* Data Structure Lab (DSL):- Practical Number - 1 (Group - A)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a code in python to study set operations using lists.

Aim :-
In SE computer engineering class, group A students play cricket, group B studen play badminton and group C student play football.
Write a python program using function to compute :-
1) List of student who play both cricket and badminton.
2) List of student who play either cricket or badminton but not both.
3) Number of student who play neither cricket nor badminton.
4) Number of student who play cricket and football but not badminton.

Objective :-
To learn set operations using list in python.

Theory :-
Set -
A set is a collection of well defined and distinct objects.

Union of Set:-

Union of two sets A and B is the set consisting all elements which are in A or in B or in both A and B.

$$A \cup B = \{x | x \in A \text{ or } x \in B\}$$

Examples:-

$A = \{a, b, c, d, e, f, i\}$

$B = \{a, b, c, d, e, f, g, h\}$

$A \cup B = \{a, b, c, d, e, f, g, h, i\}$

Intersection of Set:-

Intersection of two set A and B is the set consisting of elements which are in A as well as B.

$$A \cap B = \{x | x \in A \text{ and } x \in B\}$$

Examples:-

$A = \{1, 2, 3, 4, 5, 6, 7\}$

$B = \{0, 2, 4, 6, 8, 10\}$

$A \cap B = \{2, 4, 6\}$

Difference of Set:-

The set difference of sets A and B, is the set of all elements in A that are not in B.

$$A - B = \{x | x \in A \text{ and } x \notin B\}$$

Example:-

$A = \{1, 2, 3, 4, 5\}$

$B = \{1, 3, 5\}$

$A - B = \{2, 4\}$

$B - A = \{\phi\}$

Symmetric Difference:-

Symmetric difference of two sets which are either of the set but not in their intersection.

$$A \oplus B = \{x | x \in A-B \text{ or } x \in B-A\}$$

Example:-

$A = \{a, b, e, g\}$

$B = \{d, e, f, g\}$

$A \oplus B = \{a, b, d, f\}$

Algorithm:-

Step 1- Start

Step 2- Accept total number of student in the class.

Step 3- Enter the number of student playing cricket, badminton and football.

Step 4- Accept the name of student playing cricket, badminton and football. Store them in three different lists.

Step 5- Write function to remove duplicate elements from list.

Step 6- Remove duplicate elements from the three list created in step 4.

Step 7- Print the name of students present in each list.

Step 8- Calculate the number of students playing both cricket and badminton (Intersection of both).

Step 9 - Show the number and name of students playing both cricket and bad-

Step 10 - Calculate the number of students who play either cricket or badminton but not both.

Step 11 - Display the names and number of these students.

Step 12 - Calculate number of students who play neither cricket nor badm-inton.

Step 13 - Display total number and names of these students.

Step 14 - Calculate number of student who play cricket and football but not badminton.

Step 15 - Display total number and names of these students.

Step 16 - Stop.

Analysis -

The time complexity of all the function in the program is O(n).

Conclusion -

Hence, we have studied set operations in python using lists.

GROUP A-01:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

Program:---

```
# Group A - 1 Write a Python program to store marks scored in subject
"Fundamental of Data Structure"
# by N students in the class. Write functions to compute following:
# a) The average score of class
# b) Highest score and lowest score of class
# c) Count of students who were absent for the test
# d) Display mark with highest frequency

# Average Score Of Test :---


def average(l):
    sum = 0
    cnt = 0
    for i in range(len(l)):
        if l[i] != -999:
            sum += l[i]
            cnt += 1

    avg = sum / cnt
    print("Total Marks are : ", sum)
    print("Average Marks are : {:.2f}".format(avg))


# Highest Score In The Test :---


def Maximum(l):
    Max = l[0]
    for i in range(len(l)):
        if l[i] > Max:
            Max = l[i]
    return (Max)


# Lowest Score In The Test :---


def Minimum(l):
    for i in range(len(l)):
        if l[i] != -37:
            Min = l[i]
            break

    for j in range(i + 1, len(l)):
        if l[j] != -37 and l[j] < Min:
            Min = l[j]
    return (Min)


# Absent Number of Student For Test :---
```

```
def absentCnt(l):
    cnt = 0
    for i in range(len(l)):
        if l[i] == -37:
            cnt += 1
    return (cnt)


# Highest Frequency Of Marks :---


def maxFrequency(l):
    i = 0
    Max = 0
    print(" Marks ----> frequency count ")
    for ele in l:
        if l.index(ele) == i:
            print(ele, "---->", l.count(ele))
            if l.count(ele) > Max:
                Max = l.count(ele)
                mark = ele
        i += 1
    return (mark, Max)


# Input the Number of Students and Their Corresponding Marks :---


marksInFDS = []
noStudents = int(input("Enter total number of students : "))
for i in range(noStudents):
    marks = int(input("Enter marks of Student " + str(i + 1) + " : "))
    marksInFDS.append(marks)

flag = 1
while flag == 1:
    print("/*************MENU*************/")
    print("1. The average score of class ")
    print("2. Highest score and lowest score of class ")
    print("3. Count of students who were absent for the test ")
    print("4. Display mark with highest frequency ")
    print("5. Exit ")
    choice = int(input("Enter your choice : "))

    if choice == 1:
        average(marksInFDS)

    elif choice == 2:
        print("Highest score in the class is : ", Maximum(marksInFDS))
        print("Lowest score in the class is : ", Minimum(marksInFDS))

    elif choice == 3:
        print("Count of students who were absent for the test is : ",
absentCnt(marksInFDS))

    elif choice == 4:
```

```
        mark, count = maxFrequency(marksInFDS)
        print("Highest frequency of marks {0} is {1} ".format(mark, count))

    else:
        print("Wrong Choice,Please Choose Another Option.")
        flag = 0
```

Output:---

Enter total number of students : 5

Enter marks of Student 1 : 9

Enter marks of Student 2 : 8

Enter marks of Student 3 : 7

Enter marks of Student 4 : 6

Enter marks of Student 5 : 9

/**********MENU**********/

1. The average score of class

2. Highest score and lowest score of class

3. Count of students who were absent for the test

4. Display mark with highest frequency

5. Exit

Enter your choice : 1

Total Marks are : 39

Average Marks are : 7.80

/**********MENU**********/

1. The average score of class

2. Highest score and lowest score of class

3. Count of students who were absent for the test

4. Display mark with highest frequency

5. Exit

Enter your choice : 2

Highest score in the class is : 9

Lowest score in the class is : 6

/**********MENU**********/

1. The average score of class

2. Highest score and lowest score of class

3. Count of students who were absent for the test

4. Display mark with highest frequency

5. Exit

Enter your choice : 3

Count of students who were absent for the test is : 0

/**********MENU**********/

1. The average score of class

2. Highest score and lowest score of class

3. Count of students who were absent for the test

4. Display mark with highest frequency

5. Exit

Enter your choice : 4

Marks ----> frequency count

9 ----> 2

8 ----> 1

7 ----> 1

6 ----> 1

Highest frequency of marks 9 is 2

/**********MENU**********/

1. The average score of class

2. Highest score and lowest score of class

3. Count of students who were absent for the test

4. Display mark with highest frequency

5. Exit

Enter your choice : 5

Wrong Choice,Please Choose Another Option.

Process finished with exit code 0

* Data Structure Lab (DSL):- Practical Number - 2 (Group - A)

Name:- Kaustubh Shrikant Kabra.
Class:- Second Year Engineering
Div:- A Roll Number:-
Batch:-
Department:- Computer Department.
College:- AISSMS's IOIT.

