SIKSHA 'O' ANUSANDHAN
DEEMED TO BE UNIVERSITY

Admission Batch: **Session:**

Laboratory Record

Computer Science Workshop 2 (CSE 3141)

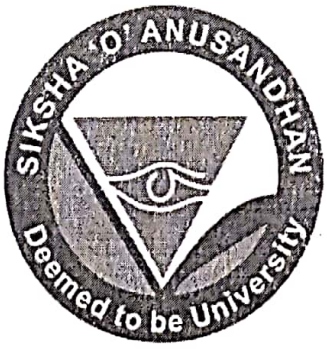
Submitted by

Name: Gaswat Mohanty

Registration No.: 1941012407

Branch: Computer Science and Engineering (CSE)

Semester: 4th Section: D



Department of Computer Science & Engineering

Faculty of Engineering & Technology (ITER)

Jagamohan Nagar, Jagamara, Bhubaneswar, Odisha - 751030

INDEX

Sl. No	Name of Program	Page No	Remarks
1	Assignment - 1 (Primitive datatypes)	1-12	

Assignment on Primitive Data Types

Q1) write a program to count the number of bits that are set 1 in an integer. Also prove that time complexity is O(n) where n is the number of bits.

```
public class A1Q1 {
    public static void main (String [] args) {
        int n = 11;
        int count = 0;
        System.out.println (Integer.toBinaryString (n));
        while (n!=0) {
            if ((n&1)==1)
                count++;
            n = n >>> 1;
        }
        System.out.println ("The total number of 1s are "+count);
    }
}
```

Q2) 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 n = 15;
        System.out.println (Integer.toBinaryString (n));
        int parity = 0;
        while (n!=0) {
            parity = parity ^ (n&1);
```

```
        n = n>>>1;
    }
    System.out.println("Parity= "+parity);
  }
}
```

Q3) 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 n = 10;
      int parity = 0;
      System.out.println(Integer.toBinaryString(n));
      while (n!=0) {
          parity = ~parity;
          n = n & (n-1);
      }
      System.out.println(parity);
  }
}
```

Q4) 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 A1Q4 {
  public static void main(String[] args) {
      int n = 10;
      int parity = 0;
      System.out.println(Integer.toBinaryString(n));
      while(n!=0) {
          parity = ~parity;
```

```
        n= n & (n-1);
      }
      System.out.println (parity);
    }
}
```

Q5) Define a function to create a lookup table of size 2^{16} whose value is the parity bits of the index.

```
public class A1Q5 {
    public static void main (String [] args) {
        int parity = 0;
        int [] lookuptable = new int [64];
        for (int i=0; i<64; i++) {
            int n = i;
            while (n!=0) {
                parity = parity ^ (n & 1);
                n = n>>>1;
            }
            System.out.println (i+ "->" + Integer.toBinaryString(i)+
                                "->" + parity);
        }
    }
}
```

Q6) write a program to calculate the parity bit of 64 bit word using look up table in $O(n/L)$ time, where n is the wordsize.

Note: consider group size is 16 bit for the problem

```
import java.util.Scanner;
public class A1Q6{
    public static void main (String [] args){
        Scanner input = new Scanner(System.in);
        int parity[] = new int [65536];
        lookup creator (parity);
        System.out.println("Enter the binary word whose parity
                            is to be calculated");
        long x = input.nextLong();
        int m = 0xFFFF;
        int ret = parity[(int)(x>>48) & m] ^ parity[(int)
                  (x>>32) & m] ^ parity[(int)(x>>16)
                  & m] ^ parity[(int)x & m];
        System.out.println("Parity of the entered number is: "
                           + ret);
    }
    static lookupcreator (int parity[]){
        for (int i=0; i<65536; i++){
            int x=i;
            x^= x>>8;
            x^= x>>4;
            x^= x>>2;
            x^= x>>1;
            parity[i] = (x & 1);
        }
    }
}
```

07) write a program to calculate parity bit of a 64 bit word using only XOR and right shift operator

```
import java.util.scanner;
public class A107{
  public static void main(String[] args){
    Scanner input = new Scanner(System.in);
    System.out.println("Enter the number whose parity bit is
                        to be calculated:");
    long x = input.nextLong();
    x^= x>>32;
    x^= x>>16;
    x^= x>>8;
    x^= x>>4;
    x^= x>>2;
    x^= x>>1;
    System.out.println("Parity is: "+(x&1));
  }
}
```

08) write a program to swap the i^{th} bit with j^{th} bit of a number.

