

SIKSHA 'O' ANUSANDHAN

DEEMED TO BE UNIVERSITY

Admission Batch: 2019-23

Session: 2021-22

Laboratory Record

Computer Science Workshop 2 (CSE 3141)

Submitted by

Name: BRAVISH GHOSH

Registration No.: 1941012333

Branch: CSE

Semester: 4 Section: Q



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	Write a java 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		
2	Write a program to find the parity bit of a number in $O(n)$ time, where n is the word size.		
3	Write a program to find the parity bit of a number in $O(k)$ time, where k is the number of set bits.		
4	Write a program to find the parity bit of a number in $O(k)$ time, where k is the number of set bits.		
5	Define a function to create a lookup table of size 256 whose value is the parity bits of the index.		
6	Write a program to calculate the parity bit of a 64 bit word using look up table in $O(n/L)$ time, where n is the word size and L is the group size. Note: Consider group size is 16 bit for the problem.		
7	Write a program to calculate parity bit of a 64 bit word using only xor and right shift operator.		
8	Write a program to swap the i th bit with j th bit of a number.		
9	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 .		
10	Write a program to find the bit reversal of a number using the lookup table created in Q9.		
11	Write a program to find the closest integer with the same weight.		
12	Write a program to compute XXY using bit wise operator.		
13	Write a program to compute X/Y using bit wise operator		
14	Write a program to compute XY using bit wise operator.		
15	Write a program to check if a decimal number is a palindrome.		
16	Write a program which test if two rectangle have a nonempty intersection. If the intersection is nonempty, return the rectangle formed by their intersection.		

1. Write a java 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.

```
import java.util.*;
public class Q1 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number: ");
        int num=sc.nextInt();
        int count=0;
        String conv="";
        while(num>0) {
            int a=num % 2;
            conv=a + conv;
            num=num / 2;
        }
        for(int i=0;i<conv.length();i++) {
            if(conv.charAt(i) == '1')
            {
                count++;
            }
        }
        System.out.println("Binary of number is "+conv);
        System.out.println("No. of 1's in the number is "+count);
    }
}
```

OUTPUT:

Enter a number:

6

Binary of number is 110

No. of 1's in the number is 2

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

```
import java.util.*;
public class Q2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a number: ");
        int k=sc.nextInt();
        int c=0;
        while(k!=0) {
            c=c^1;
            k=(k&(k-1));
        }
        System.out.println("The Parity is "+c);
        if(c==0) {
            System.out.println("Even Parity");
        }
        else
            System.out.println("Odd Parity");
    }
}
```

OUTPUT:

Enter a number:

11

The Parity is 1

Odd Parity

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

```
import java.util.*;
public class Q3 {
    public static void main(String[] args)
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number:");
        n=sc.nextInt();
        int x=n;
        int count=0;
        while(n>0)
        {
            count^=1;
            n&=(n-1);
        }
        System.out.println("Parity bit is:"+count);
    }
}
```

OUTPUT:

Enter the number:

22

Parity bit is 1

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

```
import java.util.*;
public class Q4 {
    public static void main(String[] args)
    {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number:");
        n=sc.nextInt();
        int x=n;
        int count=0;
        while(n>0)
        {
            count^=1;
            n&=(n-1);
        }
        System.out.println("Parity bit is:"+count);
    }
}
```

OUTPUT:

Enter the number:

12

Parity bit is 0

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

```
static void 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);  
    }  
}
```

6. Write a program to calculate the parity bit of a 64 bit word using look up table in $O(n/L)$ time, where n is the word size and L is the group size.

```
import java.util.*;
public class ParityBitLookUpTable
{
    static void 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);
        }
    }

    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int parity[]=new int[65536];
        lookupcreator(parity);
        System.out.println("Enter the binary word whose parity is to be calculated:");
        long x=sc.nextLong();
        int m=0xFFFF;
        int prt=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:"+prt);

    }
}
```

OUTPUT:

Enter the binary word whose parity is to be calculated:

1665

Parity of the entered number is 0

7. Write a program to calculate parity bit of a 64 bit word using only xor and right shift operator.

```
import java.util.*;
public class Q7 {
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number whose parity is to be calculated:");
        long x=sc.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));
    }
}
```

OUTPUT:

Enter the number whose parity is to be calculated:

132

Parity is 0

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

```
import java.util.*;
public class Q8 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number: ");
        int n=sc.nextInt();
        int i=2, j=3;
        if(((n>>>i)&1)!=((n>>>j)&1)) {
            int bitmark=(1<<i)|(1<<j);
            n=n^bitmark;
        }
        System.out.println("The Number after swapping is "+n);
    }
}
```

OUTPUT:

Enter the number:

9

The Number after swapping is 5

9. 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 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;
    }
}
```

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