Title:-
Write a python program to store marks scored in a subject by students in the class.

Aim:-
Write a python program to store marks scored in subject 'Fundamental of Data Structure' by N students in the class. Write function to compute following:-
1) The average score of class
2) Highest and lowest score of class.
3) Count of students who were absent for the test.
4) Display marks with higest frequency.

Objective:-
1) To study the concept of list in python.
2) To understand various operations on lists.

Theory:-
List-
List is a versatile datatype available in python. It is a sequence

in which elements are written as a list of comma separated between square brackets, the key feature of a list is that it can have elements that belong to different datatypes.

Syntax → <list_variable> = [val1, val2, ...]

Examples:-

1) A = [1, 2, 3, 4, 5]
2) B = [1, 3, 5, 7, 'A', 'B', 'c', "Hello"]

Accessing Lists -

Similar to strings, lists can also be sliced and concatenated. To access values in lists, square brackets are used to slice along with index.

Example:-

```
A = [1, 2, 3, 4, 5, 6, 7, 8]
print (A[0])          // 1
print (A[3])          // 4
print (A[1:4])        // 2 3 4
```

Updating Lists -

Once created, values in a list can be easily updated using the index and the assignment operator.

Example:-

```
A = [1, 2, 3, 4]
A[2] = 100
print A               // 1 2 100 4
```

Nested Lists -

Nested list means a list within another list. List can have element of different datatypes which can include even a list.

Example:-

list1 = [1, 'a', "abc", [2,3,5,7], 8.9]

Algorithm:-

Step1 - Start

Step2 - Read number of students from the user (N).

Step3 - Accept marks in FDS for the total number of students.

Step4 - Append the marks into a list.

Step5 - Write a function to calculate average marks of students.

Step6 - Display the average marks of the class.

Step7 - Write function to calculate highest and lowest marks in class.

Step8 - Display the highest and lowest marks.

Step9 - Write a function to calculate number of absent student.

Step10 - Display number of absent students.

Step 11 - Write a function to calculate the frequency of the entered marks.

Step 12 - Display the marks with their corresponding frequencies.

Step 13 - Display marks with highest frequency along with its frequency.

Step 14 - Stop.

Analysis:-

The time complexcity for average(), highestMarks(), lowestMarks(), absentstudents() and frequency is O(n).

Conclusion:-

Hence, we have used list to store marks and have performed various operations on it.

GROUP A-02:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

Program:---

```
# Group A - 2. In second year computer engineering class, group A student's
play cricket, group B students
# play badminton and group C students play football. Write a Python program
using functions to compute following: -
# a) List of students who play both cricket and badminton
# b) List of students who play either cricket or badminton but not both
# c) Number of students who play neither cricket nor badminton
# d) Number of student who play cricket and football but not badminton.
# (Note- While realizing the group, duplicate entries should be avoided, Do
not use SET built-in functions)

# List Of Student as per Games they Play

SeComp = ["Kaustubh Kabra", "Harsh Shah", "Onasvee Banarse", "Ankit Patil",
"Virat Kohli", "MS Dhoni",
          "Ravindra Jadeja", "Smriti Mandhana", "Lionel Messi", "Cristiano
Ronaldo", "Neymar Jr", "Saina Nehwal",
          "Shrikanth Kidambi", "PV Sindhu"]
Cricket = ["Kaustubh Kabra", "Ankit Patil", "Virat Kohil", "MS Dhoni",
"Ravindra Jadeja", "Smriti Mandhana"]
Badminton = ["Kaustubh Kabra", "Harsh Shah", "PV Sindhu", "Shrikanth
Kidambi", "Saina Nehwal", "Virat Kohli",
             "MS Dhoni"]
Football = ["Kaustubh Kabra", "Onasvee Banarse", "Lionel Messi", "Cristiano
Ronaldo", "Neymar Jr", "Virat Kohli",
            "MS Dhoni", "Ravindra Jadeja"]

# Removal of Duplicates[Here in our Program there is no Duplicates]


def removeduplicate(t):

    l1 = []
    for i in t:
        if i not in l1:
            l1.append(i)
    return l1

# Union Set[ Union of Cricket Playing Student and Badminton Playing Student]


def union(lst1, lst2):
    lst3 = lst1.copy()
    for value in lst2:
        if value not in lst3:
            lst3.append(value)
    return lst3

# Intersection Set [We have no use of this in solving this Practical]
```

```
def intersection(lst1, lst2):
    lst3 = []
    for value in lst1:
        if value in lst2:
            lst3.append(value)
    return lst3

# Difference Set[ Difference of Union Set of Cricket and Badminton Playing
Student from All Secomp Student ]


def difference(lst1, lst2):
    lst3 = []
    for value in lst1:
        if value not in lst2:
            lst3.append(value)
    return lst3

# Symmetric Difference Set[ Difference of Intersection of Cricket and
Badminton Playing Student from
# Union of Cricket and Badminton Playing Student ]


def symmetricdiff(lst1, lst2):
    d1 = difference(lst1, lst2)
    d2 = difference(lst2, lst1)
    lst3 = union(d1, d2)
    return lst3

# Required Solution[ Difference of Union of Cricket and Badminton Playing
Student from All SeComp Student ]


def ncnb(lst1, lst2, lst3):

    lst4 = difference(lst1, union(lst2, lst3))
    print(lst4)
    return lst4, len(lst4)


Cricket = removeduplicate(Cricket)
Badminton = removeduplicate(Badminton)
Football = removeduplicate(Football)

flag = 1
while flag == 1:
    print("/~~~~~~~MENU~~~~~~~/")
    print("1. List of students who played both Cricket and Badminton ")
    print("2. list of students who play either cricket or badminton but not
both ")
    print("3. Number of students who play neither cricket nor badminton ")
    print("4. Number of student who play cricket and football but not
badminton ")
    print("5. Exit")
    choice = int(input("Enter your choice : "))

    if choice == 1:
```

```
        print(" Cricket : ", Cricket)
        print(" Badminton : ", Badminton)
        print(" List of students who played both Cricket and Badminton : ")
        print(intersection(Cricket, Badminton))

    elif choice == 2:
        print(" Cricket : ", Cricket)
        print(" Badminton : ", Badminton)
        print(" list of students who play either cricket or badminton but not
both ")
        print(symmetricdiff(Cricket, Badminton))

    elif choice == 3:
        print(" Class : ", SeComp)
        print(" Cricket : ", Cricket)
        print(" Badminton : ", Badminton)
        l, cnt = ncnb(SeComp, Cricket, Badminton)
        print("Number of students who play neither cricket nor badminton : ",
cnt)
        print("List of students who play neither cricket nor badminton : ",
l)
    elif choice == 4:
        print("Class :", SeComp)
        print("Cricket :", Cricket)
        print("Badminton", Badminton)
        print("Football", Football)
        print("List of students who play cricket and football but not
badminton :")
        print(difference(SeComp, Badminton))
        print("Number of students who play cricket and football but not
badminton :")
        print(len(difference(SeComp, Badminton)))
    else:
        print("Wrong Choice,Please Choose Another Option ")
        flag = 0
```

Output:---

/~~~~~~~MENU~~~~~~~/

1. List of students who played both Cricket and Badminton

2. list of students who play either cricket or badminton but not both

3. Number of students who play neither cricket nor badminton

4. Number of student who play cricket and football but not badminton

5. Exit

Enter your choice : 1

Cricket : ['Kaustubh Kabra', 'Ankit Patil', 'Virat Kohil', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana']

Badminton : ['Kaustubh Kabra', 'Harsh Shah', 'PV Sindhu', 'Shrikanth Kidambi', 'Saina Nehwal', 'Virat Kohli', 'MS Dhoni']

List of students who played both Cricket and Badminton :

['Kaustubh Kabra', 'MS Dhoni']

/~~~~~~~MENU~~~~~~~/

1. List of students who played both Cricket and Badminton

2. list of students who play either cricket or badminton but not both

3. Number of students who play neither cricket nor badminton

4. Number of student who play cricket and football but not badminton

5. Exit

Enter your choice : 2

Cricket : ['Kaustubh Kabra', 'Ankit Patil', 'Virat Kohil', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana']

Badminton : ['Kaustubh Kabra', 'Harsh Shah', 'PV Sindhu', 'Shrikanth Kidambi', 'Saina Nehwal', 'Virat Kohli', 'MS Dhoni']

list of students who play either cricket or badminton but not both

['Ankit Patil', 'Virat Kohil', 'Ravindra Jadeja', 'Smriti Mandhana', 'Harsh Shah', 'PV Sindhu', 'Shrikanth Kidambi', 'Saina Nehwal', 'Virat Kohli']

/~~~~~~~MENU~~~~~~~/

1. List of students who played both Cricket and Badminton

2. list of students who play either cricket or badminton but not both

3. Number of students who play neither cricket nor badminton

4. Number of student who play cricket and football but not badminton

5. Exit

Enter your choice : 3

Class : ['Kaustubh Kabra', 'Harsh Shah', 'Onasvee Banarse', 'Ankit Patil', 'Virat Kohli', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr', 'Saina Nehwal', 'Shrikanth Kidambi', 'PV Sindhu']

Cricket : ['Kaustubh Kabra', 'Ankit Patil', 'Virat Kohil', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana']

Badminton : ['Kaustubh Kabra', 'Harsh Shah', 'PV Sindhu', 'Shrikanth Kidambi', 'Saina Nehwal', 'Virat Kohli', 'MS Dhoni']

['Onasvee Banarse', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr']

Number of students who play neither cricket nor badminton : 4

List of students who play neither cricket nor badminton : ['Onasvee Banarse', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr']

/~~~~~~~MENU~~~~~~~/

1. List of students who played both Cricket and Badminton

2. list of students who play either cricket or badminton but not both

3. Number of students who play neither cricket nor badminton

4. Number of student who play cricket and football but not badminton

5. Exit

Enter your choice : 4

Class : ['Kaustubh Kabra', 'Harsh Shah', 'Onasvee Banarse', 'Ankit Patil', 'Virat Kohli', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr', 'Saina Nehwal', 'Shrikanth Kidambi', 'PV Sindhu']

Cricket : ['Kaustubh Kabra', 'Ankit Patil', 'Virat Kohil', 'MS Dhoni', 'Ravindra Jadeja', 'Smriti Mandhana']

Badminton ['Kaustubh Kabra', 'Harsh Shah', 'PV Sindhu', 'Shrikanth Kidambi', 'Saina Nehwal', 'Virat Kohli', 'MS Dhoni']

Football ['Kaustubh Kabra', 'Onasvee Banarse', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr', 'Virat Kohli', 'MS Dhoni', 'Ravindra Jadeja']

List of students who play cricket and football but not badminton :

['Onasvee Banarse', 'Ankit Patil', 'Ravindra Jadeja', 'Smriti Mandhana', 'Lionel Messi', 'Cristiano Ronaldo', 'Neymar Jr']

Number of students who play cricket and football but not badminton :

7

/~~~~~~~MENU~~~~~~~~/

1. List of students who played both Cricket and Badminton

2. list of students who play either cricket or badminton but not both

3. Number of students who play neither cricket nor badminton

4. Number of student who play cricket and football but not badminton

5. Exit

Enter your choice : 5

Wrong Choice,Please Choose Another Option

Process finished with exit code 0

* Data Structure Lab (DSL) - Practical Number - 3 (Group - A)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a python program for magic square.

Aim :-
Write a python program for magic square. A magic square is a n*n matrix of the integers 1 to n^2 such that the sum of each row, column and diagonal is the same.

Objective :-
1) To study the creation of matrix using list in python.
2) To understand the concept of magic matrix.

Theory :-
A magic square is a n×n matrix of integers 1 to n^2, such that the sum of each row, column and diagonal is same. This sum is called magic constant or magic sum. It depends only on N and has the following value

$$M = \frac{n(n^2+1)}{2}.$$

Example:-

Magic square when n = 3.

2	7	6
9	5	1
4	3	8

Sum in each row, column and digonal = $\frac{3(3^2+1)}{2} = 15$.

Algorithm:-

Step 1- Start.

Step 2- Display menu to the user.

Step 3- If user wants to generate magic square matrix then accept the size of the matrix.

Step 4- Create magic square matrix for the given order and display it.

Step 5- If user wants to check a matrix for magic square then accept the order of the matrix and all the elements in matrix.

Step 6- Calculate the sum of elements of each row, column and digonal.

Step 7- If all the sum are equal, then it is a magic square.

Step 8 - If user wants to continue then go to step 2.

Step 9 - Otherwise, Stop.

Analysis:-

Time complexity of function are:-

1) Magic Matrix () → $O(n)$
2) print Matrix () → $O(n^2)$
3) check Matrix () → $O(n^2+n)$

Conclusion:-

Hence, we have created and checked magic square matrix.

GROUP A-07:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

PROGRAM:---

```
# GROUP A - 7 Write a python Program for magic square. A magic square is an
arrangemen0t of the numbers
# from 1 to N^2 (N-squared) in an NxN matrix, with each number occurring
exactly once, and such that the
# sum of the entries of any row, any column, or any main diagonal is the
same. Perform follwing operations
# a) Generate Magic Square
# b) Check whether matrix is magic square or not

# Generate odd sized magic squares


def generatesquare(n):
    magicsquare = [[0 for x in range(n)]
                   for y in range(n)]
    i = n / 2
    j = n - 1

    num = 1
    while num <= (n * n):
        if i == -1 and j == n:
            j = n - 2
            i = 0
        else:
            if j == n:
                j = 0
            if i < 0:
                i = n - 1

        if magicsquare[int(i)][int(j)]:
            j = j - 2
            i = i + 1
            continue
        else:
            magicsquare[int(i)][int(j)] = num
            num = num + 1

        j = j + 1
        i = i - 1
    print("Magic Squre for n =", n)
    print("Sum of each row or column or diagonal i.e Magic Number is :
{0:.0f}".format(n * (n * n + 1) / 2), "\n")
    print_mat(magicsquare, n)


# Printing magic square

def print_mat(t, n):
    for i in range(0, n):
        for j in range(0, n):
```

```
            print(t[i][j], end=" ")
        print()


# Determine whether a given matrix is magic matrix or not

def isMagicSquare(mat, n):
    s = 0                           # calculate the sum of the prime diagonal
    for i in range(0, n):
        s = s + mat[i][i]

    s2 = 0                          # Calculate the sum of the secondary
diagonal
    for i in range(0, n):
        s2 = s2 + mat[i][n - i - 1]
    if (s != s2):
        return False
    for i in range(0, n):           # For sums of Rows
        rowsum = 0;
        for j in range(0, n):
            rowsum += mat[i][j]
        if (rowsum != s):           # check if every row sum is equal to prime
diagonal sum
            return False
    for i in range(0, n):           # For sums of Columns
        colsum = 0
        for j in range(0, n):
            colsum += mat[j][i]
        if (s != colsum):           # check if every column sum is equal to
prime diagonal sum
            return False
    return True


flag = 1

while flag == 1:
    menu = " /~~~~~~~~~~MENU~~~~~~~~~~/ \n" \
           "1. Generate Magic Square \n" \
           "2. Determine whether matrix is magic square or not \n" \
           "3. Exit"

    print(menu)
    choice = int(input("Enter your choice : "))
    if choice == 1:
        n = int(input(" Enter the size of Magic square : "))
        generatesquare(n)
    elif choice == 2:
        n = int(input(" Enter the size of Magic square : "))


# Accept the matrix of size (n X n) and Initialize matrix
        mat = []
        print("Enter the elements rowwise:")


# For user input
```

```
        for i in range(0, n):  # A for loop for row entries
            a = []
            for j in range(0, n):  # A for loop for column entries
                a.append(int(input("Enter element : ")))
            mat.append(a)
        if (isMagicSquare(mat, n)):
            print("Magic Square")
        else:
            print("Not a magic Square")
    else:
        print("Wrong Choice Please Choose Another Option ")
        flag = 0
```