```
import java.util.Scanner;
public class A108{
  public static void main(String[] args){
    Scanner input = new Scanner(System.in);
    System.out.println("Enter the number: ");
    int n = input.nextInt();
    System.out.println(Integer.toBinaryString(n));
```

```
        System.out.println("Enter the positions to swap: ");
        int i = input.nextInt();
        int j = input.nextInt();
        if ((1 & (n >> i)) != (1 & (n >> j))) {
            int bitMask = (1 << i) | (1 << j);
            n ^= bitMask;
        }
        System.out.println(Integer.toBinaryString(n));
    }
}
```

8a) Design a function to create a lookup table A such that for every 16 bit number y, A[y] holds the bit-reversal of y.

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

Q10) write a program to find the bit reversal of a number using the lookup table created in Q9.

```
import java.util.Scanner;
public class A1Q10{
    static void reversalLookup(int lookup[]){
        for(int i=0; i<65536; i++){
            int n=i;
            int r=0;
            while(n>0){
                r<<=1
                if((n&1)==1)
                    r^=1;
                n>>=1;
            }
            lookup[i]=r;
        }
    }
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int lookup[] = new int[65536];
        reversalLookup(lookup);
        System.out.println("Enter the number whose bit reversal
                            is to be calculated");
        int n = input.nextInt();
        if(n>=65536)
            System.out.println("the entered number is outside
                                the range of the lookup table);
        else
            System.out.println(lookup[n]);
    }
}
```

Q11) write a program to find the closest integer with same weight.

```
public class A1Q11 {
    public static void main (String [] args) {
        long p = 67;
        long y = closest (p);
        System.out.println (y);
    }
    static long closest (long p) {
        for (int i=0; i<63; i++) {
            if ((((p>>> i) & 1) != ((p>>> (i+1)) & 1))) {
                p ^= (1<<i) | (1<<(i+1));
                return p;
            }
        }
        return Long.MIN_VALUE;
    }
}
```

Q12) write a program to compute x*y using bit wise operator.

```
public class A1Q12 {
    public static void main (String [] args) {
        System.out.println (Multiply (p, y));
    }
    static long add (long p, long y) {
        while (y != 0) {
            long carry = p & y;
            p = p ^ y;
```

```
        y = carry << 1;
    }
    return p;
}
static long multiply (long p, long y) {
    long sum = 0;
    while (p != 0) {
        if ((p & 1) != 0) {
            sum = add (sum, y);
            p >>= 1;
            y <<= 1;
        }
        return sum;
    }
}
```

Q13) write a program to compute x / y using bit wise operator.

```
import java.util.scanner;
public class A1Q13 {
    public static void main (String[] args) {
        Scanner input = new scanner (System.in);
        System.out.println ("Enter the numbers: ");
        int a = input.nextInt();
        int b = input.nextInt();
        int t = 0, q = 0;
        for (int i = 0; a >= b; i++) {
            if ((b << i) > a) {
                q |= (1 << (i - 1));
```

```
            a -= (b << (i - 1));
            i = -1;
        }
    }
    System.out.println(a);
  }
}
```

Q14) write a program to compute x^y using bit wise operator.

```
import java.util.scanner;
public class A1Q14 {
   public static void main (String [] args) {
      Scanner input = new Scanner (System.in);
      System.out.println ("Enter the numbers: ");
      int x = input.nextInt();
      int y = input.nextInt();
      int res = 1;
      while (y != 0) {
         if ((y & 1) == 1)
            res *= x;
         x *= x;
         y >>= 1;
      }
      System.out.println (res);
   }
}
```

Q15) write a program to check if a decimal number is a palindrome.

```
import java.util.*;
public class A1Q15 {
  public static void main (String[] args) {
```

```
Scanner input = new Scanner(System.in);
System.out.println("Enter the number: ");
int n = input.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((n/msd)+" "+(n % 10));
    if ((n/msd) != (n% 10)) {
        System.out.println("Entered number is not
                                    palindrome.");
        System.exit(-1);
    }
    n %= msd;
    n /= 10;
    msd /= 100;
    System.out.println(n);
}
System.out.println("Entered number is palindrome.");
}
}
```

Q16) write a program that tests if two rectangle have a non-empty intersection. If the intersection is non-empty return the rectangle formed by their intersection.

```
public class A1Q16{
  public static void main(String[] args){
      Rectangle r1 = new Rectangle(1,8, 10, 15);
      Rectangle r2 = new Rectangle(6, 2, 12, 22);
      boolean intersectedRes= isIntersected(r1,r2);
```

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

  static boolean isIntersected(Rectangle R1, Rectangle R2) {
      return R1.pcor <= R2.pcor +
                R2.width && R2.pcor <= R1.pcor +
                R1.width && R1.ycor <= R2.ycor +
                R2.height && R2.ycor <= R1.ycor + R1.height;
  }
}

class Rectangle {
  int pcor, ycor=0; double width, height=0;

  Rectangle(int p, int y, double width, double height) {
      pcor=p;
      ycor=y;
      this.width = width;
      this.height = height;
  }
}
```