** **void** sort(**int** arr[])

{

**int** n = arr.length;

// Build heap (rearrange array)

**for** (**int** i = n / 2 - 1; i >= 0; i--)

heapify(arr, n, i);

// One by one extract an element from heap

**for** (**int** i = n - 1; i > 0; i--) {

// Move current root to end

**int** temp = arr[0];

arr[0] = arr[i];

arr[i] = temp;

// call max heapify on the reduced heap

heapify(arr, i, 0);

}

}

// To heapify a subtree rooted with node i which is

// an index in arr[]. n is size of heap

**void** heapify(**int** arr[], **int** n, **int** i)

{

**int** largest = i; // Initialize largest as root

**int** l = 2 \* i + 1; // left = 2\*i + 1

**int** r = 2 \* i + 2; // right = 2\*i + 2

// If left child is larger than root

**if** (l < n && arr[l] > arr[largest])

largest = l;

// If right child is larger than largest so far

**if** (r < n && arr[r] > arr[largest])

largest = r;

// If largest is not root

**if** (largest != i) {

**int** swap = arr[i];

arr[i] = arr[largest];

arr[largest] = swap;

// Recursively heapify the affected sub-tree

heapify(arr, n, largest);

}

}

/\* A utility function to print array of size n \*/

**static** **void** printArray(**int** arr[])

{

**int** n = arr.length;

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

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

System.***out***.println();

}

**void** bubbleSort(**int** arr[])

{

**int** n = arr.length;

**for** (**int** i = 0; i < n-1; i++)

**for** (**int** j = 0; j < n-i-1; j++)

**if** (arr[j] > arr[j+1])

{

// swap temp and arr[i]

**int** temp = arr[j];

arr[j] = arr[j+1];

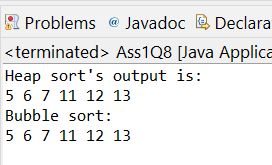
arr[j+1] = temp;

}

}

}

## OUTPUT:



# Q9. Write a program to find Leap year.

**import** java.util.Scanner;

**public** **class** Ass1Q9 {

**public** **static** **void** main(String []args)

{

**int** y; **boolean** flag=**false**;

Scanner obj=**new** Scanner(System.***in***);

System.***out***.println("Enter an year");

y=obj.nextInt();

**if**(y%4==0)

{

**if**(y%100==0)

{

**if**(y%400==0)

flag=**true**;

}

**else**

flag=**true**;

}

**if**(flag)

System.***out***.println("Leap year");

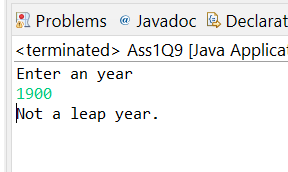
**else**

System.***out***.println("Not a leap year.");

}

}

## OUTPUT:



# Q10. Write a program GreatestCommonDivisor that finds the greatest common divisor (gcd) of two integers using Euclid’s algorithm, which is an iterative computation based on the following observation: if x is greater than y, then if y divides x, the gcd of x and y is y; otherwise, the gcd of x and y is the same as the gcd of x % y and y.

**import** java.util.Scanner;

**public** **class** Ass1Q10 {

**public** **static** **void** main(String []args)

{

Scanner obj=**new** Scanner(System.***in***);

**int** x,y,hcf;

x=obj.nextInt(); y=obj.nextInt();

hcf=*gcd*(x,y);

System.***out***.println("Gcd of "+x+" and "+y+" is: "+hcf);

}

**public** **static** **int** gcd(**int** x,**int** y)

{

**if** (y == 0)

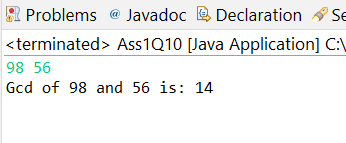
**return** x;

**return** *gcd*(y, x % y);

}

}

## OUTPUT:



# Q11.Write a program converting to binary Input - 19 output -10011 Input - 100000000 Output – 101111101011110000100000000.

**import** java.util.Scanner;

**public** **class** Ass1Q11 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner obj=**new** Scanner(System.***in***);

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

**int** n=obj.nextInt();

*decBin*(n);

}

**public** **static** **void** decBin(**int** n)

{

**int** []bin=**new** **int**[1000];**int** j=0;

**while**(n>0)

{

bin[j]=n%2;

n=n/2;

j++;

}

**for**(**int** i=j-1;i>=0;i--)

{

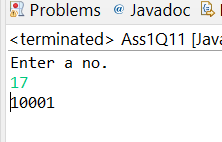
System.***out***.print(bin[i]);

}

}

}

## OUTPUT:



# Q12. Write a program Checkerboard that takes an integer command-line argument n and uses a loop nested within a loop to print out a two-dimensional n-by-n checkerboard pattern with alternating spaces and asterisks.

**import** java.util.Scanner;

**public** **class** Ass1Q12 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner obj=**new** Scanner(System.***in***);

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

**int** n=obj.nextInt();

*checkers*(n);

}

**public** **static** **void** checkers(**int** n)

{

**for**(**int** i=1;i<=n;i++)

{

**for**(**int** j=1;j<=n;j++)

{

**if**(j%2==1)

System.***out***.print("\* ");

}

System.***out***.println();

}

}

}

## OUTPUT:

