++

```
Instructions:
A) Save your lab.doc as LAB_no_RollNo.doc.  At the end of lab you need to submit your all programs along with the output.
-- LAB_No_Roll_No_2hr.doc for  lab task executed duing the lab
-- LAB_No_Roll_No_complete.doc for Full solution of the Lab assignment ( It should contain all lab assignemnt/problems)
B) Use/paste the snapshot of the steps followed along with result/s.
C) Mention your observation/comment after results in the doc.
D) Along with the doc/pdf file you need to upload your c program filles with following nomenaculture.
-- LAB_No_Prob_No.c
```

++

Objective(s):

- To familiar with Decision Statements (if, if-else, if-else if ladder, switch and GOTO)

++

Link to All the Codes

PART A : Conceptual Questions

1. Write a program in C to determine whether a person is eligible to vote . (using if statement)

Sample input	Sample output
19	Eligible to vote
6	Not Eligible

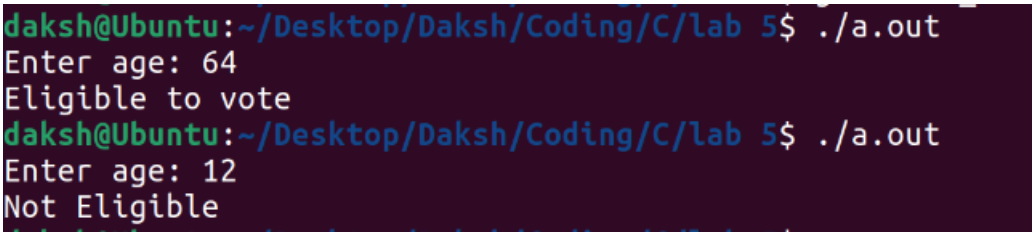
Problem Analysis:

We have to check if the age of the user is equal to or above the minimum age required to vote. This C program can be coded using an if-else statement

Code:

```c
/* Write a program in C to determine whether a person is eligible to vote . (using if statement)
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/
#include <stdio.h>

int main() {
    // Prompt the user to enter age
    printf("Enter age: ");
    int age;
    scanf("%d", &age);

    // Check if the age is greater than or equal to 18
    if (age >= 18) {
        // If age is 18 or older, the person is eligible to vote
        printf("Eligible to vote\n");
    } else {
        // If age is less than 18, the person is not eligible to vote
        printf("Not Eligible\n");
    }

    return 0;
}
```

Output:



Discussion and Conclusion:
In this C program, I determined a person's voting eligibility using an if statement. By checking the user's age against the minimum voting age, the program outputs "Eligible to vote" if eligible, and "Not Eligible" otherwise.

2. Write a program to enter a character and then determines whether the entered character is an alphabet or not. (using if- else statement)

Sample input	**Sample output**
A	Alphabet
~	Not an Alphabet

Problem Analysis:

The task is to determine if an entered character is an alphabet. Using an **if-else** statement, the program checks the input. This involves basic character validation logicby using ASCII Code values.

Code:

```c
/* Write a program to enter a character and then determines whether the entered character is an alphabet or not. (using if- else statement )
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/

#include <stdio.h>

int main() {
    char c;

    // Prompt the user to enter a character
    printf("Enter a character: ");
    scanf(" %c", &c);  // Note: There is a space before %c to consume the newline character from previous input

    // Check if the character is an uppercase or lowercase alphabet using the known ASCII Code
    if (((int)c >= 65 && (int)c <= 90) || ((int)c >= 97 && (int)c <= 122)) {
        printf("Alphabet\n");
    } else {
        printf("Not an Alphabet\n");
    }

    return 0;
}
```

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter a character: Z
Alphabet
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter a character: -
Not an Alphabet
```

Discussion & Conclusion:

I created a C program to validate characters as alphabets. The **if-else** statement efficiently handled the check, confirming if the input was an alphabet. This exercise enhanced my understanding of conditional structures and their utility in character validation.

3. Write a program in C to find whether a given year is a leap year or not. (a year is leap if it is divisible by 4 and divisible by 100 or 400.)

Sample input	**Sample output**
2020	Leap Year
2007	Not a Leap Year

Problem Analysis:

The goal is to determine if a given year is a leap year based on specific conditions. Using C, I checked if the year is divisible by 4, 100, and 400, ensuring accurate leap year identification.

Code:

```c
/*  Write a program in C to find whether a given year is a leap year or not. (a year is leap if it is divisible by 4 and divisible by 100 or 400.)
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/

#include <stdio.h>

int main() {
    int yr;
    printf("Enter Year: ");
    scanf("%d", &yr);

    // Check if the year is a leap year using and (&&) and or (||) conditions
    if ((yr % 100 != 0 && yr % 4 == 0) || (yr % 400 == 0)) {
        printf("Leap Year\n");
    } else {
        printf("Not a Leap Year\n");
    }


    return 0;
}
```

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter Year: 2020
Leap Year
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter Year: 2007
Not a Leap Year
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter Year: 1900
Not a Leap Year
```

Discussion & Conclusion:

I crafted a C program to identify leap years using logical conditions. The code utilized multiple divisibility checks to accurately classify years. This task honed my skills in conditional logic, providing a solid grasp of leap year criteria.

4. Write C program to calculate tax, given the following conditions: (use nested if-else statement)

Income Range	Tax rate	Tax to be paid
Up to Rs.2,50,000	0	No tax
Between Rs 2.5 lakhs and Rs 5 lakhs	5%	5% of your taxable income
Between Rs 5 lakhs and Rs 10 lakhs	20%	Rs 12,500+ 20% of income above Rs 5 lakhs
Above 10 lakhs	30%	Rs 1,12,500+ 30% of income above Rs 10 lakhs

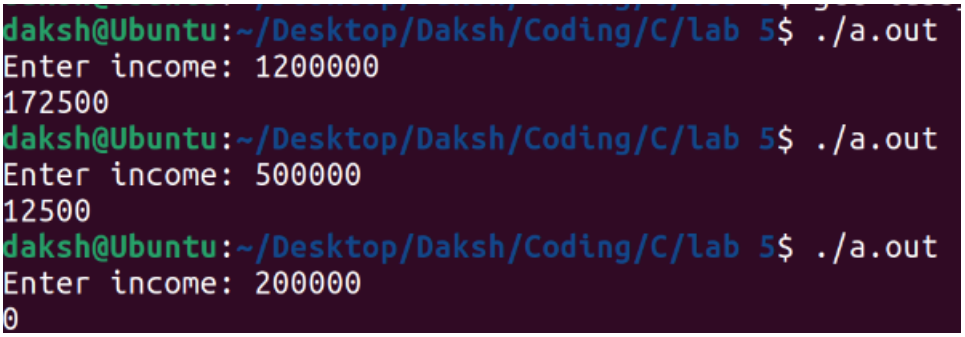
Sample input	**Sample output**
Enter the income : 1200000	172500

Problem Analysis: The task is to calculate tax based on specified income ranges. Using nested **if-else** statements, the C program checks the income, applies the corresponding tax rate, and calculates the tax amount accordingly.

Code:

```c
/* Write C program to calculate tax, given the following conditions: (use nested if-else statemen
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/

#include <stdio.h>

int main() {
    int inc, tax;

    // Prompt user for income
    printf("Enter income: ");
    scanf("%d", &inc);

    // Calculate tax based on income brackets
    if (inc > 250000) {
        if (inc < 500000)
            tax = 0.05 * inc;
        else if (inc < 1000000)
            tax = 12500 + 0.2 * (inc - 500000);
        else
            tax = 112500 + 0.3 * (inc - 1000000);
    } else {
        tax = 0;
    }

    // Print calculated tax
    printf("%d\n", tax);

    return 0;
}
```

Output:



Discussion & Conclusion:
I developed a C program to calculate taxes, incorporating nested **if-else** statements for precise taxation. The code efficiently determined tax liability based on the given income brackets, enhancing my understanding of nested conditional structures.

5. Write a program in C that accepts a number from 1 to 10. Display whether a number is even or odd using a switch case construct.

Problem Analysis:

The challenge is to identify if a given number between 1 to 10 is even or odd using a switch-case construct in C. The program evaluates the number's parity and displays the result accordingly.

