

## EXPERIMENT NO 1

### CREATION OF DATAWAREHOUSE

```
MariaDB [supermarket]> create database test1;  
Query OK, 1 row affected (0.01 sec)
```

```
MariaDB [supermarket]> create table demo1  
-> ( customer varchar(10) PRIMARY KEY,  
-> name varchar(10) UNIQUE);  
Query OK, 0 rows affected (0.29 sec)
```

```
MariaDB [supermarket]> create table demo2  
-> (customer varchar(10),  
-> sales int NOT NULL);  
Query OK, 0 rows affected (0.52 sec)
```

### \*FOREIGN KEY\*

```
MariaDB [supermarket]> alter table demo2  
-> add foreign key(customer) references demo1(customer);  
Query OK, 0 rows affected (0.88 sec)  
Records: 0 Duplicates: 0 Warnings: 0 Records: 0 Duplicates: 0 Warnings: 0
```

### \*NATURAL JOIN\*

```
MariaDB [supermarket]> select * from sales natural join customer  
-> where sales.Customer_Id=customer.Customer_Id;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
---+-----+-----+-----+-----+-----+-----+-----+-----+  
---+                                     | Customer_Id |  
Product_Id | Date_Id | Store_Id | Units | TotalSales | TotalCost | Profit | First_Name |  
Last_Name | Street | City | State | Zip | Country |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+-----+  
| C01 | P06 | D150119 | S03 | 3 | 291 | 270 | 21 |  
Divyang | Oza | Ghodbunder Road | Thane | Maharashtra | 400607 | India |  
| C07 | P17 | D250818 | S01 | 2 | 296 | 240 | 56 |  
Poushali | Khan | P.O. Box 122, 5 | Chandigarh | Chandigarh | 381255 | India |  
| C16 | P24 | D150119 | S02 | 1 | 319 | 250 | 79 |  
Sumitra | Kumar | 987-1613 Purus. | Port Blair | Andaman and Nic | 107703 | India |  
| C47 | P12 | D220419 | S04 | 5 | 525 | 475 | 50 |  
Shatrunjay | Jain | 3418 Ipsum. Av. | Daman | Daman and Diu | 610207 | India |  
| C18 | P05 | D121118 | S06 | 2 | 280 | 240 | 40 |  
Akroor | Khan | 726-493 Ac, Ave | Imphal | Manipur | 463007 | India |  
| C11 | P11 | D220419 | S06 | 6 | 570 | 498 | 72 |  
Prasata | Jain | Ap #319-5966 Mo | Shimla | Himachal Prades | 571315 | India |  
| C18 | P18 | D220419 | S01 | 10 | 1650 | 1400 | 250 |  
Akroor | Khan | 726-493 Ac, Ave | Imphal | Manipur | 463007 | India |  
| C07 | P07 | D150119 | S06 | 8 | 3960 | 3600 | 360 |  
Poushali | Khan | P.O. Box 122, 5 | Chandigarh | Chandigarh | 381255 | India |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+-----+  
8 rows in set (0.00 sec)
```

### **\*DISTINCT\***

```
MariaDB [supermarket]> select distinct customer.First_Name,customer.Last_Name from customer
where customer_id='C01';
```

```
+-----+-----+
| First_Name | Last_Name |
+-----+-----+
| Divyang    | Oza       |
+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [supermarket]> select distinct sales.Customer_Id,
customer.First_Name,customer.Last_Name,SUM(sales.TotalSales) from customer, sales where
sales.Customer_Id=customer.Customer_Id Group By sales.Customer_Id;
```

```
+-----+-----+-----+-----+
| Customer_Id | First_Name | Last_Name | SUM(sales.TotalSales) |
+-----+-----+-----+-----+
| C01          | Divyang    | Oza       | 291                    |
| C07          | Poushali   | Khan      | 4256                   |
| C11          | Prasata    | Jain      | 570                    |
| C16          | Sumitra    | Kumar     | 319                    |
| C18          | Akroor     | Khan      | 1930                   |
| C47          | Shatrunjay | Jain      | 525                    |
+-----+-----+-----+-----+
6 rows in set (0.03 sec)
```

### **\*JOIN\***

```
MariaDB [supermarket]> select Product_Name,Price from product cross join sales where
product.Product_Id=sales.Product_Id;
```

```
+-----+-----+
| Product_Name          | Price |
+-----+-----+
| Refined Sunflower Oil | 140   |
| Refined Soyabean Oil  | 97    |
| Refined Soyabean Oil Jar | 495   |
| Aloo Mutter           | 95    |
| Palak Paneer          | 105   |
| BRU Instant Coffee, 100g | 148   |
| BRU Green Label Coffee | 165   |
| Tetley Green Tea, Lemon and Ho | 319   |
+-----+-----+
8 rows in set (0.22 sec)
```

### **\*GROUP BY\***

```
MariaDB [supermarket]> select distinct sales.Customer_Id, customer.First_Name,time.Year from
sales,customer,time where Year='2018' AND sales.Customer_Id=customer.Customer_ID Group By
sales.Customer_Id, customer.First_Name;
```

```
+-----+-----+-----+
| Customer_Id | First_Name | Year |
+-----+-----+-----+
| C01          | Divyang    | 2018 |
| C07          | Poushali   | 2018 |
+-----+-----+-----+
```

C11	Prasata	2018
C16	Sumitra	2018
C18	Akroor	2018
C47	Shatrunjay	2018

+-----+-----+-----+

6 rows in set (0.12 sec)

MariaDB [supermarket]> select customer.First\_Name, time.Date, sales.TotalSales from customer, time, sales where customer.Customer\_Id=sales.Customer\_Id and sales.Date\_Id=time.Date\_Id and Profit > 200;

+-----+-----+-----+

First_Name	Date	TotalSales
Poushali	2019-01-15	3960
Akroor	2019-04-22	1650

+-----+-----+-----+

2 rows in set (0.00 sec)

### **\*INNER JOIN\***

MariaDB [supermarket]> select distinct customer.First\_Name, customer.Last\_Name,sales.TotalSales,sales.Date\_Id from customer inner join sales on customer.Customer\_Id=sales.Customer\_Id;

+-----+-----+-----+-----+

First_Name	Last_Name	TotalSales	Date_Id
Divyang	Oza	291	D150119
Poushali	Khan	296	D250818
Sumitra	Kumar	319	D150119
Shatrunjay	Jain	525	D220419
Akroor	Khan	280	D121118
Prasata	Jain	570	D220419
Akroor	Khan	1650	D220419
Poushali	Khan	3960	D150119

+-----+-----+-----+-----+

8 rows in set (0.00 sec)

## EXPERIMENT NO 2

### OLAP OPERATIONS

#### **\*ROLLUP\***

Rollup operations on profit earned on each product at following cities

MariaDB [supermarket]> select distinct customer.City, product.Product\_Name,SUM(Profit) from customer, sales, product where sales.Customer\_Id=customer.Customer\_Id and product.Product\_Id=sales.Product\_Id group by customer.City, product.Product\_Name with Rollup;

City	Product_Name	SUM(Profit)
Chandigarh	BRU Instant Coffee, 100g	56
Chandigarh	Refined Soyabean Oil Jar	360
Chandigarh	NULL	416
Daman	Palak Paneer	50
Daman	NULL	50
Imphal	BRU Green Label Coffee	250
Imphal	Refined Sunflower Oil	40
Imphal	NULL	290
Port Blair	Tetley Green Tea, Lemon and Ho	79
Port Blair	NULL	79
Shimla	Aloo Mutter	72
Shimla	NULL	72
Thane	Refined Soyabean Oil	21
Thane	NULL	21
NULL	NULL	928

15 rows in set (0.01 sec)

Rollup Operations on Profits earned from each customer during time quarters

MariaDB [supermarket]> select customer.Customer\_Id, customer.First\_Name, time.quarter, SUM(Profit) from sales,customer,time where sales.Customer\_Id=customer.Customer\_Id and time.Date\_Id=sales.Date\_Id group by customer.Customer\_Id, customer.First\_Name, time.quarter ASC with Rollup;

Customer_Id	First_Name	quarter	SUM(Profit)
C01	Divyang	Q1	21
C01	Divyang	NULL	21
C01	NULL	NULL	21
C07	Poushali	Q1	360
C07	Poushali	Q3	56
C07	Poushali	NULL	416
C07	NULL	NULL	416
C11	Prasata	Q2	72
C11	Prasata	NULL	72
C11	NULL	NULL	72
C16	Sumitra	Q1	79
C16	Sumitra	NULL	79
C16	NULL	NULL	79
C18	Akroor	Q2	250
C18	Akroor	Q4	40
C18	Akroor	NULL	290

C18	NULL	NULL	290
C47	Shatrunjay	Q2	50
C47	Shatrunjay	NULL	50
C47	NULL	NULL	50
NULL	NULL	NULL	928

+-----+-----+-----+-----+

21 rows in set (0.00 sec)

### Rollup operations on profit earned by each branch from each store during time quarters

MariaDB [supermarket]> select store.Store\_Name, branch.Branch\_Name, time.quarter,SUM(Profit) from sales, branch,store, time where sales.Store\_Id=store.Store\_Id and sales.Date\_Id=time.Date\_Id and branch.Branch\_Id=store.Branch\_Id group by store.Store\_Name, branch.Branch\_Name, time.quarter with Rollup;

Store_Name	Branch_Name	quarter	SUM(Profit)
DMart	Kasarwadvali	Q1	79
DMart	Kasarwadvali	NULL	79
DMart	Kolshet	Q2	250
DMart	Kolshet	Q3	56
DMart	Kolshet	NULL	306
DMart	Lalbaug	Q2	50
DMart	Lalbaug	NULL	50
DMart	Mulund	Q1	21
DMart	Mulund	NULL	21
DMart	NULL	NULL	456
Star Bazaar	Lalbaug	Q1	360
Star Bazaar	Lalbaug	Q2	72
Star Bazaar	Lalbaug	Q4	40
Star Bazaar	Lalbaug	NULL	472
Star Bazaar	NULL	NULL	472
NULL	NULL	NULL	928

+-----+-----+-----+-----+

16 rows in set (0.00 sec)

### **\*SLICE\***

#### Slice operation to get the number of products sold in time quarter

MariaDB [supermarket]> select count(units) from sales,customer,product,store,time where sales.Customer\_Id=customer.Customer\_Id and product.Product\_Id=sales.Product\_Id and sales.Date\_Id=time.Date\_Id and time.Quarter='Q1';

count(units)
18

1 row in set (0.04 sec)

#### Slice operation to get the name of cities sold from all stores on Monday

MariaDB [supermarket]> select distinct sales.Customer\_Id,customer.First\_Name,customer.Last\_Name,time.Date from sales,customer,product,store,time where sales.Customer\_Id=customer.Customer\_Id and product.Product\_Id=sales.Product\_Id and sales.Date\_Id=time.Date\_Id and time.Day='MON';

Customer_Id	First_Name	Last_Name	Date
-------------	------------	-----------	------

Customer ID	Customer Name	Branch	Date
C47	Shatrunjay	Jain	2019-04-22
C18	Akroor	Khan	2018-11-12
C11	Prasata	Jain	2019-04-22
C18	Akroor	Khan	2019-04-22

4 rows in set (0.00 sec)

Slice operation to get the name of the customers who bought certain products

MariaDB [supermarket]> select distinct customer.Last\_Name, customer.First\_Name, Units as Units\_Bought from customer,sales,product,time,branch,store where customer.Customer\_Id=sales.Customer\_Id and sales.Date\_Id=time.Date\_Id and branch.Branch\_Id=store.Branch\_Id and product.Product\_Id=sales.Product\_Id and Product.Product\_Name in ("Refined Soyabean Oil","Palak Paneer","Tetley Green Tea, Lemon and Ho");

Last_Name	First_Name	Units_Bought
Oza	Divyang	3
Kumar	Sumitra	1
Jain	Shatrunjay	5

3 rows in set (0.00 sec)

**\*DICE\***

Dice operation to get the count of units sold to customer named Divyang,Sumitra,Akroor from branch Thane & Lalbaug

MariaDB [supermarket]> select count(Units) as Units\_Sold from customer,sales,product,time,branch,store where customer.Customer\_Id=sales.Customer\_Id and sales.Date\_Id=time.Date\_Id and branch.Branch\_Id=store.Branch\_Id and product.Product\_Id=sales.Product\_Id and customer.First\_Name in ("Divyang","Akroor","Sumitra") and branch.City in ("Thane","Lalbaug");

Units_Sold
16

1 row in set (0.00 sec)

Dice operation to get the Customer names who bought on Monday or Saturday with certain customer IDs.

MariaDB [supermarket]> select distinct customer.First\_Name, customer.Last\_Name, TotalSales,time.day from customer,sales,product,time,branch,store where customer.Customer\_Id=sales.Customer\_Id and sales.Date\_Id=time.Date\_Id and branch.Branch\_Id=store.Branch\_Id and product.Product\_Id=sales.Product\_Id and customer.Customer\_ID in ("C01","C47","C07","C11") and time.Day in ("MON","SAT");

First_Name	Last_Name	TotalSales	day
Poushali	Khan	296	SAT
Shatrunjay	Jain	525	MON
Prasata	Jain	570	MON

3 rows in set (0.00 sec)

Dice operation to find out customers who bought certain products in month January

```
MariaDB [supermarket]> select distinct customer.Last_Name, customer.First_Name, Units as  
Units_Bought from customer,sales,product,time,branch,store where  
customer.Customer_Id=sales.Customer_Id and sales.Date_Id=time.Date_Id and  
branch.Branch_Id=store.Branch_Id and product.Product_Id=sales.Product_Id and  
Product.Product_Name in ("Refined Soyabean Oil","Palak Paneer","Tetley Green Tea, Lemon and  
Ho") and time.Month in ("JAN");
```

```
+-----+-----+-----+  
| Last_Name | First_Name | Units_Bought |  
+-----+-----+-----+  
| Oza      | Divyang   | 3           |  
| Kumar    | Sumitra   | 1           |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

**\*PIVOT\***

Pivot operation to find out customers who bought on Monday, Tuesday, Saturday with certain Customer Id.

```
MariaDB [supermarket]> select distinct time.day,TotalSales,Profit, customer.Last_Name,  
customer.First_Name from customer,sales,product,time,branch,store where  
customer.Customer_Id=sales.Customer_Id and sales.Date_Id=time.Date_Id and  
branch.Branch_Id=store.Branch_Id and product.Product_Id=sales.Product_Id and  
customer.Customer_ID in ("C01","C47","C07","C11") and time.Day in ("MON","SAT","TUE");
```

```
+-----+-----+-----+-----+-----+  
| day | TotalSales | Profit | Last_Name | First_Name |  
+-----+-----+-----+-----+-----+  
| TUE | 291 | 21 | Oza | Divyang |  
| SAT | 296 | 56 | Khan | Poushali |  
| MON | 525 | 50 | Jain | Shatrunjay |  
| MON | 570 | 72 | Jain | Prasata |  
| TUE | 3960 | 360 | Khan | Poushali |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

Pivot Operation to find out customers who bought certain products.

```
MariaDB [supermarket]> select distinct time.Date,time.day,TotalSales, customer.Last_Name,  
customer.First_Name from customer,sales,product,time,branch,store where  
customer.Customer_Id=sales.Customer_Id and sales.Date_Id=time.Date_Id and  
branch.Branch_Id=store.Branch_Id and product.Product_Id=sales.Product_Id and  
customer.Customer_ID in ("C01","C47","C07","C11") and Product.Product_Name in ("Refined  
Soyabean Oil","Palak Paneer","Tetley Green Tea, Lemon and Ho");
```

```
+-----+-----+-----+-----+-----+  
| Date | day | TotalSales | Last_Name | First_Name |  
+-----+-----+-----+-----+-----+  
| 2019-01-15 | TUE | 291 | Oza | Divyang |  
| 2019-04-22 | MON | 525 | Jain | Shatrunjay |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

### EXPERIMENT NO. 3

#### Decision Tree

```
#include <stdio.h>
#include<string.h>
#include<math.h>
float entrophy(int ayes,int ano,int atotal){           //for entrophy calculation formula -
(pi/total)*log2(pi/total);
    float temp=-((ayes/(float)atotal)*log2(ayes/(float)atotal))-
((ano/(float)atotal)*log2(ano/(float)atotal));
    return temp;
}
int main(int argc, char **argv)
{
    FILE *input;

    char outlook[20],temp[20],humidity[20],windy[20],play[20];           //file operations
    int temp1=0;
    float maxGain=0;           //calculate maximum gain to
find selected nodes

    int totalplay=0;           //for play class YES/NO
    float EntrophyPlay=0;

    float EntrophyWindyTrue=0,EntrophyWindyFalse=0;           //WINDY
    int totalWindyFalse=0,totalWindyTrue=0;

    float EntrophyHumidityHigh=0,EntrophyHumidityNormal=0;           //HUMIDITY
    int totalHumidityHigh=0,totalHumidityNormal=0;

    float EntrophySunny=0,EntrophyOvercast=0,EntrophyRainy=0;           //OUTLOOK
    int totalSunny=0,totalOvercast=0,totalRainy=0;

    float EntrophyHot=0,EntrophyMild=0,EntrophyCool=0;           //TEMPERATURE
    int totalHot=0,totalMild=0,totalCool=0;

    float totalOutlook=0,totalTemp=0,totalHumidity=0,totalWindy=0;
    float gainOutlook=0,gainTemp=0,gainHumidity=0,gainWindy=0;

    //Total Yes or No Of Play
    printf("\n\nThe Play Class\n\n");

    int yes=0,no=0;
    input=fopen("decisiontreeinput.txt","r");
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

    while(!feof(input)){
        fscanf(input,"%s",play);

        if(strcmp(play,"yes")==0){
            yes++;
        }
    }
}
```



```

        totalplay++;
    }
    if(strcmp(play,"no")==0){
        no++;totalplay++; }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number of yes is %d\n",yes);
printf("The Number of no is %d\n",no);
printf("The Number of TOTAL PLAY is %d\n",totalplay);
EntropyPlay=entropy(yes,no,totalplay);
printf("The Entrophy is %f\n",EntropyPlay);

// Now Calculating For OUTLOOK Sunny(Yes NO);
printf("\n\nThe Outlook Sunny \n\n"); //same as playclass just to add extra compare and

yes=0;no=0;totalSunny=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(outlook,"sunny")==0){
        yes++;
        totalSunny++;
    }
    if(strcmp(play,"no")==0 && strcmp(outlook,"sunny")==0){
        no++;totalSunny++; }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number of SUNNY yes is %d\n",yes);
printf("The Number of SUNNY no is %d\n",no);
printf("The Number of SUNNY TOTAL is %d\n",totalSunny);
EntropySunny=entropy(yes,no,totalSunny); //function CALL

temp1=isnan(EntropySunny); //to calculate ISNAN
because log(0)*0 gives ERROR NAN(not a number)
if(temp1==1)
    EntropySunny=0;

printf("The Entrophy of SUNNY is %f\n",EntropySunny);

// Now Calculating For OUTLOOK Overcast(Yes NO);
printf("\n\nThe Outlook Overcast \n\n");
yes=0;no=0;totalOvercast=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

```

```

        if(strcmp(play,"yes")==0 && strcmp(outlook,"overcast")==0){
            yes++;
            totalOvercast++;
        }
        if(strcmp(play,"no")==0 && strcmp(outlook,"overcast")==0){
            no++;totalOvercast++;
        }
        fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

    }
    printf("The Number oF OVERCAST yes is %d\n",yes);
    printf("The Number oF OVERCAST no is %d\n",no);
    printf("The Number oF OVERCAST TOTAL is %d\n",totalOvercast);
    EntrophyOvercast=entropy(yes,no,totalOvercast);

    temp1=isnan(EntrophyOvercast);
    if(temp1==1)
        EntrophyOvercast=0;

    printf("The Entrophy of OVERCAST is %f\n",EntrophyOvercast);

// Now Calculating For OUTLOOK Rainy(Yes NO);
printf("\n\nThe Outlook Rainy \n\n");
yes=0;no=0;totalRainy=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(outlook,"rainy")==0){
        yes++;
        totalRainy++;
    }
    if(strcmp(play,"no")==0 && strcmp(outlook,"rainy")==0){
        no++;totalRainy++;
    }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF Rainy yes is %d\n",yes);
printf("The Number oF Rainy no is %d\n",no);
printf("The Number oF Rainy TOTAL is %d\n",totalRainy);
EntrophyRainy=entropy(yes,no,totalRainy);

temp1=isnan(EntrophyRainy);
if(temp1==1)
    EntrophyRainy=0;

printf("The Entrophy of Rainy is %f\n",EntrophyRainy);

//Average and Gain of Outlook

printf("\nCalculate the average Entrophy information of Outlook\n");

```

```

totalOutlook=((totalSunny/(float)totalplay)*(EntropySunny))+((totalOvercast/(float)totalplay)
)*(EntropyOvercast))+((totalRainy/(float)totalplay)*(EntropyRainy));
printf("Average Entropy of Outlook is %f\n",totalOutlook);
gainOutlook=EntropyPlay-totalOutlook;
printf("Gain of Outlook is %f\n",gainOutlook);

```

```

// Now Calculating For Windy(Yes NO)   True;
printf("\n\nThe Windy of TRue \n\n");

```

```

yes=0;no=0;totalWindyTrue=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(windy,"true")==0){
        yes++;
        totalWindyTrue++;
    }
    if(strcmp(play,"no")==0 && strcmp(windy,"true")==0){
        no++;totalWindyTrue++;
    }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);
}
printf("The Number of Windy yes is %d\n",yes);
printf("The Number of Windy no is %d\n",no);
printf("The Number of TOTAL Windy is %d\n",totalWindyTrue);
EntropyWindyTrue=entropy(yes,no,totalWindyTrue);

temp1=isnan(EntropyWindyTrue);
if(temp1==1)
    EntropyWindyTrue=0;

printf("The Entropy Wiindy True is %f\n",EntropyWindyTrue);

```

```

// Now Calculating For Windy(Yes NO)   False;
printf("\n\nThe Windy False\n\n");

```

```

yes=0;no=0;totalWindyFalse=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(windy,"false")==0){
        yes++;
        totalWindyFalse++;
    }
}

```

```

        if(strcmp(play,"no")==0 && strcmp(windy,"false")==0){
            no++;totalWindyFalse++; }
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF Windy FALSE yes is %d\n",yes);
printf("The Number oF Windy falseno is %d\n",no);
printf("The Number oF TOTAL Windy False is %d\n",totalWindyFalse);
EntropyWindyFalse=entropy(yes,no,totalWindyFalse);

temp1=isnan(EntropyWindyFalse);
if(temp1==1)
    EntropyWindyFalse=0;

printf("The Entrophy Windy False is %f\n",EntropyWindyFalse);

```

//Average and Gain of Windy

```

printf("\nCalculate the average Entrophy information of Windy\n");

totalWindy=((totalWindyFalse/(float)totalplay)*(EntropyWindyFalse))+((totalWindyTrue/(float)
totalplay)*(EntropyWindyTrue));
printf("Average Entropy of Outlook is %f\n",totalWindy);
gainWindy=EntropyPlay-totalWindy;
printf("Gain of Outlook is %f\n",gainWindy);

```

// Now Calculating For Windy(Yes NO) True;

```

printf("\n\nTHE Windy \n\n");

```

```

yes=0;no=0;totalWindyTrue=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(windy,"true")==0){
        yes++;
        totalWindyTrue++; }
    if(strcmp(play,"no")==0 && strcmp(windy,"true")==0){
        no++;totalWindyTrue++; }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF Windy yes is %d\n",yes);
printf("The Number oF Windy no is %d\n",no);
printf("The Number oF TOTAL Windy is %d\n",totalWindyTrue);
EntropyWindyTrue=entropy(yes,no,totalWindyTrue);

```

```
temp1=isnan(EntropyWindyTrue);
if(temp1==1)
    EntropyWindyTrue=0;

printf("The Entropy Wiindy True is %f\n",EntropyWindyTrue);
```

```
// Now Calculating For Humidity(Normal,high)    Normal;
printf("\n\nThe Humudity Normal\n\n");
```

```
yes=0;no=0;totalHumidityNormal=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(humidity,"normal")==0){
        yes++;
        totalHumidityNormal++;
    }
    if(strcmp(play,"no")==0 && strcmp(humidity,"normal")==0){
        no++;totalHumidityNormal++;
    }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);
}
printf("The Number oF Humidity NORMAL yes is %d\n",yes);
printf("The Number oF Humidity NORMALno is %d\n",no);
printf("The Number oF TOTAL Humidity NORMAL is %d\n",totalHumidityNormal);
EntropyHumidityNormal=entropy(yes,no,totalHumidityNormal);

temp1=isnan(EntropyHumidityNormal);
if(temp1==1)
    EntropyHumidityNormal=0;

printf("The Entropy of HUMIDITY Normal is %f\n",EntropyHumidityNormal);
```

```
// Now Calculating For Humidity(Normal,high)    high;
printf("\n\nThe Humudity HIGH\n\n");
```

```
yes=0;no=0;totalHumidityHigh=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(humidity,"high")==0){
        yes++;
        totalHumidityHigh++;
    }
    if(strcmp(play,"no")==0 && strcmp(humidity,"high")==0){
        no++;totalHumidityHigh++;
    }
}
```

```

fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF humidity High yes is %d\n",yes);
printf("The Number oF Humidity High no is %d\n",no);
printf("The Number oF TOTAL HUMIDITY HIGH is %d\n",totalHumidityHigh);
EntropyHumidityHigh=entropy(yes,no,totalHumidityHigh);

temp1=isnan(EntropyHumidityHigh);
if(temp1==1)
    EntropyHumidityHigh=0;

printf("The Entrophy OF HUMDITY HIGH is %f\n",EntropyHumidityHigh);


//Average and Gain of Humidity

printf("\nCalculate the average Entrophy information of Windy\n");

totalHumidity=((totalHumidityNormal/(float)totalplay)*(EntropyHumidityNormal))+((totalHumidityHigh/(float)totalplay)*(EntropyHumidityHigh));
printf("Average Entropy of HUMIDITY is %f\n",totalHumidity);
gainHumidity=EntropyPlay-totalHumidity;
printf("Gain of HUMIDITY is %f\n",gainHumidity);


// Now Calculating For Temp Hot(Yes NO);
printf("\n\nThe Temp Hot \n\n");

yes=0;no=0;totalHot=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(temp,"hot")==0){
        yes++;
        totalHot++;
    }
    if(strcmp(play,"no")==0 && strcmp(temp,"hot")==0){
        no++;totalHot++;
    }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF Hot yes is %d\n",yes);
printf("The Number oF Hot no is %d\n",no);
printf("The Number oF TOTAL Hot is %d\n",totalHot);
EntropyHot=entropy(yes,no,totalHot);

temp1=isnan(EntropyHot);
if(temp1==1)

```

```

        EntrophyHot=0;

printf("The HOT Entrophy is %f\n",EntrophyHot);


// Now Calculating For Temp MILD(Yes NO);
printf("\n\nThe Temp MILD \n\n");
yes=0;no=0;totalMild=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(temp,"mild")==0){
        yes++;
        totalMild++;
    }
    if(strcmp(play,"no")==0 && strcmp(temp,"mild")==0){
        no++;totalMild++; }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF MILD yes is %d\n",yes);
printf("The Number oF MILD no is %d\n",no);
printf("The Number oF TOTAL MILD is %d\n",totalMild);
EntrophyMild=entropy(yes,no,totalMild);

temp1=isnan(EntrophyMild);
if(temp1==1)
    EntrophyMild=0;

printf("The Entrophy MILD is %f\n",EntrophyMild);


// Now Calculating For Temp Cool(Yes NO);
printf("\n\nThe Temp Cool \n\n");
yes=0;no=0;totalCool=0;temp1=0;
input=fopen("decisiontreeinput.txt","r");
fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

while(!feof(input)){
    fscanf(input,"%s",play);

    if(strcmp(play,"yes")==0 && strcmp(temp,"cool")==0){
        yes++;
        totalCool++;
    }
    if(strcmp(play,"no")==0 && strcmp(temp,"cool")==0){
        no++;totalCool++; }
    fscanf(input,"%s%s%s%s",outlook,temp,humidity,windy);

}
printf("The Number oF COLD yes is %d\n",yes);
printf("The Number oF COLD no is %d\n",no);
printf("The Number oF TOTAL COOL is %d\n",totalCool);

```

[illegible]



```

        printf("WINDY");    }
    else{
        if(gainTemp>gainWindy)
            printf("WINDY");
        else
            printf("Temp");
    }
}
if(maxGain==gainWindy){
    printf("Gain Windy is node");}

    return 0;
}

```

### **Decisiontreeinput.txt**

```

outlook temp. humidity windy play
sunny hot high false no
sunny hot high true no
overcast hot high false yes
rainy mild high false yes
rainy cool normal false yes
rainy cool normal true no
overcast cool normal true yes
sunny mild high false no
sunny cool normal false yes
rainy mild normal false yes
sunny mild normal true yes
overcast mild high true yes
overcast hot normal false yes
rainy mild high true no

```

### **OUTPUT**

```

divyang@Divyang:~/DWM$ gcc decisiontree.c -lm
divyang@Divyang:~/DWM$ ./a.out

```

#### **The Play Class**

```

The Number oF yes is 9
The Number oF no is 5
The Number oF TOTAL PLAY is 14
The Entrophy is 0.940286

```

#### **The Outlook Sunny**

```

The Number oF SUNNY yes is 2
The Number oF SUNNY no is 3
The Number oF SUNNY TOTAL is 5
The Entrophy of SUNNY is 0.970951

```

The Outlook Overcast

The Number of OVERCAST yes is 4

The Number of OVERCAST no is 0

The Number of OVERCAST TOTAL is 4

The Entropy of OVERCAST is 0.000000

The Outlook Rainy

The Number of Rainy yes is 3

The Number of Rainy no is 2

The Number of Rainy TOTAL is 5

The Entropy of Rainy is 0.970951

Calculate the average Entropy information of Outlook

Average Entropy of Outlook is 0.693536

Gain of Outlook is 0.246750

The Windy of True

The Number of Windy yes is 3

The Number of Windy no is 3

The Number of TOTAL Windy is 6

The Entropy of Windy True is 1.000000

The Windy False

The Number of Windy False yes is 6

The Number of Windy False no is 2

The Number of TOTAL Windy False is 8

The Entropy of Windy False is 0.811278

Calculate the average Entropy information of Windy

Average Entropy of Outlook is 0.892159

Gain of Outlook is 0.048127

The Windy

The Number of Windy yes is 3

The Number of Windy no is 3

The Number of TOTAL Windy is 6

The Entropy of Windy True is 1.000000

The Humidity Normal

The Number of Humidity NORMAL yes is 6

The Number of Humidity NORMAL no is 1

The Number of TOTAL Humidity NORMAL is 7

The Entrophy of HUMIDITY Normal is 0.591673

THE Humudity HIGH

The Number oF humidity High yes is 3  
The Number oF Humidity High no is 4  
The Number oF TOTAL HUMIDITY HIGH is 7  
The Entrophy OF HUMDITY HIGH is 0.985228

Calculate the average Entrophy information of Windy  
Average Entropy of HUMIDITY is 0.788450  
Gain of HUMIDITY is 0.151836

THE Temp Hot

The Number oF Hot yes is 2  
The Number oF Hot no is 2  
The Number oF TOTAL Hot is 4  
The HOT Entrophy is 1.000000

THE Temp MILD

The Number oF MILD yes is 4  
The Number oF MILD no is 2  
The Number oF TOTAL MILD is 6  
The Entrophy MILD is 0.918296

THE Temp Cool

The Number oF COLD yes is 3  
The Number oF COLD no is 1  
The Number oF TOTAL COOL is 4  
The Entrophy is 0.811278

Calculate the average Entrophy information of Temp  
Average Entropy of Temp is 0.911063  
Gain of Temp is 0.029223  
Maximum Gain is 0.246750  
Gain Outlook is node

		OutLook	
/(sunny)		(overcast)	\(rainy)
HUmidity		YES	WINDY
/(high)	\(normal)	/(false)	\(true)
NO	YES	YES	NO

## EXPERIMENT NO 4

### APRIORI

```
#include <stdio.h>

int main(int argc, char **argv)
{
    FILE *input;
    char col1[20],col2[20],col3[20],col4[20];
    int c1=0,c2=0,c3=0,c4=0,c5=0;
    int c12=0,c13=0,c14=0,c15=0;
    int c23=0,c24=0,c25=0;
    int c34=0,c35=0;
    int c45=0;
    int c123=0,c125=0,c124=0,c234=0,c235=0,c135=0,c245=0;
    int cFinal[50];
    int minSupport=2;
    int confidence=60;
    int candidate=0,list=0;

    int i=0;
    printf("The Minimum support is %d\n",minSupport);
    printf("The Confidence is %d%\n",confidence);

    input=fopen("input.txt","r"); //taking input from file //reading
entire file
    fscanf(input,"%s%s%s%s",col1,col2,col3,col4);
    while(!feof(input)){
        cFinal[i]=atoi(col1);printf("%d\t",cFinal[i]);i++; //string to int
        cFinal[i]=atoi(col2);printf("%d\t",cFinal[i]);i++;
        cFinal[i]=atoi(col3);printf("%d\t",cFinal[i]);i++;
        cFinal[i]=atoi(col4);printf("%d\t\n",cFinal[i]);i++;
        fscanf(input,"%s%s%s%s",col1,col2,col3,col4);
    }
    for(int j=0;j<i;j++){ //calculating number of occurence of each item
        if(cFinal[j]==1)
            c1++;
        if(cFinal[j]==2)
            c2++;
        if(cFinal[j]==3)
            c3++;
        if(cFinal[j]==4)
            c4++;
        if(cFinal[j]==5)
            c5++;
    }

    printf("\n\nCandidate Set %d \n",candidate); //printing candidate and
itemset
    printf("1\t%d\n",c1);
    printf("2\t%d\n",c2);
    printf("3\t%d\n",c3);
    printf("4\t%d\n",c4);
```

```

printf("5\t%d\n",c5);

printf("\nList = %d \n",list);
if(c1 >= minSupport) printf("1\t%d\n",c1);
if(c1 >= minSupport) printf("2\t%d\n",c2);
if(c1 >= minSupport) printf("3\t%d\n",c3);
if(c1 >= minSupport) printf("4\t%d\n",c4);
if(c1 >= minSupport) printf("5\t%d\n",c5);

candidate++;
list++;

input=fopen("input.txt","r"); //taking input from file
fscanf(input,"%s%s%s%s",col1,col2,col3,col4); //now comparing 1-2,2-
3,3-4,.....
while(!feof(input)){
    if((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) || (strcmp(col1,"1")==0 &&
strcmp(col3,"2")==0) || (strcmp(col1,"1")==0 && strcmp(col4,"2")==0) || (strcmp(col2,"1")==0
&& strcmp(col3,"2")==0) || (strcmp(col2,"1")==0 && strcmp(col4,"2")==0) ||
(strcmp(col3,"1")==0 && strcmp(col4,"2")==0))
        c12++;
    if((strcmp(col1,"1")==0 && strcmp(col2,"3")==0) || (strcmp(col1,"1")==0 &&
strcmp(col3,"3")==0) || (strcmp(col1,"1")==0 && strcmp(col4,"3")==0) || (strcmp(col2,"1")==0
&& strcmp(col3,"3")==0) || (strcmp(col2,"1")==0 && strcmp(col4,"3")==0) ||
(strcmp(col3,"1")==0 && strcmp(col4,"3")==0))
        c13++;
    if((strcmp(col1,"1")==0 && strcmp(col2,"4")==0) || (strcmp(col1,"1")==0 &&
strcmp(col3,"4")==0) || (strcmp(col1,"1")==0 && strcmp(col4,"4")==0) || (strcmp(col2,"1")==0
&& strcmp(col3,"4")==0) || (strcmp(col2,"1")==0 && strcmp(col4,"4")==0) ||
(strcmp(col3,"1")==0 && strcmp(col4,"4")==0))
        c14++;
    if((strcmp(col1,"1")==0 && strcmp(col2,"5")==0) || (strcmp(col1,"1")==0 &&
strcmp(col3,"5")==0) || (strcmp(col1,"1")==0 && strcmp(col4,"5")==0) || (strcmp(col2,"1")==0
&& strcmp(col3,"5")==0) || (strcmp(col2,"1")==0 && strcmp(col4,"5")==0) ||
(strcmp(col3,"1")==0 && strcmp(col4,"5")==0))
        c15++;
    if((strcmp(col1,"2")==0 && strcmp(col2,"3")==0) || (strcmp(col1,"2")==0 &&
strcmp(col3,"3")==0) || (strcmp(col1,"2")==0 && strcmp(col4,"3")==0) || (strcmp(col2,"2")==0
&& strcmp(col3,"3")==0) || (strcmp(col2,"2")==0 && strcmp(col4,"3")==0) ||
(strcmp(col3,"2")==0 && strcmp(col4,"3")==0))
        c23++;
    if((strcmp(col1,"2")==0 && strcmp(col2,"4")==0) || (strcmp(col1,"2")==0 &&
strcmp(col3,"4")==0) || (strcmp(col1,"2")==0 && strcmp(col4,"4")==0) || (strcmp(col2,"2")==0
&& strcmp(col3,"4")==0) || (strcmp(col2,"2")==0 && strcmp(col4,"4")==0) ||
(strcmp(col3,"2")==0 && strcmp(col4,"4")==0))
        c24++;
    if((strcmp(col1,"2")==0 && strcmp(col2,"5")==0) || (strcmp(col1,"2")==0 &&
strcmp(col3,"5")==0) || (strcmp(col1,"2")==0 && strcmp(col4,"5")==0) || (strcmp(col2,"2")==0
&& strcmp(col3,"5")==0) || (strcmp(col2,"2")==0 && strcmp(col4,"5")==0) ||
(strcmp(col3,"2")==0 && strcmp(col4,"5")==0))
        c25++;
}

```