OUTPUT:---

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 1

Enter the size of Magic square : 5

Magic Squre for n = 5

Sum of each row or column or diagonal i.e Magic Number is : 65

9 3 22 16 15

2 21 20 14 8

25 19 13 7 1

18 12 6 5 24

11 10 4 23 17

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 2

Enter the size of Magic square : 5

Enter the elements rowwise:

Enter element : 1

Enter element : 2

Enter element : 3

Enter element : 4

Enter element : 5

Enter element : 6

Enter element : 7

Enter element : 8

Enter element : 9

Enter element : 10

Enter element : 11

Enter element : 12

Enter element : 13

Enter element : 14

Enter element : 15

Enter element : 16

Enter element : 17

Enter element : 18

Enter element : 19

Enter element : 20

Enter element : 21

Enter element : 22

Enter element : 23

Enter element : 24

Enter element : 25

Not a magic Square

/~~~~~~~~~~MENU~~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 2

Enter the size of Magic square : 5

Enter the elements rowwise:

Enter element : 9

Enter element : 3

Enter element : 22

Enter element : 16

Enter element : 15

Enter element : 2

Enter element : 21

Enter element : 20

Enter element : 14

Enter element : 8

Enter element : 25

Enter element : 19

Enter element : 13

Enter element : 7

Enter element : 1

Enter element : 18

Enter element : 12

Enter element : 6

Enter element : 5

Enter element : 24

Enter element : 11

Enter element : 10

Enter element : 4

Enter element : 23

Enter element : 17

Magic Square

/~~~~~~~~~MENU~~~~~~~~~/

1. Generate Magic Square

2. Determine whether matrix is magic square or not

3. Exit

Enter your choice : 3

Wrong Choice Please Choose Another Option

Process finished with exit code 0

* Data Structure Lab (DSL) - Practical Number - 4. (Group - B)

Name :- Koustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a python program to perform ternary search.

Aim :-
Write a python program to maintain club members, sort on roll number in ascending order. Write function "Ternary Search" to search whether particular student is member of club or not. Ternary search is modified binary search that divides array into 3 halves instead of two.

Objective :-
To understand and implement ternary search in python.

Theory :-
Ternary search is a divide and conquer algorithm that can be used to find an element in an array. It is similar to binary search where we divide the array into two parts but in this algorithm, we divide the array into three parts and determine which has the key.

Steps to perform Ternary Search:-

1) First, we compare the key with the element at mid1. If found equal, we return mid1.

2) If not, then we compare the key with the element at mid2. If found equal, we return mid2.

3) If not, then we check whether the key is less than element at mid1. If yes, then we recur to first half of array.

4) If not, then we check whether the key is greater than the element at mid2. If yes, then we recur to third half of array.

5) If not, then we recur to the second half of the array.

Algorithm:-

Step1 - Start

Step2 - Display menu to user and accept his choice.

Step3 - If user enter 1, accept the roll number in an array.

Step4 - If user enter 2, then display the roll number.

Step5 - If user enter 3, then sort the roll numbers using selection sort and store the sorted roll number in another array.

Step 6 - If user enter 4, then accept the roll number to be searched

Step 7 - Search the roll number in the sorted array using non recursive sear

Step 8 - Display the index of the roll number if found.

Step 9 - If user enters 5, then accept the roll number to be searched.

Step 10 - Search the roll number in the sorted array using recursive ternary search.

Step 11 - Display the index of the roll number if found.

Step 12 - Go to step 2 if user wants to continue.

Step 13 - Stop

Analysis :-

The time complexity of Ternary Search is $O(\log_3^n)$.

Conclusion :-

Hence, we have performed ternary search in an array.

Group B-13:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:-

```
"""
Write a python program to maintain club members, sort on roll numbers
in ascending order. Write function "Ternary_Search" to search
whether particular student is member of club or not.
Ternary search is modified binary search that divides
array into 3 halves instead of two.
"""

import array as arr

def Ternary_Search(arr,ele):
    left=0
    right=len(arr)-1
    while(left<=right):
        mid1=left+(right-left)//3
        mid2=left+2*(right-left)//3

        if(ele==arr[left]):
            return left
        elif(ele==arr[right]):
            return right
        elif(ele<arr[left] or ele>arr[right]):
            return -1
        elif(ele<=arr[mid1]):
            right=mid1
        elif(ele>=arr[mid2]):
            left=mid2
        elif(ele>arr[mid1] and ele<arr[mid2]):
            left=mid1+1
            right=mid2-1
    return -1

def Recursive_Ternary(arr,ele,left,right):

    if(left<=right):
        mid1=left+(right-left)//3
        mid2=right-(right-left)//3
        if(ele==arr[mid1]):
            return mid1
        if(ele==arr[mid2]):
            return mid2

        if(ele<arr[mid1]):
            return Recursive_Ternary(arr, ele, left, mid1-1)
        elif(ele>arr[mid2]):
            return Recursive_Ternary(arr, ele, mid2+1, right)
        else:
            return Recursive_Ternary(arr, ele, mid1+1, mid2-1)
    return -1

def accept():
```

```
    A=arr.array('I',[])
    n=int(input("Enter number of students: "))
    for i in range(0,n):
        A.append(int(input("Enter roll number: ")))
    return A

def display(A):
    for i in range(0, len(A)):
        print("\t", A[i], end=" ")
    print()

def sel_sort(A):
    for i in range(len(A)):
        for j in range(i+1,len(A)):
            if(A[j]<A[i]):
               temp=A[i]
               A[i]=A[j]
               A[j]=temp
    return A

A=arr.array('I',[])
sort_A=arr.array('I',[])

while True:

    print("\n1)Accept roll number \n2)Print roll number \n3)sort roll
numbers\n4)Non-recursive Ternary Search \n5)Recursive Ternary Search
\n6)Exit")
    ch=int(input("Enter your choice: "))

    if(ch==1):
        A=accept()

    elif(ch==2):
        display(A)

    elif(ch==3):
        sort_A=sel_sort(A)
        print("The sorted roll numbers are:")
        display(sort_A)

    elif(ch==4):
        ele=int(input("Enter roll number to be searched: "))
        r=Ternary_Search(sort_A, ele)
        if(r==-1):
            print("Roll number not found!!")
        else:
            print("The roll number", ele, "is present at index", r)
    elif(ch==5):
        ele=int(input("Enter roll number to be searched: "))
        r=Recursive_Ternary(sort_A, ele, 0, len(sort_A)-1)
        if(r==-1):
            print("Roll number not found!!")
        else:
            print("The roll number", ele, "is present at index", r)
    elif(ch==6):
        print("Thank you")
```

```
        break
    else:
        print("Wrong choice")
        break
```

OUTPUT:-

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 1

Enter number of students: 7

Enter roll number: 16

Enter roll number: 17

Enter roll number: 18

Enter roll number: 20

Enter roll number: 25

Enter roll number: 78

Enter roll number: 69

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 2

16 17 18 20 25 78 69

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 3

The sorted roll numbers are:

16 17 18 20 25 69 78

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 4

Enter roll number to be searched: 20

The roll number 20 is present at index 3

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 5

Enter roll number to be searched: 20

The roll number 20 is present at index 3

1)Accept roll number

2)Print roll number

3)sort roll numbers

4)Non-recursive Ternary Search

5)Recursive Ternary Search

6)Exit

Enter your choice: 6

Thank you

* Data Structure Lab (DSL) - Parbetical Number - 5 (Group - B)

Name :- Kaustubh Shrikant Kabra
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a python program to perform selection and bubble sort.