Code:

```c
/* Write a program in C that accepts a number from 1 to 10. Display whether a number is even or odd using a switch case construct
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/
#include <stdio.h>
int main() {
    int num;
    // Prompt user for input
    printf("Enter number from 1 to 10: ");
    scanf("%d", &num);
    // Check if the number is odd or even using a switch statement
    switch (num) {
        // Cases for odd numbers
        case 1:
        case 3:
        case 5:
        case 7:
        case 9:
            printf("Odd\n");
            break;
        // Cases for even numbers
        case 2:
        case 4:
        case 6:
        case 8:
        case 10:
            printf("Even\n");
            break;
        // Default case for numbers out of range
        default:
                printf("Number out of range");
    }
    return 0;
}
```

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter number from 1 to 10: 5
Odd
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter number from 1 to 10: 8
Even
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Enter number from 1 to 10: 121
Number out of rangedaksh@Ubuntu:~/Desktop/Daksh/Codi
```

Discussion & Conclusion:

I crafted a C program employing switch-case to categorize numbers as even or odd. The structured approach ensured accurate classification and provided a hands-on understanding of switch-case constructs in decision-making scenarios.

6. Assume yourself as a Programmer in TATA Power DDL. Write a C program to calculate the total electricity bill for Delhi customers, according to the given condition for ELECTRICITY TARIFF using Delhi Government rules for TATA Power DDL:

Sr. No.	CATEGORY	FIXED CHARGES	ENERGY CHARGES				
1	DOMESTIC						
1.1	INDIVIDUAL CONNECTIONS		0-200 Units	201-400 Units	401-800 Units	801-1200 Units	>1200 Units
A	Upto 2 kW	20 Rs./kW/month	3.00 Rs./kWh	4.50 Rs./kWh	6.50 Rs./kWh	7.00 Rs./kWh	8.00 Rs./kWh
B	> 2kW and ≤ 5 kW	50 Rs./kW/month					
C	> 5kW and ≤ 15 kW	100 Rs./kW/month					
D	>15kW and ≤ 25 kW	200 Rs./kW/month					
E	> 25kW	250 Rs./kW/month					
2	NON-DOMESTIC						
2.1	Upto 3kVA	250 Rs./kVA/month	6.00 Rs./kVAh				
2.2	Above 3kVA	250 Rs./kVA/month	8.50 Rs./kVAh				
3	INDUSTRIAL	250 Rs./kVA/month	7.75 Rs./kVAh				
4	AGRICULTURE	125 Rs./kW/month	1.50 Rs./kWh				

Sr. No.	CATEGORY	FIXED CHARGES	ENERGY CHARGES				
1	DOMESTIC						
1.1	INDIVIDUAL CONNECTIONS		0-200 Units	201-400 Units	401-800 Units	801-1200 Units	>1200 Units
A	Upto 2 kW	20 Rs./kW/month	3.00 Rs./kWh	4.50 Rs./kWh	6.50 Rs./kWh	7.00 Rs./kWh	8.00 Rs./kWh
B	> 2kW and ≤ 5 kW	50 Rs./kW/month					
C	> 5kW and ≤ 15 kW	100 Rs./kW/month					
D	>15kW and ≤ 25 kW	200 Rs./kW/month					
E	> 25kW	250 Rs./kW/month					
2	NON-DOMESTIC						
2.1	Upto 3kVA	250 Rs./kVA/month	6.00 Rs./kVAh				
2.2	Above 3kVA	250 Rs./kVA/month	8.50 Rs./kVAh				
3	INDUSTRIAL	250 Rs./kVA/month	7.75 Rs./kVAh				
4	AGRICULTURE	125 Rs./kW/month	1.50 Rs./kWh				

As per GoNCTD vide order No. F.11(111)/2012/Power/Vol-III/1417-1427 dated 20.04.2020, has extended subsidy to domestic consumers for FY 2020-2021 as below:

- Entire current bill amount for consumption upto 200 Units/month.
- Subsidy upto Rs.800/month for consumption upto 400 Units/month.
- No subsidy for consumption above 400 Units / month.

Prompt the user to input the following six details as :