```

        if((strcmp(col1,"3")==0 && strcmp(col2,"4")==0) || (strcmp(col1,"3")==0 &&
strcmp(col3,"4")==0) || (strcmp(col1,"3")==0 && strcmp(col4,"4")==0) || (strcmp(col2,"3")==0
&& strcmp(col3,"4")==0) || (strcmp(col2,"3")==0 && strcmp(col4,"4")==0) ||
(strcmp(col3,"3")==0 && strcmp(col4,"4")==0))
            c34++;
        if((strcmp(col1,"3")==0 && strcmp(col2,"5")==0) || (strcmp(col1,"3")==0 &&
strcmp(col3,"5")==0) || (strcmp(col1,"3")==0 && strcmp(col4,"5")==0) || (strcmp(col2,"3")==0
&& strcmp(col3,"5")==0) || (strcmp(col2,"3")==0 && strcmp(col4,"5")==0) ||
(strcmp(col3,"3")==0 && strcmp(col4,"5")==0))
            c35++;
        if((strcmp(col1,"4")==0 && strcmp(col2,"5")==0) || (strcmp(col1,"4")==0 &&
strcmp(col3,"5")==0) || (strcmp(col1,"4")==0 && strcmp(col4,"5")==0) || (strcmp(col2,"4")==0
&& strcmp(col3,"5")==0) || (strcmp(col2,"4")==0 && strcmp(col4,"5")==0) ||
(strcmp(col3,"4")==0 && strcmp(col4,"5")==0))
            c45++;
        fscanf(input,"%s%s%s%s",col1,col2,col3,col4);
    }

```

```

printf("\n\nCandidate Set %d \n",candidate);
printf("12=%d\n",c12);
printf("13=%d\n",c13);
printf("14=%d\n",c14);
printf("15=%d\n",c15);
printf("23=%d\n",c23);
printf("24=%d\n",c24);
printf("25=%d\n",c25);
printf("34=%d\n",c34);
printf("35=%d\n",c35);
printf("45=%d\n",c45);

```

```

printf("\nList = %d \n",list);
if(c12 >= minSupport) printf("12\t%d\n",c12);
if(c13 >= minSupport) printf("13\t%d\n",c13);
if(c14 >= minSupport) printf("14\t%d\n",c14);
if(c15 >= minSupport) printf("15\t%d\n",c15);
if(c23 >= minSupport) printf("23\t%d\n",c23);
if(c24 >= minSupport) printf("24\t%d\n",c24);
if(c25 >= minSupport) printf("25\t%d\n",c25);
if(c34 >= minSupport) printf("34\t%d\n",c34);
if(c35 >= minSupport) printf("35\t%d\n",c35);
if(c45 >= minSupport) printf("36\t%d\n",c45);
candidate++;
list++;

```

```

input=fopen("input.txt","r"); //taking input from file
fscanf(input,"%s%s%s%s",col1,col2,col3,col4);
while(!feof(input)){
    if(((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) && (strcmp(col3,"3")==0) )||
((strcmp(col2,"1")==0 && strcmp(col3,"2")==0) && (strcmp(col4,"3")==0) ))
        c123++;
    if(((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) && (strcmp(col3,"4")==0) )||
((strcmp(col2,"1")==0 && strcmp(col3,"2")==0) && (strcmp(col4,"4")==0) ))
        c124++;
}

```

```

        if(((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) && (strcmp(col3,"5")==0) )||
((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) && (strcmp(col4,"5")==0) ))
            c125++;
        if(((strcmp(col1,"1")==0 && strcmp(col2,"3")==0) && (strcmp(col3,"5")==0) )||
((strcmp(col2,"1")==0 && strcmp(col3,"3")==0) && (strcmp(col4,"5")==0) ))
            c135++;
        if(((strcmp(col1,"2")==0 && strcmp(col2,"3")==0) && (strcmp(col3,"4")==0) )||
((strcmp(col2,"2")==0 && strcmp(col3,"3")==0) && (strcmp(col4,"4")==0) ))
            c234++;
        if(((strcmp(col1,"2")==0 && strcmp(col2,"3")==0) && (strcmp(col3,"5")==0) )||
((strcmp(col2,"2")==0 && strcmp(col3,"3")==0) && (strcmp(col4,"5")==0) ))
            c235++;
        if(((strcmp(col1,"2")==0 && strcmp(col2,"4")==0) && (strcmp(col3,"5")==0) )||
((strcmp(col2,"2")==0 && strcmp(col3,"4")==0) && (strcmp(col4,"5")==0) ))
            c245++;
        fscanf(input,"%s%s%s%s",col1,col2,col3,col4);
    }

    printf("\n\nCandidate Set %d \n",candidate);
    printf("C123=%d\n",c123);
    printf("C124=%d\n",c124);
    printf("C125=%d\n",c125);
    printf("C135=%d\n",c135);
    printf("C234=%d\n",c234);
    printf("C235=%d\n",c235);
    printf("C245=%d\n",c245);

    printf("\nList = %d \n",list);
    if(c123 >= minSupport) printf("123\t%d\n",c123);
    if(c124 >= minSupport) printf("124\t%d\n",c124);
    if(c125 >= minSupport) printf("125\t%d\n",c125);
    if(c135 >= minSupport) printf("135\t%d\n",c135);
    if(c234 >= minSupport) printf("234\t%d\n",c234);
    if(c235 >= minSupport) printf("235\t%d\n",c235);
    if(c245 >= minSupport) printf("245\t%d\n",c245);
    candidate++;
    list++;
    printf("\nTherefore rules are \n\n\n");
    printf("1-2--->1-2-3 = %.0f%%\n", (c123/(float)c12)*100);
    printf("2-3--->1-2-3 = %.0f%%\n", (c123/(float)c23)*100);
    printf("1-3--->1-2-3 = %.0f%%\n", (c123/(float)c13)*100);

    printf("1-2--->1-2-5 = %.0f%%\n", (c125/(float)c12)*100);
    printf("2-5--->1-2-5 = %.0f%%\n", (c125/(float)c25)*100);
    printf("1-5--->1-2-5 = %.0f%%\n", (c125/(float)c15)*100);

    printf("2--->2-5 = %.0f%%\n", (c25/(float)c2)*100);
    printf("5--->2-5 = %.0f%%\n", (c25/(float)c5)*100);

    printf("1--->1-5 = %.0f%%\n", (c15/(float)c1)*100);
    printf("5--->1-5 = %.0f%%\n", (c15/(float)c5)*100);

```

```
printf("5 from 2-5 and 5 from 1-5 are association");
```

```
return 0;
```

```
}
```

### Input.txt

```
1 2 5 -
2 4 - -
2 3 - -
1 2 4 -
1 3 - -
2 3 - -
1 3 - -
1 2 3 5
1 2 3 -
```

### OUTPUT

```
divyang@Divyang:~/DWM$ gcc apriori.c -lm
```

```
apriori.c: In function 'main':
```

```
apriori.c:20:31: warning: unknown conversion type character '\x0a' in format [-Wformat=]
printf("The Confidence is %d%\n",confidence);
      ^~
```

```
apriori.c:25:13: warning: implicit declaration of function 'atoi' [-Wimplicit-function-declaration]
cFinal[i]=atoi(col1);printf("%d\t",cFinal[i]);i++; //string to int
      ^~
```

```
apriori.c:66:7: warning: implicit declaration of function 'strcmp' [-Wimplicit-function-declaration]
if((strcmp(col1,"1")==0 && strcmp(col2,"2")==0) || (strcmp(col1,"1")==0 &&
      ^~~~~
```