Aim :-
Write a python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using.
1) Selection Sort
2) Bubble sort and display top five scores.

Objective :-
1) To study the concept of array in python.
2) To learn and understand selection sort.
3) To learn and understand bubble sort.

Theory :-
Bubble Sort :-
This is the simplest kind of sorting method. It is performed in several iterations which are called passes. In this

algorithm, the adjacent elements are compared and their positions are swapped if they are not in intended order.

Selection Sort :-

Selection sort is an algorithm that selects the smallest element from an unsorted array in each interation and places that element at the begining of the unsorted list.

Algorithm :-

Step 1 - Start

Step 2 - Display menu to user and accept his choice.

Step 3 - If user enter 1, then accept the total number of students and their percentage. Store their percentage in an array.

Step 4 - If user enter 2, then display the percentage.

Step 5 - If user enter 3, then sort the array using bubble sort technique.

Step 6 - Display the sorted array and the top five scores.

Step 7 - If user enters 4, then sort the array using selection sort technique

Step 8 - Display the sorted array and the top five scores.

Step 9 - Go to step 2, if user wants to continue.

Step 10 - Stop

Analysis :-
The time complexity of bubble sort is $O(n \log n)$ and the time complexity of selection sort is $O(n^2)$.

Conclusion:-
Hence, we have performed selection sort and bubble sort on an array.

Group B-14:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:-

```
"""
Write a python program to store first year percentage of students in array.
Write function for sorting array of floating point numbers in ascending
order using a) Selection Sort b) Bubble sort and display top five scores.
"""
import array as arr

def accept():
    a=arr.array('f',[])
    n=int(input("Enter number of students: "))
    for i in range(n):
        a.append(float(input("Enter first year percentage of student {}:
".format(i+1))))
    return a

def print_per(arr):
    for i in range(0, len(arr)):
        print("\t {0:.2f}".format(arr[i]), end=" ")
    print()

def bubble_sort(arr):
    flg=0
    for i in range(len(arr)):
        for j in range(0,(len(arr)-i-1)):
            if(arr[j]>=arr[j+1]):
                flg=1
                temp=arr[j]
                arr[j]=arr[j+1]
                arr[j+1]=temp
        if(flg==0):
            break
    print("\nElements after sorting are-\n")
    print_per(arr)
    top_five(arr)

def selection_sort(arr):
    for i in range(len(arr)):
        for j in range(i+1,len(arr)):
            if(arr[j]<=arr[i]):
                temp=arr[i]
                arr[i]=arr[j]
                arr[j]=temp
    print("\nElements after sorting are-\n")
    print_per(arr)
    top_five(arr)

def top_five(arr):
    j=0
    print("\nThe top scores are- ")
    for i in reversed(arr):
        print("\t {0:.2f}".format(i), end=" ")
```

```
        j=j+1
        if(j==5):
            break
    print()

A=arr.array('f',[])
sort_A=arr.array('f',[])

while(True):
    print("\n**********MENU**********")
    print("Enter 1 to accept percentage")
    print("Enter 2 to display percentage")
    print("Enter 3 to sort using bubble sort technique")
    print("Enter 4 to sort using selection sort technique")
    print("Enter 5 to exit")
    c=int(input("Enter your choice: "))
    if(c==1):
        A=accept()
    elif(c==2):
        print_per(A)
    elif(c==3):
        bubble_sort(A)
    elif(c==4):
        selection_sort(A)
    elif(c==5):
        print("Thank you")
        break
    else:
        print("Enter correct choice")
```

Output:-

**********MENU**********

Enter 1 to accept percentage

Enter 2 to display percentage

Enter 3 to sort using bubble sort technique

Enter 4 to sort using selection sort technique

Enter 5 to exit

Enter your choice: 1

Enter number of students: 7

Enter first year percentage of student 1: 86

Enter first year percentage of student 2: 76

Enter first year percentage of student 3: 48

Enter first year percentage of student 4: 57

Enter first year percentage of student 5: 96

Enter first year percentage of student 6: 84

Enter first year percentage of student 7: 75

**********MENU**********

Enter 1 to accept percentage

Enter 2 to display percentage

Enter 3 to sort using bubble sort technique

Enter 4 to sort using selection sort technique

Enter 5 to exit

Enter your choice: 2

86.00 76.00 48.00 57.00 96.00 84.00 75.00

**********MENU**********

Enter 1 to accept percentage

Enter 2 to display percentage

Enter 3 to sort using bubble sort technique

Enter 4 to sort using selection sort technique

Enter 5 to exit

Enter your choice: 3

Elements after sorting are-

48.00 57.00 75.00 76.00 84.00 86.00 96.00

The top scores are-

96.00 86.00 84.00 76.00 75.00

**********MENU**********

Enter 1 to accept percentage

Enter 2 to display percentage

Enter 3 to sort using bubble sort technique

Enter 4 to sort using selection sort technique

Enter 5 to exit

Enter your choice: 4

Elements after sorting are-

48.00 57.00 75.00 76.00 84.00 86.00 96.00

The top scores are-

96.00 86.00 84.00 76.00 75.00

**********MENU**********

Enter 1 to accept percentage

Enter 2 to display percentage

Enter 3 to sort using bubble sort technique

Enter 4 to sort using selection sort technique

Enter 5 to exit

Enter your choice: 5

Thank you

* Data Structure Lab (DSL) - Practical Number - 6 (Group - B)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a python program to demonstrate quick sort.

Aim :-
Write a python program to store first year percentage of student in array. Write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.

Objective :-
To learn and implement quick sort on array of floating numbers.

Theory :-
Quick Sort :-
Quick Sort is an algorithm based on divide and conquer approach in which the array is split into subarrays and these subarrays are recursively called to sort elements.

Algorithm :-

Step 1 - Start

Step 2 - Display menu to user and enter his choice.

Step 3 - If user enter 1, then accept the percentage and store them in an array for N students.

Step 4 - If user enter 2, then print the percentage for all the students.

Step 5 - If user enter 3, then sort the percentage using quick sort technique is ascending order.

Step 6 - Store the sorted percentage in another array.

Step 7 - Display the sorted percentage.

Step 8 - Display the top five scores from the sorted array.

Step 9 - Go to step 2 if user wants to continue.

Step 10 - Stop.

Analysis :-

The worst case time complexity of quick sort is $O(n^2)$ and average case is $O(n \log n)$.

Conclusion :-

Hence we have demonstrated quick sort on an array.