```
import java.util.*;
public class BitReversalUsingLookUpTable
{
    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 sc=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=sc.nextInt();
        if(n>=65536)
            System.out.println("The entered number is outside the range of the lookup table");
        else
            System.out.println(lookup[n]);
    }
}
```

OUTPUT:

Enter the number whose bit reversal is to be calculated:

200

19

Name: BRAVISH GHOSH

Regd. Number:

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

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

```
import java.util.*;
public class Q11 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a Number: ");
        int a=sc.nextInt();
        int i=0;
        int b=a;
        while(b!=0) {
            if(((b>>i)&1)!=((b>>(i+1))&1)) {
                a=a^(1<<i);
                a=a^(1<<(i+1));
                break;
            }
            i++;
        }
        System.out.println("The closest integer of the number is "+a);
    }
}
```

OUTPUT:

Enter a Number:

9

The closest integer of the number is 10

Name: BRAVISH GHOSH

Regd. Number:

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

12. Write a program to compute XXY using bit wise operator.

```
import java.util.*;
public class Q12 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the numbers: ");
        int x=sc.nextInt();
        int y=sc.nextInt();
        int res=0;
        while(y>0) {
            if((y&1)!=0) {
                res+=x;
            }
            x=x<<1;
            y=y>>1;
        }
        System.out.println(res);
    }
}
```

OUTPUT:

Enter the numbers:

12

13

156

Name: BRAVISH GHOSH

Regd. Number:

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

13. Write a program to compute X/Y using bit wise operator.

```
import java.util.*;
public class Q13 {

    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the numbers:");
        int x=sc.nextInt();
        int y=sc.nextInt();
        int t=0, q=0;
        for(int i=0;x>=y;i++)
        {
            if((y<<i)>x)
            {
                q|=(1<<(i-1));
                x-=(y<<(i-1));
                i=-1;
            }
        }
        System.out.println(q);
    }
}
```

OUTPUT:

Enter the numbers:

50

5

10

Name: BRAVISH GHOSH

Regd. Number:

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

14. Write a program to compute X^Y using bit wise operator.

```
import java.util.*;
public class Q14{

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the number: ");
        double x=sc.nextInt();
        int y=sc.nextInt();
        double res=1.0;
        if(y<0) {
            x=1/x;
            y=-y;
        }
        while(y!=0) {
            if((y&1)==1)
                res=res*x;
            x=x*x;
            y=y>>>1;

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

OUTPUT:

Enter the number:

5

3

125.0

Name: BRAVISH GHOSH

Regd. Number:

Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)

15. Write a program to check if a decimal number is a palindrome.

```
import java.util.*;
public class Q15 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        int x=sc.nextInt();
        int y=x;
        int rev=0;
        while(x!=0) {
            int rem=x%10;
            rev=(rev*10)+rem;
            x=x/10;
        }
        System.out.println(rev);
        if(y==rev) {
            System.out.println("Palindrome");
        }
        else {
            System.out.println("Not Palindrome");
        }
    }
}
```

OUTPUT:

Enter a number:

113322

223311

Not Palindrome

16. Write a program which test if two rectangle have a nonempty intersection. If the intersection is nonempty, return the rectangle formed by their intersection.

```
class Rectangle{
    int x,y,width,height;
    public Rectangle(int x,int y,int width ,int height) {
        this.x=x;
        this.y=y;
        this.width=width;
        this.height=height;
    }
    void Disp() {
        System.out.println("x: "+x+" y: "+y+" width: "+width+" height: "+height);
    }
}

public class Q16 {
    public static Rectangle intersectRectangle(Rectangle R1,Rectangle R2) {
        if(!isIntersect(R1,R2)) {
            return new Rectangle(0,0,-1,-1);
        }
        return new Rectangle (
            Math.max(R1.x, R2.x),Math.max(R1.y, R2.y),
            Math.min(R1.x+R1.width,R2.x+R2.width)-Math.max(R1.x,R2.x),
            Math.min(R1.y+R1.height,R2.y+R2.height)-Math.max(R1.y,R2.y));
    }
    public static boolean isIntersect(Rectangle R1,Rectangle R2)
    {
        return R1.x<=R2.x+R2.width && R1.x+R1.width>=R2.x && R1.y<=R2.y+R2.height &&
        R1.y+R1.height>=R2.y;
    }
    public static void main(String[] args) {
        Rectangle R1= new Rectangle(2,3,4,5);
        R1.Disp();
        Rectangle R2= new Rectangle(7,5,10,12);
        R2.Disp();
        boolean result =isIntersect(R1,R2);
        if(result)
            System.out.println("Rectangles intersect each other");
        else
            System.out.println("Rectangles do not intersect each other");
        Rectangle R3=intersectRectangle(R1,R2);
        R3.Disp();
    }
}
```

OUTPUT:

x: 2 y: 3 width: 4 height: 5

x: 7 y: 5 width: 10 height: 12

Rectangles do not intersect each other

