Assignment - 1

1) Write a java program to count the number of bits that are se
1 in an integer. Also prove that time complexity is $O(n)$ where n is the number of bits.

```
~~pad~~ public class A1Q1 {
        public static void main(String args[]){
              int x = 9;
              short nb = 0;
              while (x!=0){
                  nb+=(x&1);
                  x>>>=1;
              }
              System.out.println(nb);
          }
      }
```

Output:-

```
2
```

2) Write a program to find the parity bit of a number in O(n) time, where n is the word size.

```
public class A1Q2 {
    public static void main (String args[]){
        int x = 11;
        short r = 0;
        while (x!=0){
            r ^= (x & 1);
            x >>>1;
        }
        System.out.println(r);
    }
}
```

Output :-

```
1
```

3) Write a program to find the parity bit of a number in O(k) time, where k is the number of set bits.

```
public class A1Q3{
    public static void main (String args[]){
        int x = 11;
        short r = 0;
        while (x!=0){
            r^= 1;
            x &= (x-1);
        }
        System.out.println(r);
    }
}
```

Output :-

```
1
```

4) Write a program to find the parity bit of a number in O(K) time, where K is the number of retas ~~[illegible]~~ bits.

```
public class A1Q4{
    public static void main(String args[]){
        int x = 11;
        short n = 0;
        while (x != 0){
        n ^= 1;
        x &= (x-1);
       }
      System.out.println(n);
    }
  }
```

Output:-

```
1
```

5) Define a function to create a lookup table of size 2^{16} words value is the parity bit of the index

```
public class Q5 {
    static int pcParity[];
    public static int parity (int n){
            int result = 0;
            while (n != 0){
            result = result ^ 1;
            n = n & (n-1);
            }
        return (result);
    }
  public static void ~~main~~ main (String args[]){
        ~~pcParity~~
            lookUpTable ();
        }

    public static void lookUpTable ( ){
     pcParity = new int [(int)(Math.pow (2.16))];
      for( int i =0; i<pcParity.length ; i++)
       pcParity [i] = parity(i);
    }
  }
```

Output :-

0

6) Write a program to calculate to calculate the parity bit of 64 bit word using lookup table in O(n/l) time, where is the word size, l is the group size.

```
public class parityUsingLookUpTable{
    public static void main (String args[]){
        long x =12;
        System.out.println(parity(x));
    }
    public static short parity(long x){
        final int WORD_SIZE=2;
        final int BIT_MASK =3;
        int pcParity[] = {0,1,1,0};
       return (short)(pcParity)[(int)((x>>>3 * WORD
       SIZE))& BIT_MASK)]^
       pcParity[(int)((x>>> 2*Word_SIZE))&
       BIT_MASK)]^
       pcParity [(int)((x>>> WORD_SIZE))&
        BIT_MASK)]^
       pcParity [(int)((x & BIT_MASK))];
    }
}
```

Output :-

0

7) Write a program to calculate parity bit of a 64 bit word using XOR & right shift operator.

```
import java.util.Scanner;
public class Q7{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number");
        Long n = sc.nextLong();
        Long f = 64;
        while(f!=0){
            n^=(n>>>f);
            f!=2;
        }
    }
}
```

Output :-

```
Enter a number
  232
Parity bit is 0
```

8) Write a program to swap the ith bit with jth bit of a number.

```
~~pu~~ public class A1Q8{
        public static void main(String args[]){
                long x = 73;
                int i = 1;
                int j = 6;
                if(((x>>>i)&1) != ((x>>>j)&1)){
                        long c = (1l<<i) | (1l<<j);
                        x ^= c;
                }
                System.out.println(x);
        }
}
```

output :-

```
11
```

Page no.   Signature

9) Design a function to create a lookup table A such that for every 16 bit no. y, A[y] holds the bit-reversal of y

```
static void reversalLookUp (int lookUp [ ]) {
    for ( int i = 0; i< 65536 ; i++) {
        int n = 1;
        int r = 0;
        while ( n>0) {
            r <<=1;
            if ((n & 1)==1)
                r ^=1;
            n >>=1;
        }
        lookUp [i] = r;
    }
}
```