```
strcmp(col3,"2")==0) || (strcmp(col1,"1")==0 && strcmp(col4,"2")==0) || (strcmp(col2,"1")==0
&& strcmp(col3,"2")==0) || (strcmp(col2,"1")==0 && strcmp(col4,"2")==0) ||
(strcmp(col3,"1")==0 && strcmp(col4,"2")==0))
      ^~~~~~
```

```
divyang@Divyang:~/DWM$ ./a.out
The Minimum support is 2
The Confidence is 60%
```

1	2	5	0
2	4	0	0
2	3	0	0
1	2	4	0
1	3	0	0
2	3	0	0
1	3	0	0
1	2	3	5
1	2	3	0

Candidate Set 0



1	6
2	7
3	6
4	2
5	2

List = 0

1	6
2	7
3	6
4	2
5	2

Candidate Set 1

12=4  
13=4  
14=1  
15=2  
23=4  
24=2  
25=2  
34=0  
35=1  
45=0

List = 1

12	4
13	4
15	2
23	4
24	2
25	2

Candidate Set 2

C123=2  
C124=1  
C125=2  
C135=0  
C234=0  
C235=1  
C245=0

List = 2

123	2
125	2

Therefore rules are

1-2--->1-2-3 =50%  
2-3--->1-2-3 =50%  
1-3--->1-2-3 =50%

1-2--->1-2-5 =50%

2-5--->1-2-5 =100%

1-5--->1-2-5 =100%

2--->2-5 = 29%

5--->2-5 = 100%

1--->1-5 = 33%

5--->1-5 = 100%

5 from 2-5 and 5 from 1-5 are association

## EXPERIMENT NO 5

### KMEANS

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

void main()
{
    int i,n,ogArray[20],k1[20],k2[20];

    printf("Enter the number of elements: ");
    scanf("%d",&n);
    printf("Enter the elements in ascending order: ");
    for(i=0;i<n;i++)
        scanf("%d",&ogArray[i]);

    int m1,m1old,m2,m2old,j=0,k=0,sum1=0,sum2=0;
    m1=m1old=ogArray[1], m2=m2old=ogArray[n-2];

    //gen1
    for(i=0;i<n;i++)
        if(abs(ogArray[i]-m1)<abs(ogArray[i]-m2))
        {
            k1[j] = ogArray[i];
            j++;
        }
        else
        {
            k2[k] = ogArray[i];
            k++;
        }

    for(i=0;i<j;i++)
        sum1 += k1[i];
    m1 = sum1/j;

    for(i=0;i<k;i++)
        sum2 += k2[i];
    m2 = sum2/k;

    //gen2
    while(m1 != m1old && m2 != m2old)
    {
        for(i=0;i<n;i++)
            if(abs(ogArray[i]-m1) < abs(ogArray[i]-m2))
            {
                k1[j] = ogArray[i];
                j++;
            }
            else
            {
                k2[k] = ogArray[i];
                k++;
            }
    }
}
```

```

    }

    for(i=0;i<j;i++)
        sum1 += k1[i];
    m1old = m1;
    m1 = sum1/j;

    for(i=0;i<k;i++)
        sum2 += k2[i];
    m2old = m2;
    m2 = sum2/k;
}

printf("\nMean 1: %d Partition 1:\n",m1);
for(i=0;i<j;i++)
    printf("%d\t",k1[i]);
printf("\nMean 2:%d Partition 2:\n",m2);
for(i=0;i<k;i++)
    printf("%d\t",k2[i]);

}

```

divyang@Divyang:~/DWM\$ gcc kmeans.c -lm

divyang@Divyang:~/DWM\$ ./a.out

Enter the number of elements: 8

Enter the elements in ascending order: 1

6  
9  
10  
13  
18  
20  
24

Mean 1: 6 Partition 1:

1      6      9      10

Mean 2:18 Partition 2:

13      18      20      24

## EXPERIMENT NO 6

### LINEAR REGRESSION

```
#include<stdio.h>

void main()
{
    int i,j,n,x[10],y[10],xBar,yBar,abBar,asqBar,A[10],B[10],AB[10],ASQUARE[10];
    int sumX=0,sumY=0,sumAB=0,sumASQUARE=0;
    float beta0, beta1, inputX, outputY;

    printf("Enter the number of values of x and y:");
    scanf("%d",&n);

    printf("\nEnter the values of x: ");
    for(i=0;i<n;i++)
        scanf("%d",&x[i]);
    printf("\nEnter the values of y: ");
    for(i=0;i<n;i++)
        scanf("%d",&y[i]);

    for(i=0;i<n;i++)
    {
        sumX += x[i];
        sumY += y[i];
    }

    xBar = sumX/n;
    yBar = sumY/n;

    for(i=0;i<n;i++)
    {
        A[i] = x[i] - xBar;
        B[i] = y[i] - yBar;
    }

    for(i=0;i<n;i++)
        AB[i] = A[i]*B[i];

    for(i=0;i<n;i++)
        ASQUARE[i] = A[i]*A[i];

    for(i=0;i<n;i++)
    {
        sumAB += AB[i];
        sumASQUARE += ASQUARE[i];
    }

    abBar = sumAB/n;
    asqBar = sumASQUARE/n;

    beta1 = (float)abBar/(float)asqBar;
    beta0 = yBar - (beta1*xBar);
```

```
printf("values of beta0: %f and beta1: %f\n",beta0,beta1);
printf("Enter value of x for prediction:\n");
scanf("%f",&inputX);

outputY = beta0 + (beta1*inputX);
printf("Value of y = %f at x = %f\n", outputY,inputX);
}
```

```
divyang@Divyang:~/DWM$ gcc linear.c
divyang@Divyang:~/DWM$ ./a.out
Enter the number of values of x and y:3
Enter the values of x: 4
10
15
Enter the values of y: 16
21
26
values of beta0: 12.900001 and beta1: 0.900000
Enter value of x for prediction:
2
Value of y = 14.700001 at x = 2.000000
```

## EXPERIMENT NO 8

### PAGE RANK

```

#include <stdio.h>
//void formula()
int main(int argc, char **argv)
{
    float n= 4;
    printf("Let us assume there are %.0f nodes\n",n);

    float A,B,C,D;
    A=1/n;B=1/n;C=1/n;D=1/n;                                //all initial values are
1/n
    printf("Intial Stage of A =%f\n",A); //just printing
    printf("Intial Stage of B =%f\n",B);
    printf("Intial Stage of C =%f\n",C);
    printf("Intial Stage of D =%f\n",D);
    printf("A ----->B\n");                                //diagram
    printf(" ^ ^|\n");
    printf(" | ||\n");
    printf(" | ||\n");
    printf(" | ||\n");
    printf(" | ||\n");
    printf(" | ||\n");
    printf(" \\\| |\n");
    printf(" C -----|\n");
    printf(" ^ |\n");
    printf(" | |\n");
    printf(" | |\n");
    printf(" | |\n");
    printf(" | |\n");
    printf(" \\\| |\n");
    printf(" D<-----\n");

    int ia=1,ib=2,ic=2,id=2;                                //incoming of node(i)
    int oa=2,ob=1,oc=3,od=1;                                //outgoing of node(o)
    float pra,prb,prc,prd;                                    //page rank of node(pr)
    printf("First Iteration\n\n");
    pra=C/oc;
    //formula=number of incoming nodes/outgoing nodes
    prb=C/oc+A/oa;                                            //if more than 1
incoming add that with respected outgoing nodes
    prc=A/oa+D/od;
    prd=C/oc+B/ob;
    printf("A = %f\n",pra);                                    //just printing the answers

    printf("B = %f\n",prb);
    printf("C = %f\n",prc);
    printf("D = %f\n",prd);

    printf("SEcond Iteration\n\n");
    A=pra;pra=0;                                              //now old values=new
values
    B=prb;prb=0;                                              //initialize pr
values of node=0 becoz didnt wanted to use more variables
    C=prc;prc=0;