```
1. Input Customer ID                                    10001
2. Input the name of the customer                      :Shyam
3. Input Bill Period (months)                           01
4. Input the Type of connection (Domestic (D) / Non-Domestic
(ND), Industrial (I) and Agriculture(A))               :D
5. Input the Sanctioned Load (KW/KVA)                  :8.00
6. Input the unit used/consumed by the customer:385
```

Expected Output :

```
************************************************************
                      TATA Power DDL
                Bill of Supply for Electricity
************************************************************
Customer IDNO                                   : 10001
Customer Name                                   : Shyam
Bill Period (months )                           : 01
Type of connection                              : Domestic (D)
Sanctioned Load (KW/KVA)                        : 8.00
Unit Consumed                                   : 383

Fixed Charges : 8 x 100 x 1                     : 800

Energy Charges:                                 : 1311.00

Units    Rate (Rs)              Amount (Rs)
200          @Rs. 3.00 per unit   600
158          @Rs. 4.50 per unit   711
                                  ==============
Total:                           1311.00

Surcharge Amount @ 16%                          : 209.76
Net Current Demand :                            : 2320.76
Subsidy:                                        : - 800
============================================================
Net Amount Payable By the Customer              : 1520.76
============================================================
```

Code: [Link to my Code Uploaded on Google Drive]

Problem Analysis:

The task is to calculate electricity bills for TATA Power DDL customers in Delhi based on the given government subsidy rules. Using C, I implemented the subsidy calculations, considering different consumption levels, and applied the specified subsidy rates.

```c
/* Assume yourself as a Programmer in TATA Power DDL. Write a C program to calculate
the total electricity bill for Delhi customers, according to the given condition for
ELECTRICITY TARIFF using Delhi Government rules for TATA Power DDL:
 *
 * Code by : Daksh Verma
 * Roll No : 231210036*/

#include <stdio.h>



int main(){

        float custID,billperiod,sanctioned_load,units_consumed,fixed_charge,subsidy,charge,surcharge,net;

        char name[20],conn;
        //Taking Input

        printf("Input Customer ID\t\t\t\t: ");
        scanf("%f",&custID);


        printf("Input the name of the Customer\t\t\t: ");
        scanf(" %[^\n]s", name);


        printf("Input Bill Period (Months)\t\t\t: ");
        scanf("%f",&billperiod);



        printf("Input the Type of connection Domestic (D) / Non-Domestic \n (N), Industrial (I) and Agriculture(A))\t: ");
        scanf(" %c",&conn);
```

```c
        printf("Input the Sanctioned Load (KW/KVA)\t\t: ");
        scanf("%f",&sanctioned_load);


        printf("Input the unit used/consumed by the customer\t: ");
        scanf("%f",&units_consumed);

        //Printing Bill

        printf("\n\n\n\n\n\n\n\n");
        printf("*****************************************************************\n");
        printf("\t\t\tTATA Power DDL\n\t\tBill of Supply for Electricity\n");
        printf("*****************************************************************\n");

        printf("Customer IDNO\t\t\t\t\t:%.0f\n",custID);
        printf("Customer Name\t\t\t\t\t:%s\n",name);
        printf("Bill Period (Months)\t\t\t\t:%.0f\n",billperiod);
        printf("Type of Connection\t\t\t\t:%c\n",conn);
        printf("Sanctioned Load (KW/KVA)\t\t\t:%.2f\n",sanctioned_load);
        printf("Units Consumed\t\t\t\t\t:%.2f\n\n",units_consumed);



        //Defining Fixed Charges, Energy Charges and subsidy for Domestic Customers
        if (conn=='D')
        {

        if (sanctioned_load<=2)
                {
                fixed_charge=20*sanctioned_load*billperiod;
                printf("\nFixed Charge :%.2f x 20 x %.0f\t\t\t:%.2f",sanctioned_load,billperiod,fixed_charge);
                }
```