Group B-16:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:-

```
"""
Write a python program to store first year percentage of students in array.
Write function for sorting array of floating point numbers in ascending
order using quick sort and display top five scores.
"""

import array as arr

def accept():
    a=arr.array('f',[])
    n=int(input("Enter number of students: "))
    for i in range(n):
        a.append(float(input("Enter first year percentage of student {}:
".format(i+1))))
    return a

def print_per(arr):
    for i in range(0, len(arr)):
        print("\t {0:.2f}".format(arr[i]), end=" ")
    print()

def partition(arr,start,end):
    pivot=arr[start]
    low=start+1
    high=end

    while True:

        while(low<=high and arr[low]<=pivot):
            low=low+1

        while(low<=high and arr[high]>=pivot):
            high=high-1

        if(low<=high):
            arr[low],arr[high]=arr[high],arr[low]

        else:
            break

    arr[start],arr[high]=arr[high],arr[start]
    return high

def quick_sort(arr,start,end):
    if(start>=end):
        return

    p=partition(arr,start,end)
    quick_sort(arr,start,p-1)
    quick_sort(arr,p+1,end)
    return arr
```

```
def top_five(arr):
    j=0
    print("\nThe top scores are- ")
    for i in reversed(arr):
        print("\t {0:.2f}".format(i), end=" ")
        j=j+1
        if(j==5):
            break
    print()

A=arr.array('f',[])
sort_A=arr.array('f',[])

while True:

    print("\n1)Accept percentage \n2)Print percentage \n3)Sort and display
top scores\n4)Exit \n")
    ch=int(input("Enter your choice: "))

    if(ch==1):
        A=accept()

    elif(ch==2):
        print_per(A)

    elif(ch==3):
        print("Percentages after sorting-")
        sort_A=quick_sort(A,0,len(A)-1)
        print_per(sort_A)
        top_five(sort_A)

    elif(ch==4):
        print("Thank you")
        break
    else:
        print("Wrong choice")
        break;
```

OUTPUT:-

1)Accept percentage

2)Print percentage

3)Sort and display top scores

4)Exit

Enter your choice: 1

Enter number of students: 7

Enter first year percentage of student 1: 35

Enter first year percentage of student 2: 21

Enter first year percentage of student 3: 75

Enter first year percentage of student 4: 94

Enter first year percentage of student 5: 75

Enter first year percentage of student 6: 16

Enter first year percentage of student 7: 82

1)Accept percentage

2)Print percentage

3)Sort and display top scores

4)Exit

Enter your choice: 2

35.00 21.00 75.00 94.00 75.00 16.00 82.00

1)Accept percentage

2)Print percentage

3)Sort and display top scores

4)Exit

Enter your choice: 3

Percentages after sorting-

16.00 21.00 35.00 75.00 75.00 82.00 94.00

The top scores are-

94.00 82.00 75.00 75.00 35.00

1)Accept percentage

2)Print percentage

3)Sort and display top scores

4)Exit

Enter your choice: 4

Thank you

* Data Structure Lab (DSL) - Practical Number - 7 (Group - C)

Name :- Shaustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
Write a C++ program to create a students club.

Aim :-
Depart of computer Engineering has students club named 'Pinnacle Club'. Student of 2nd, 3rd and 4th year of department can be granted membership on request. Similarly, one may cancel the membership of club. Store student PRN and Name. Write functions to:-
a) Add and delete members, president and secretary
b) Compute total number of members.
c) Display members.
d) Two linked list exists for two divisions. Concatenate two lists.

Objective :-
1) To study the concept of linked lists.
2) To understand various operations on singly linked list.

Theory:-

Linked List-

A linked list is a set of nodes where each node has two fields 'data' and 'link'. The 'data' field stores actual piece of information and 'link' field is used to point to the next node. Basically, 'link' field is nothing but address only.

Types of linked list:-

1) Singly linked list.
2) Singly circular linked list.
3) Doubly linear linked list.
4) Doubly circular linked list.

Operations on singly linked list:-

1) Creation
2) Insertion
3) Deletion
4) Reverse
5) Search
6) Display

Algorithm:-

Step 1- Start

Step 2- Create a class node which consists the structure of the

Step 3 - Create a friend class dll which consists of the head pointer, constructor and functions to operate on the linked list.

Step 4 - Display menu to the user and accept his choice.

Step 5 - If user enter 1, then accept details of the president.

Step 6 - Add the president details in first node of linked list.

Step 7 - If user enter 2, then accept details of the secretary.

Step 8 - Add the secretary details in last node of linked list.

Step 9 - If user enter 3, then accept and add member details after a particular node in the linked list.

Step 10 - If user enter 4, then delete the president.

Step 11 - If user enter 5, then delete the secretary.

Step 12 - If user enter 6, then accept the PRN number of a member and delete it.

Step 13 - If user enter 7, then display PRN number and names of all the members.

Step 14 - If user enter 8, then display total number of members in the club.

Step 15 - If user enter 9, then reverse the linked list.

Step 16 - If user enter 10, the concatenate two divisions.

Step 17 - Go to step 4, if user wants to continue.

Step 18 - Stop.

Analysis:-

Time complexity on various operations are -

1) For insert_president, insert_secretary(), del_president, del_secretary it is O(1)

2) For insert_member, del_member, display(), count(), it is O(n)

3) For concatenate and reverse - it is O(n).

Conclusion:-

Thus, we have created a students club using singly linked list.

