Pureti Kundhana Surya Sri

A creative and detailed individual has the capacity to develop an effective and efficient solution with notolerance for errors and possess a positive attitude towards individual goals and an organizational goal.

9346144578 kundhupureti@gmail.com Tadepalligudem

EDUCATION DETAILS

Bachelor's in technology, Electronics and communication engineering — **2019 - 2023**
Sasi Instutite of Technology & engineering -9 CGPA

Board of Intermediate, MPC — **2018 - 2019**
Aditya Junior College -10 CGPA

Secondary School Certificate — **2016 - 2017**
Kimberley E.M School – 9.8 CGPA

TECHNICAL SKILLS

- C
- Python
- Html
- SQL

CAREER SKILLS

- Innovative Thinking
- Problem Solving
- Teamwork & Collaboration
- Research & Analysis
- Leadership

AREA OF INTEREST

- English
- Telugu
- Hindi

CERTIFICATIONS

- Certified in DCA & C Language – Srikara Computers
- Certified in MATLAB for signal processing

HOBBIES

- Decoring the home
- Gardening
- Travelling

PROJECT- 1

College Project

Mutual Coupling Mitigation Using Metallic Strip Between the Array Elements

- The aim of the project is to reduce the mutual coupling between the array elements by using HFSS software.
- When two antenna elements are close together, the mutual coupling takes place due to free space radiations.

ACHIEVEMENTS

- Hackerrank Basic Certificate in “Python” and “Problem Solving”.
- Participated in " Web designing - WebX’20 ” Workshop organized by Sasi Institute of Technology and Engineering on 15th Feb, 2020.
- Organised as a student coordinator for conducting the workshop on “Photoshopping” by LAKSYA Association, Sasi institute of technology & engineering.
- Participation Certificate in “ Poster Presentation ” in Techno Cultural Fest, and organised “E-CUBIC” ANVESHNA-2023, Sasi institute of technology and engineering.

DECLARATION

I solemnly declare that the information furnished above is free from errors to the best of my knowledge and belief.

Date: 31-5-2023
Place: Tadepalligudem

Signature
P. Kundhana

Assignment-2

Bitwise Operators

Bitwise operators are used to performing the manipulation of individual bits of a number.

They can be used with any integral type (char, short, int).

They are used when performing update and query operations of the binary indexed trees.

There are 4 types of bitwise operators:

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)
- Bitwise complement (~)

1.Bitwise AND :

This operator is a binary operator, denoted by &.

It returns bit by bit AND of input values i.e, if both of the bits are 1, it gives 1, else it shows 0.

Eg: a=5=0101

b=7=0111

0101 = 5

2.Bitwise OR :

This operator is a binary operator denoted by 1.

It returns bit by bit OR of input values i.e, if either of the bits is 1, it is gives 1, else it shows 0.

Eg: a=5=0101

b=7=0111

0111=7

3.Bitwise XOR :

This operator is a binary operator, denoted by '^'.

It returns bit by bit XOR of input values, i.e, if bits are different, it gives 1, else it shows 0.

Eg: a=5=0101

b=7=0111

0010=2

4. Bitwise Complement (~):

This operator is a unary operator denoted by ~.

Assignment-2

It returns the one's complement representation of the input value i.e, with all the bits inverted, which means it makes every 0 to 1 , and every 1 to 0.

Eg: a=5= 0101

~ 0101

1010 = 10

```
Class Demo
{
public static void main(String [] args)
{
System.out.println("Bitwise AND :" +(a&b));
System.out.println("Bitwise OR :" +(a|b));
System.out.println("Bitwise XOR :" +(a^b));
System.out.println("Bitwise complement :" +(~a));
}
}
```

Bit Shift Operators:

It also called as shift operators .

Shift operators are used to shift the bits of a number left or shift, thereby multiplying or dividing the number by two, respectively.

1.Left Shift Operator :

The left shift operator shifts all bits towards the left by a certain number of specified bits.

It is denoted by <<.

Eg. 0 0 1 0

0 0 1 0

2. Signed Right shift operators :

The signed right shift operator shifts all bits towards the right by a certain no. the right by a certain number of specified bits. It denoted by >>.

2 -> 0 0 1 0

0 0 1 0

3.Unsigned Right shift operators:

It also provides an unsigned right shift. It denoted by >>>.

Eg: 8 = 1 0 0 0 -8 =1 0 0 0