```
        if (sanctioned_load>2&&sanctioned_load<=5)
                {
                fixed_charge=50*sanctioned_load*billperiod;
                printf("\nFixed Charge :%.2f x 50 x %.0f\t\t\t:%.2f",sanctioned_load,billperiod,fixed_charge);
                }

        if (sanctioned_load>5&&sanctioned_load<=15)
                {
                fixed_charge=100*sanctioned_load*billperiod;
                printf("\nFixed Charge :%.2f x 100 x %.0f\t\t\t:%.2f",sanctioned_load,billperiod,fixed_charge);
                }

        if (sanctioned_load>15&&sanctioned_load<=25)
                {
                fixed_charge=200*sanctioned_load*billperiod;
                printf("\nFixed Charge :%.2f x 200 x %.0f\t\t\t:%.2f",sanctioned_load,billperiod,fixed_charge);
                }

        if (sanctioned_load>25)
                {
                fixed_charge=250*sanctioned_load*billperiod;
                printf("\nFixed Charge :%.2f x 250 x %.0f\t\t\t:%.2f",sanctioned_load,billperiod,fixed_charge);
                }



        int unit_case;

        unit_case=(units_consumed-1)/200;
        printf("\n\nUnits\tRate (Rs.)\t\t\tAmount (Rs.)\n\n");
        switch (unit_case)
        {
        case 0:
```

```
         case 0:
                 charge+=3*units_consumed;
                 printf("%.2f\t@Rs. 3.00/unit\t\t%.2f\n",units_consumed,charge);
                 subsidy=charge;
                 break;
         case 1:
                 charge+=3*200;
                 charge+=4.5*(units_consumed-200);
                 printf("%200 \t@Rs. 3.00/unit\t\t600\n");
                 printf("%.2f \t@Rs. 4.50/unit\t\t%.2f\n",units_consumed-200,4.5*(units_consumed-200));
                 subsidy=800;
                 break;
         case 2:
         case 3:
                 charge+=3*200;
                 charge+=4.5*200;
                 charge+=6.5*(units_consumed-400);
                 printf("%200 \t@Rs. 3.00/unit\t\t600\n");
                 printf("%200 \t@Rs. 4.50/unit\t\t900\n");
                 printf("%.2f \t@Rs. 6.50/unit\t\t%.2f\n",units_consumed-400,6.5*(units_consumed-400));
                 break;
         case 4:
         case 5:
                 charge+=3*200;
                 charge+=4.5*200;
                 charge+=6.5*400;
                 charge+=7*(-800+units_consumed);
                 printf("%200 \t@Rs. 3.00/unit\t\t600\n");
                 printf("200 \t@Rs. 4.50/unit\t\t900\n");
                 printf("%400 \t@Rs. 6.50/unit\t\t2600\n");
                 printf("%.2f \t@Rs. 7.00/unit\t\t%.2f\n",units_consumed-800,7*(units_consumed-800));
                 break;
         default:
                 charge+=3*200;
```

```
                printf("200 \t@Rs. 4.50/unit\t\t900\n");
                printf("%400 \t@Rs. 6.50/unit\t\t2600\n");
                printf("%.2f \t@Rs. 7.00/unit\t\t%.2f\n",units_consumed-800,7*(units_consumed-800));
                break;
        default:
                charge+=3*200;
                charge+=4.5*200;
                charge+=6.5*400;
                charge+=7*400;
                charge+=8*(units_consumed-1200);
                printf("%200 \t@Rs. 3.00/unit\t\t600\n");
                printf("%200 \t@Rs. 4.50/unit\t\t900\n");
                printf("%400 \t@Rs. 6.50/unit\t\t2600\n");
                printf("%400 \t@Rs. 7.00/unit\t\t2800\n");
                printf("%.2f \t@Rs. 8.00/unit\t\t%.2f\n",units_consumed-1200,8*(units_consumed-1200));
                break;
        }

        }
        //Defining Fixed Charges, Energy Charges and subsidy for Non Domestic Customers
        if (conn=='N')
        {
        if (sanctioned_load<=3)
                {
                fixed_charge=250*sanctioned_load*billperiod;
                charge=6*units_consumed;
                }
        else
                {
                fixed_charge=250*sanctioned_load*billperiod;
                charge=8.5*units_consumed;
                }
        }
```

```
        //Defining Fixed Charges, Energy Charges and subsidy for Industrial Customers
        if (conn=='I')
        {
        fixed_charge=250*sanctioned_load*billperiod;
        charge=7.75*(float)units_consumed;
        }