GROUP C-19:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

PROGRAM:---

```
#include<iostream>

#include<string.h>

using namespace std;

 struct node

{    int prn,rollno;

     char name[50];

     struct node *next;

};

class info

{       node
*s=NULL,*head1=NULL,*temp1=NULL,*head2=NULL,*temp2=NULL,*head=NULL,*temp=NULL;

       int b,c,i,j,ct;

       char a[20];

       public:


          node *create();

          void insertp();

          void insertm();

          void delm();

          void delp();

          void dels();

          void display();

          void count();
```

```
        void reverse();
        void rev(node *p);
        void concat();

} ;
node *info::create()
{   node *p=new(struct node);
    cout<<"Enter name of student:-- \n";
    cin>>a;
    strcpy(p->name,a);
      cout<<"\n  Enter prn no. of student:-- \n";
    cin>>b;
    p->prn=b;
    cout<<"Enter student rollno:-- \n";
    cin>>c;
    p->rollno=c;
    p->next=NULL;
    return  p;
 }
 void info::insertm()
 {
     node *p=create();

   if(head==NULL)
   {   head=p;
```

```
  }
 else
 {     temp=head;
       while(temp->next!=NULL)
       {    temp=temp->next;   }
          temp->next=p;
  }

}
  void info::insertp()
{
    node *p=create();

  if(head==NULL)
  {    head=p;
  }
 else
 {     temp=head;
        head=p;
          head->next=temp->next;

  }

}
```

```
void info::display()
{       if(head==NULL)
        {   cout<<"linklist is empty \n";
        }
        else
        {
         temp=head;
                cout<<"    Prn    rolln0   NAME  \n";
                while(temp->next!=NULL)
                {   cout<<"   \n"<<temp->prn<<"   "<<temp->rollno<<"   "<<temp->name;
                    temp=temp->next;
                }
               cout<<"   "<<temp->prn<<"   "<<temp->rollno<<"   "<<temp->name;
        }
}
void info::delm()
{  int m,f=0;
  cout<<"\n Enter the prn no. of student whose data you want to delete:-- \n";
   cin>>m;
   temp=head;
   while(temp->next!=NULL)
   {
      if(temp->prn==m)
       {        s->next=temp->next;
                delete(temp);       f=1;
```

```
        }
        s=temp;
        temp=temp->next;
    }     if(f==0)
        {   cout<<"\n Sorry memeber not deleted \n";   }
}
void info::delp()
{    temp=head;
   head=head->next;
     delete(temp);
   }
   void info::dels()
{
   temp=head;
   while(temp->next!=NULL)
   {   s=temp;
   temp=temp->next;
   }    s->next=NULL;
     delete(temp);

}
void info::count()
{     temp=head;    ct=0;
      while(temp->next!=NULL)
      {    temp=temp->next; ct++;   }
```

```
        ct++;
        cout<<"  Count of members is:-- \n"<<ct;

}
void info::reverse()
{    rev(head);   }
void info::rev(node *temp)
{    if(temp==NULL)
     { return;   }
     else
     {   rev(temp->next); }
     cout<<"    "<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;
}

void info::concat()
{  int k,j;
   cout<<"Enter no. of members in list1;-- \n";
   cin>>k;
    head=NULL;
   for(i=0;i<k;i++)
   { insertm();
     head1=head;

   } head=NULL;
  cout<<"Enter no. of members in list2:-- \n";
```

```
    cin>>j;
    for(i=0;i<j;i++)
    { insertm();
      head2=head;

    } head=NULL;

     temp1=head1;
   while(temp1->next!=NULL)
   {   temp1=temp1->next;   }
        temp1->next=head2;


                    temp2=head1;
                   cout<<"    PRN    ROLL_NO    NAME   \n";
                 while(temp2->next!=NULL)
                 {
                   cout<<"\n   "<<temp2->prn<<"   "<<temp2->rollno<<"    "<<temp2->name<<"\n";;
                   temp2=temp2->next;
                 }
                   cout<<"\n   "<<temp2->prn<<"   "<<temp2->rollno<<"    "<<temp2->name<<"\n";
  }
int main()
{ info s;
int i;
```

```
  char ch;
do{
  cout<<"\n  1. To Insert President  ";
  cout<<"\n  2. To Insert Secretary ";
  cout<<"\n  3. To Insert Member";
  cout<<"\n  4. To Delete President  ";
  cout<<"\n  5. To Delete Secretary ";
  cout<<"\n  6. To Delete Member";
  cout<<"\n  7. To Display Data  ";
  cout<<"\n  8. Count of Members ";
  cout<<"\n  9. To display reverse of string  ";
  cout<<"\n  10.To concatenate two strings \n ";
  cout<<"\n Choice the Options ";
  cin>>i;
 switch(i)
 {      case 1:   s.insertp();
                  break;
        case 2:   s.insertm();
                  break;
        case 3:   s.insertm();
                  break;
        case 4:   s.delp();
                  break;
        case 5:   s.delm();
```

```
                    break;
            case 6:   s.dels();
                    break;
            case 7:   s.display();
                    break;
            case 8:   s.count();
                    break;
            case 9:   s.reverse();
                    break;
            case 10:  s.concat();

                    break;
            default:  cout<<"\n unknown choice \n";
        }
          cout<<"\n Do you want to continue enter y/n \n";
          cin>>ch;

     }while(ch=='y'||ch=='n');

  return 0;
}
```

OUTPUT:---

1. To Insert President
2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 1

Enter name of student:--

Kaustubh

Enter prn no. of student:--

741852963

Enter student rollno:--

20

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 1

Enter name of student:--

Harsh

Enter prn no. of student:--

741852369

Enter student rollno:--

25

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 2

Enter name of student:--

Onasvee

Enter prn no. of student:--

741258963

Enter student rollno:--

16

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 3

Enter name of student:--

Unnati

Enter prn no. of student:--

147852963

Enter student rollno:--

18

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 3

Enter name of student:--

Akash

Enter prn no. of student:--

852147963

Enter student rollno:--

78

Do you want to continue enter y/n

y

1. To Insert President
2. To Insert Secretary
3. To Insert Member
4. To Delete President
5. To Delete Secretary
6. To Delete Member
7. To Display Data
8. Count of Members
9. To display reverse of string
10. To concatenate two strings

Choice the Options 7

```
   Prn     rolln0    NAME

741852369    25    Harsh
741258963    16    Onasvee
147852963    18    Unnati    852147963    78    Akash
 Do you want to continue enter y/n
y

  1. To Insert President
  2. To Insert Secretary
  3. To Insert Member
  4. To Delete President
  5. To Delete Secretary
  6. To Delete Member
  7. To Display Data
  8. Count of Members
  9. To display reverse of string
  10.To concatenate two strings

 Choice the Options 4

 Do you want to continue enter y/n
y

  1. To Insert President
```

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 7

Prn rolln0 NAME

741258963 16 Onasvee

147852963 18 Unnati 852147963 78 Akash

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 5

Enter the prn no. of student whose data you want to delete:--

852147963

Sorry memeber not deleted

Do you want to continue enter y/n

y

1. To Insert President

2. To Insert Secretary

3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 8

Count of members is:--

3

Do you want to continue enter y/n

y

1. To Insert President
2. To Insert Secretary
3. To Insert Member
4. To Delete President
5. To Delete Secretary
6. To Delete Member
7. To Display Data
8. Count of Members
9. To display reverse of string
10.To concatenate two strings

Choice the Options 9

852147963 78 Akash 147852963 18 Unnati 741258963 16 Onasvee

Do you want to continue enter y/n

y

1. To Insert President
2. To Insert Secretary
3. To Insert Member

4. To Delete President

5. To Delete Secretary

6. To Delete Member

7. To Display Data

8. Count of Members

9. To display reverse of string

10.To concatenate two strings

Choice the Options 10

Enter no. of members in list1;--

2

Enter name of student:--

Onasvee

Enter prn no. of student:--

741258963

Enter student rollno:--

78

Enter name of student:--

Akash

Enter prn no. of student:--

963852741

Enter student rollno:--

78

Enter no. of members in list2:--

1

Enter name of student:--

Unnati

Enter prn no. of student:--

741258963

Enter student rollno:--

18

PRN	ROLL_NO	NAME
741258963	78	Onasvee
963852741	78	Akash
741258963	18	Unnati

Do you want to continue enter y/n

n

* Data Structure Lab (DSL) - Practical Number - 8 (Group - C)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering.
Div :- A Roll Number :-
Batch :-
Department :- Computer Department.
College :- AISSMS's IOIT

Title :-
Write a C++ program to implement ticket booking system.

Aim :-
The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with use array to store pointer (head pointer) to each row on demand
a) The list of available seats is to be displayed.
b) The seats are to be booked.
c) The booking can be cancelled.

Objectives :-
1) To study the concept of doubly circular linked list.
2) To understand operations on doubly circular linked list.

Theory:-

Doubly circular Linked List:-

Doubly circular linked list has properties of both doubly linked list and circular linked list in which two consecutive elements are linked or connected by previous and next pointer. The last node points to the first node by next pointer and also the first node points to the last node by previous pointer.

Advantages:-

1) List can be traversed from both the direction i.e from head to tail or from tail to head.

2) Jumping from head to tail or from tail to head is done in constant time O(1).

3) Circular doubly linked list are used for implementation of advanced data structures like Fibonacci Heap.

Algorithm:-

Step1- Start

Step2- Create structure for doubly circular linked list.

Step3- Create a class ticket with constructor and member methods to operate on doubly circular linked list.

Step4- Using the display method, display all the seats with their status being un-booked.

Step 5 - If user wants to book tickets, then accept the seat row and column number to be booked.

Step 6 - If the particular seat is not book, then change its status to booked ('B').

Step 7 - If user wants to cancel the tickets, then accept the seat row and column number to be cancelled.

Step 8 - If that particular seat is booked, then change its status to not booked ('A')

Step 9 - Display the seats and their status if user wants to see the unbooked seats.