10) Write a program to find the bit reversal of a number using the lookup table created in Q9.

```
import java. ~~util~~ util. Scanner;
public class Q10{
    static void reversal lookUp ( int LookUp [ ]) {
        for(int i =0; i<65536 ; i++){
            int ~~m~~ n = i;
            int r = 0;
            while (n>0) {
                r <<=1;
                if ((n & 1) ==1)
                    r ^=1;
                n >>=1;
            }
```

11) Write a program to find the ~~number~~ closest integer with the same weight.

```
public class A1Q11{
    public static void main(String args[]){
        int x = 10;
        int n= x & ~(x-1);
        if ((n&1) == 1){
            n = (~x & ~(~x-1));
        int mask = n|(n>>1);
        System.out.println(x ^ mask);
    }
}
```

Output :-

```
9
```

12) Write a program to compute x + y using Bitwise.

```
public class Q12 {
    public static void main (String args[]) {
        int x = 12, y = 2;
        int sum = 0;
        while (x != 0) {
            if ((x & 1) != 0)
                sum = add (sum, y);
            }
            x >>>= 1;
            y <<= 1;
        }
        System.out.println (sum);
    }
public static int add (int x, int y) {
    int carry;
    while (y != 0) {
        carry = x & y;
        x = x ^ y;
        y = carry << 1;
    }
    return x;
  }
}
```

Output :-

```
~~2~~ 24
```

13) Write a program to compute x/y using bitwise:

```
public class q13 {
    public static void main (String args[]){
        long x=6, y=3;
        long result =0;
        int power =0;
        long y power = y<<power;
        while(x>=y){
            while(ypower>x){
                ypower >>>=1;
                --power;
            }
        System.out.println(result);
    }
}
```

Output :-

2

14) Write a program to compute x^y using bitwise.

```
public class Q14 {
    public static void main (String args[]){
        int x =2, y =5;
        int result = 1;
        int p = y;
        while (p!=0){
            if ((p&1)!=0)
                result *=x;
        }
        x *= x;
        p>>>=1;
    }
    System.out.println(result);
  }
}
```

Output:-
32

15) Write a program to check if a decimal is palindrome

```
import java.util.Scanner;
public class Q15{
    public static void main (String args[]){
     Scanner sc = new Scanner (System.in);
     System.out.println ("Enter ");
     int n = sc.nextInt ( );
     int nod = (int) Math.log10(n)+1 ;
     System.out.println (nod);
     int msd = (int) Math.pow (10, nod-1);
     System.out.println (msd);
    while (n!=0){
      System.out.println (" Not Palindrome");
      System.exit(0);
    }
    n% = msd;
    n = n/10;
    msd = msd/100;
    System.out.println (n);
   }
  System.out.println (" Number is Palindrome");
}}
```

Output :-

```
121
3
100
1
2
22
0  Entered number is palindrome
```

Signature

16) Write a program which test if 2 rectangle have a non empty intersection if the intersection is non empty, return the rectangle formed by their intersection.

```
public class Q24{
    public static void main(String args[]){
        Rectangle A = new Rectangle(1,2,3,4);
        Rectangle B = new Rectangle(3,2,4,3);
        checkIntersect(A,B);
    }
    public static void checkIntersect(Rectangle A, Rectangle B){
        if(A.get()<=(B.getX()+B.getW()) &&
        (A.getX()+A.getW()<=(B.get()) &&
        (A.getY())<=(B.getY()+B.getH()) &&
        (A.get Y()+A.getH()) >= B.getY())){
        System.out.println("Not Intersect");
        else{
        System.out.println("Intersect");
        int x,y,h,w;
        x = Math.max(A.getX(), B.getX());
        y = Math.max(A.getY(), B.getY());
        h = Math.~~max~~ min(A.y+B.h, A.y+B.h)-
            Math.~~max~~ max(A.y+B.y);
        w = Math.min(A.x+A.w, B.x+B.w)-
            Math.max(A.x, B.x);
        System.out.println("Intersecting rectangle"
        +x+"   "+y+"   "+h+"   "w+"   ");
}}}
public class Rectangle {
    int x, y, h, w;
    Rectangle (int x, int y, int h, int w) {
            this.x = x;
            this.y = y;
            this.h = h;
            this.w = w;
        }
    } ~~int~~ int getX(){
            return x;
        }
      int getY(){
          return y;
      }
    int getH(){
      return h;
    }
     int get W(){
        return w;
     }
   }
```

Output:-

```
Intersecting
  3 2 4 1
```