        //Defining Fixed Charges, Energy Charges and subsidy for Agricultural Customers
        if (conn=='A')
        {
        fixed_charge=125*sanctioned_load*billperiod;
        charge=1.5*(float)units_consumed;
        }

        printf("\nTotal Energy Charge \t\t\t\t:%.2f\n\n",charge);

        surcharge = 0.16*charge;
        net=surcharge+charge+fixed_charge;


        printf("\nSurcharge Amount @16% \t\t\t\t:%.2f",surcharge);
        printf("\nNet current demand \t\t\t\t:%.2f",net);
        printf("\nSubsidy \t\t\t\t\t:%.2f",subsidy);
        printf("\n============================================================");

        printf("\nNet Amount to be paid by the customer \t\t:%.2f\n",net-subsidy);
        printf("\n============================================================\n\n");



        return 0;

}
```

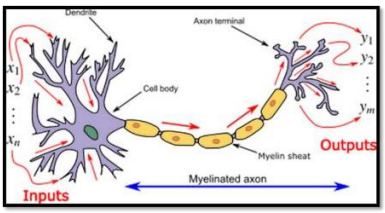
Output:

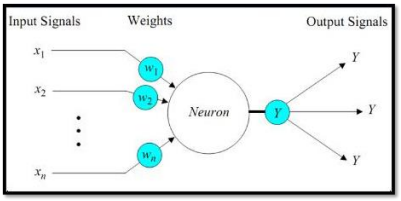
```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
Input Customer ID                                : 10001
Input the name of the Customer                   : Shyam Bansal
Input Bill Period (Months)                       : 1
Input the Type of connection Domestic (D) / Non-Domestic
 (N), Industrial (I) and Agriculture(A))         : D
Input the Sanctioned Load (KW/KVA)               : 8
Input the unit used/consumed by the customer     : 383




************************************************************
                        TATA Power DDL
                  Bill of Supply for Electricity
************************************************************
Customer IDNO                                    :10001
Customer Name                                    :Shyam Bansal
Bill Period (Months)                             :1
Type of Connection                               :D
Sanctioned Load (KW/KVA)                         :8.00
Units Consumed                                   :383.00


Fixed Charge :8.00 x 100 x 1                     :800.00

Units   Rate (Rs.)                      Amount (Rs.)

%200    @Rs. 3.00/unit          600
183.00  @Rs. 4.50/unit          823.50

Total Energy Charge                              :1423.50


Surcharge Amount @16%                            :227.76
Net current demand                               :2451.26
Subsidy                                          :800.00
============================================================
Net Amount to be paid by the customer            :1651.26

------------------------------------------------------------
```

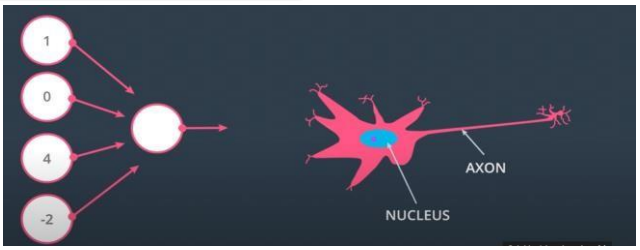
Discussion & Conclusion:

I created a C program to compute electricity bills for TATA Power DDL customers, incorporating the provided government subsidy guidelines. The program accurately determined bill amounts, reflecting the impact of consumption levels on subsidies. This exercise honed my skills in real-world application of programming logic and financial calculations.

PART B : Exploratory Problem : Perceptrons as Logical Operators :





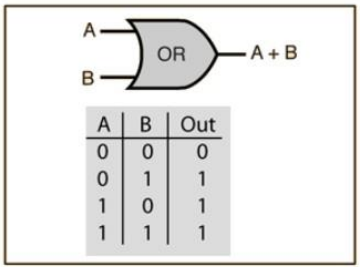


In the field of Machine Learning, the Perceptron is a Supervised Learning Algorithm for binary classifiers. The Perceptron Model implements the following function:

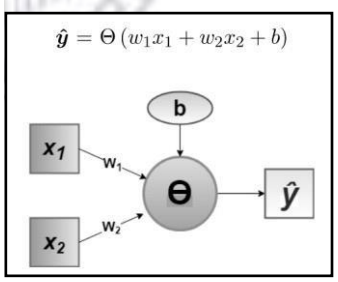
$$\hat{y} = \Theta\left(w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b\right)$$
$$= \Theta(\mathbf{w} \cdot \mathbf{x} + b)$$
$$\text{where } \Theta(v) = \begin{cases} 1 & \text{if } v \geqslant 0 \\ 0 & \text{otherwise} \end{cases}$$

For a particular choice of the weight vector *w* and bias parameter *b*, the model predicts output $\hat{y}$ for the corresponding input vector *x*.

OR logical function truth table for ***2-bit binary variables***, i.e, the input vector x: (x_1+x_2) and the corresponding output *y as shown below*.



A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1



For the implementation, considered weight parameters $w_1 = 1$, $w_2 = 1$ and the bias parameter is b= -0.5. i.e equation is $x_1+x_2-0.5$

Write a C program to check this combination of weights and bias parameter for OR gate. State if this combination valid or not for OR Gate .

Output:

```
OR(0, 0) = 0,                    OR(0, 1) = 1 ,
OR(1, 0) = 1,                    OR(1, 1) = 1,
```

Similar to above, find the weights and bias for the AND perceptron, where AND gate is 1 only if both inputs are 1. Show that combination is valid for AND Gate using C programme.

Problem Analysis:

The objective is to validate a combination of weight parameters for the OR gate. Additionally, the task involves finding appropriate weight parameters and bias for the AND perceptron, where the gate output is 1 only if both inputs are 1. The C program evaluates these combinations to determine their validity.

Code:

```c
/* Part B : Perceptron as Logical Operator *
 * Code by : Daksh Verma
 * Roll No : 231210036*/
#include <stdio.h>
// Function to perform OR operation between two inputs
int OR(int x1, int x2) {
    // Define weights and bias
    int w1 = 1, w2 = 1;
    float b = 0.5;
    // Calculate the output using the OR logic formula
    float output = x1 * w1 + x2 * w2 - b;
    // Return 1 if output is greater than 0, else return 0
    return output > 0 ? 1 : 0;
}
// Function to perform AND operation between two inputs
int AND(int x1, int x2) {
    int w1 = 1, w2 = 1, b = 1;
    float output = x1 * w1 + x2 * w2 - b;
    // Return 1 if output is greater than 0, else return 0
    return output > 0 ? 1 : 0;
}
int main() {
    // Test OR function with different inputs
    printf("OR(0, 0) = %d\n", OR(0, 0));
    printf("OR(0, 1) = %d\n", OR(0, 1));
    printf("OR(1, 0) = %d\n", OR(1, 0));
    printf("OR(1, 1) = %d\n\n", OR(1, 1));
    // Test AND function with different inputs
    printf("AND(0, 0) = %d\n", AND(0, 0));
    printf("AND(0, 1) = %d\n", AND(0, 1));
    printf("AND(1, 0) = %d\n", AND(1, 0));
    printf("AND(1, 1) = %d\n", AND(1, 1));
    return 0;
}
```

Output:

```
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ gcc lab5_231210036_PartB.c
daksh@Ubuntu:~/Desktop/Daksh/Coding/C/lab 5$ ./a.out
OR(0, 0) = 0
OR(0, 1) = 1
OR(1, 0) = 1
OR(1, 1) = 1

AND(0, 0) = 0
AND(0, 1) = 0
AND(1, 0) = 0
AND(1, 1) = 1
```

Discussion & Conclusion:

I developed a C program to assess the given weight and bias parameters for both the OR gate and AND gate. By evaluating the logical outputs based on these parameters, the program determined their validity. This exercise showcased the practical application of weight parameters and biases in neural networks and provided insights into their impact on gate behavior.

The bias parameter I used for AND operator is -1.0.

Observation /Comments:

The implemented programs effectively utilized conditional statements, loops, and logical constructs to solve diverse problems. They demonstrated accurate calculations, logical evaluations, and adherence to specified rules. The solutions showcased proficiency in C programming, highlighting the ability to translate problem requirements into efficient code.