```

```

        D=prd;prd=0;
        pra=C/oc;
from above
        prb=C/oc+A/oa;
        prc=A/oa+D/od;
        prd=C/oc+B/ob;
        float prttotal= pra+prb+prc+prd;
be 1
        printf("A = %f\n",pra);
        printf("B = %f\n",prb);
        printf("C = %f\n",prc);
        printf("D = %f\n",prd);
        printf("PRTOTAL = %f\n",prttotal);

        if(pra>prb){
greatest number
                if(pra>prc){
                        if(pra>prd){
                                printf("A is the most searched Website");}}}
        if(prb>pra){
                if(prb>prc){
                        if(prb>prd){
                                printf("B is the most searched Website");}}}

        if(prc>pra){
                if(prc>prb){
                        if(prc>prd){
                                printf("C is the most searched Website");}}}

        if(prd>pra){
                if(prd>prb){
                        if(prd>prc){
                                printf("D is the most searched Website");}}}

        return 0;
}

```

divyang@Divyang:~/DWM\$ gcc pr.c

divyang@Divyang:~/DWM\$ ./a.out

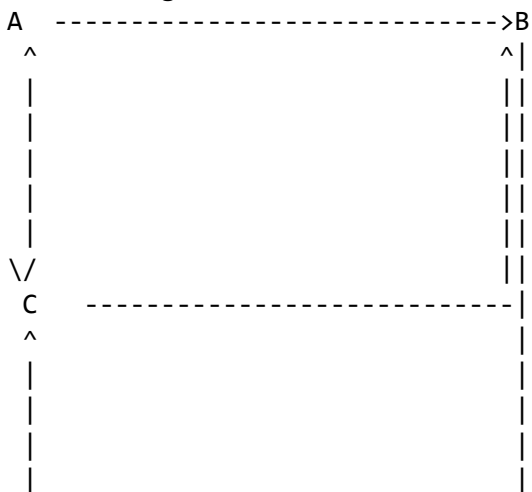
Let us assume there are 4 nodes

Intial Stage of A =0.250000

Intial Stage of B =0.250000

Intial Stage of C =0.250000

Intial Stage of D =0.250000





```

|
\
D<-----
First Iteration

```

```

A = 0.083333
B = 0.208333
C = 0.375000
D = 0.333333
SEcond Iteration

```

```

A = 0.125000
B = 0.166667
C = 0.375000
D = 0.333333
PRTOTAL = 1.000000
C is the most searched Website

```

```

#include<stdio.h>
#define damp 1

void main()
{
    int n,i,j,k,outlink[10],iter;
    float pr[10],prNew[10];
    int adj[10][10];

    printf("Enter number of nodes:");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("Enter the number of outlinks for node %d: ",i);
        scanf("%d",&outlink[i]);
    }

    //Adjacency Matrix
    printf("\n----CREATING ADJACENCEY MATRIX----\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("Link between %d and %d ?: ",i,j);
            scanf("%d",&adj[i][j]);
        }

    printf("\nAdjacency Matrix\n\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            printf("%d\t",adj[i][j]);
        printf("\n");
    }
}

```

```

for(i=0;i<n;i++)
{
    pr[i] = 1/(float)n;
    prNew[i] = 0;
}

printf("\nEnter number of iterations required: ");
scanf("%d",&iter);

for(k=0;k<iter;k++)//iter
{
    for(i=0;i<n;i++)//all nodes
        for(j=0;j<n;j++)
            if(adj[j][i] == 1)
                prNew[i] += ((1-damp) + (damp * (pr[j]/(float)outlink[j])));
}

for(i=0;i<n;i++)
    printf("\nPage Rank of Node %d is %f",i,prNew[i]);
}

```

```

divyang@Divyang:~/DWM$ gcc pr1.c
divyang@Divyang:~/DWM$ ./a.out
Enter number of nodes:4
Enter the number of outlinks for node 0: 2
Enter the number of outlinks for node 1: 3
Enter the number of outlinks for node 2: 1
Enter the number of outlinks for node 3: 0

```

-----CREATING ADJACENCY MATRIX-----

```

Link between 0 and 0 ? : 0
Link between 0 and 1 ? : 1
Link between 0 and 2 ? : 0
Link between 0 and 3 ? : 1
Link between 1 and 0 ? : 1
Link between 1 and 1 ? : 0
Link between 1 and 2 ? : 1
Link between 1 and 3 ? : 1
Link between 2 and 0 ? : 0
Link between 2 and 1 ? : 0
Link between 2 and 2 ? : 1
Link between 2 and 3 ? : 0
Link between 3 and 0 ? : 0
Link between 3 and 1 ? : 0
Link between 3 and 2 ? : 0
Link between 3 and 3 ? : 0

```

Adjacency Matrix

0	1	0	1
1	0	1	1
0	0	1	0
0	0	0	0

Enter number of iterations required: 2

Page Rank of Node 0 is 0.166667

Page Rank of Node 1 is 0.250000

Page Rank of Node 2 is 0.666667

Page Rank of Node 3 is 0.416667

## EXPERIMENT NO 9

### HITS

```
#include<stdio.h>

void main()
{
    int i,j,n,adj[10][10],adjTrans[10][10];
    int hubWeightVec[3] = {1,1,1}, AuthWeightVec[3] = {0,0,0};

    printf("Enter number of nodes: ");
    scanf("%d",&n);

    printf("\n----Creating Adjacency Matrix----\n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
        {
            printf("Link between %d and %d ?:",i,j);
            scanf("%d",&adj[i][j]);
        }

    printf("\nGiven Matrix:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            printf("%d\t",adj[i][j]);
        printf("\n");
    }

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            adjTrans[j][i] = adj[i][j];

    printf("\nTranspose Matrix:\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
            printf("%d\t",adjTrans[i][j]);
        printf("\n");
    }

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            AuthWeightVec[i] += adjTrans[i][j] * hubWeightVec[j];

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            hubWeightVec[i] += adj[i][j] * AuthWeightVec[j];

    printf("Authority Weight Vector:\n");
    for(i=0;i<n;i++)
        printf("%d\t", AuthWeightVec[i]);
    printf("\nHub Weight Vector:\n");
```

```

for(i=0;i<n;i++)
    printf("%d\t", hubWeightVec[i]);

printf("\n");
for(i=0;i<n;i++)
    if(hubWeightVec[i]>AuthWeightVec[i])
        printf("Node %d is a Hub\n",i);
    else
        printf("Node %d is an Authority\n",i);
}

```

```

divyang@Divyang:~/DWM$ gcc hits.c
divyang@Divyang:~/DWM$ ./a.out
Enter number of nodes: 4

```

----Creating Adjacency Matrix----

```

Link between 0 and 0 ? :0
Link between 0 and 1 ? :1
Link between 0 and 2 ? :1
Link between 0 and 3 ? :0
Link between 1 and 0 ? :1
Link between 1 and 1 ? :1
Link between 1 and 2 ? :1
Link between 1 and 3 ? :0
Link between 2 and 0 ? :1
Link between 2 and 1 ? :0
Link between 2 and 2 ? :0
Link between 2 and 3 ? :1
Link between 3 and 0 ? :0
Link between 3 and 1 ? :0
Link between 3 and 2 ? :1
Link between 3 and 3 ? :1

```

Given Matrix:

0	1	1	0
1	1	1	0
1	0	0	1
0	0	1	1

Transpose Matrix:

0	1	1	0
1	1	0	0
1	1	0	1
0	0	1	1

Authority Weight Vector:

9	2	4	3
---	---	---	---

Hub Weight Vector:

13	9	6	9
----	---	---	---

Node 0 is a Hub

Node 1 is a Hub

Node 2 is a Hub

Node 3 is a Hub