Step 10 - Go to step 4 if user wants to continue.

Step 11 - Stop.

Analysis :-

Time complexity of

1) Constructor is - $O(n^2)$
2) Display is - $O(n^2)$
3) Booking and cancelling - $O(n)$
4) multiple_ticket - $O(n)$.

Page :

Conclusion:-

Hence, we have implemented ticket booking system using doubly circular linked list.

GROUP C-19:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

PROGRAM:---

```
#include<iostream>
using namespace std;
 struct node
{
    int seatc,seatr;
    string status;
    struct node *next ,*prev;
}*head[10],*last[10];

class ticket
{
public:
    ticket()
    {
        for(int i=1 ; i<=10 ; i++)
        {
            head[i]=last[i]=NULL;
            struct node* temp;
            for(int j=1 ; j<=7 ; j++)
```

```
        {
            temp=create_node(i,j);
            if(head[i]==last[i] && head[i]==NULL)
            {
                head[i]=last[i]=temp;
                head[i]->next=last[i]->next=NULL;
                head[i]->prev=last[i]->prev=NULL;
            }
            else  //Insertion at beginning
            {
                temp->next=head[i];
                head[i]->prev=temp;
                head[i]=temp;
                head[i]->prev=last[i];
                last[i]->next=head[i];
            }
        }
    }
}
node* create_node(int x,int y)
{
    struct node*temp;
```

```
    temp=new(struct node);
    if(temp==NULL)
    {
        cout<<"\nMemory not allocated";
        return 0;
    }
    else
    {
        temp->seatr=x;
        temp->seatc=y;
        temp->status="A";
        temp->next=NULL;
        temp->prev=NULL;
        return temp;
    }
}
void book()
{
    int x,y;
    cout<<"\nEnter row and column";
    cin>>x>>y;
    struct node* temp;
```

```
    temp=head[x];
    for(int i=1 ; i<=7 ; i++)
    {
        if(temp->seatc==y)
        {
            if(temp->status=="A")
            {
                temp->status="B";
            }
            else
            {
                cout<<"\nSORRY !! Already booked!!";
            }
        }
        temp=temp->next;
    }
    display();
}
void cancel(){
    int x,y;
    cout<<"\nEnter row and column to cancel booking : ";
    cin>>x>>y;
```

```
    struct node* temp;
    temp=head[x];
    for(int i=1 ; i<=7 ; i++)
    {
        if(temp->seatc==y)
        {
            if(temp->status=="B")
            {
                temp->status="A"
            }
            else
            {
                cout<<"\nSORRY !! Already unbooked!!";
            }
        }
        temp=temp->next;
    }
    display();
}
void display()
{
    struct node* temp;
```

```
        for(int j=1 ; j<=10 ; j++)
        {
            temp=head[j];
            for(int i=1 ; i<=7 ; i++)
            {
                cout<<temp->seatr<<","<<temp->seatc<<temp->status<<"\t";
                temp=temp->next;
            }
            cout<<"\n";
        }
    }
};
int main()
{
    ticket t;
    int ch;
    t.display();
    do{
        cout<<"\n1.Book Ticket \n2.Cancel Booking  \n3.EXIT";
        cout<<"\n Enter Your Choice:";
        cin>>ch;
        switch(ch)
```

```
        {
        case 1:t.book();break;
        case 2:t.cancel();break;
        }
    }while(ch!=3);
    return 0;
}
```

OUTPUT:---

```
1,7A   1,6A   1,5A   1,4A   1,3A   1,2A   1,1A
2,7A   2,6A   2,5A   2,4A   2,3A   2,2A   2,1A
3,7A   3,6A   3,5A   3,4A   3,3A   3,2A   3,1A
4,7A   4,6A   4,5A   4,4A   4,3A   4,2A   4,1A
5,7A   5,6A   5,5A   5,4A   5,3A   5,2A   5,1A
6,7A   6,6A   6,5A   6,4A   6,3A   6,2A   6,1A
7,7A   7,6A   7,5A   7,4A   7,3A   7,2A   7,1A
8,7A   8,6A   8,5A   8,4A   8,3A   8,2A   8,1A
9,7A   9,6A   9,5A   9,4A   9,3A   9,2A   9,1A
10,7A  10,6A  10,5A  10,4A  10,3A  10,2A  10,1A


1.Book Ticket
2.Cancel Booking
3.EXIT
```

Enter Your Choice:--1

Enter row and column:--5

4

1,7A 1,6A 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4B 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT

Enter Your Choice:--1

Enter row and column:--5

4

SORRY !! Already booked!!1,7A 1,6A 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4B 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT

Enter Your Choice:--2

Enter row and column to cancel booking:-- 5

4

1,7A 1,6A 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4A 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT

Enter Your Choice:--1

Enter row and column:--5

4

1,7A 1,6A 1,5A 1,4A 1,3A 1,2A 1,1A

2,7A 2,6A 2,5A 2,4A 2,3A 2,2A 2,1A

3,7A 3,6A 3,5A 3,4A 3,3A 3,2A 3,1A

4,7A 4,6A 4,5A 4,4A 4,3A 4,2A 4,1A

5,7A 5,6A 5,5A 5,4B 5,3A 5,2A 5,1A

6,7A 6,6A 6,5A 6,4A 6,3A 6,2A 6,1A

7,7A 7,6A 7,5A 7,4A 7,3A 7,2A 7,1A

8,7A 8,6A 8,5A 8,4A 8,3A 8,2A 8,1A

9,7A 9,6A 9,5A 9,4A 9,3A 9,2A 9,1A

10,7A 10,6A 10,5A 10,4A 10,3A 10,2A 10,1A

1.Book Ticket

2.Cancel Booking

3.EXIT

Enter Your Choice:--3

* Data Structure Lab (DSL) - Practical Number - 9 (Group - D)

Name :- Kaustubh Shrikant Kulkarni.
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-
To check whether the given string is palindrome or not using stack in C++.

Aim :-
A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization and spaces are ignored. One way to check for a palindrome is to reverse the characters in the string and then compare them the original - in a palindrome, the sequence will be identical. Write a C++ program with functions -
1) To print original and reversal string using stack.
2) To check whether the given string is palindrome or not.

Objective :-
1) To study and learn stack.
2) To implement operations on stack.
3) To understand and check palindrome string.

Theory :-

A stack is an ordered list in which all insertions and deletions are made at one end called the top. It is a datastructure which passes LIFO i.e.

Palindrome String -

A palindrome is a string of character thats the same forward and backward. Typically, punctuation, capitalization and spaces are ignored.

Examples :-

1) Was it a car or a cat I saw?
2) I did, did I?
3) Top spot.

Algorithm :-

Step 1 - Start

Step 2 - Create a class palindrome which consists of stacks to store the string and member methods performing operations on stack.

Step 3 - Accept a string from the user and store it in stack.

Step 4 - Remove whitespaces, punctuations, special characters from the string. Also convert capital letters to small letters.

Step 5 - Display the updated string.

Step 6 - Reverse the updated string.

Step 7 - Display the reversed string.

Step 8 - Check each and every character in the reversed string and the updated string serially.

Step 9 - If each and every character is same, then it is a palindrome string.

Step 10 - Stop.

Analysis:-

Time complexity of:-

1) push() and pop() is - O(1).
2) Reverse(), remove_whitespace(), check_palin-drome() is - O(n).

Conclusion:-

Hence, we have checked for a palindrome string using stack.

GROUP D-25:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

PROGRAM:---

```
#include<iostream>
#include<string.h>
#include<ctype.h>
using namespace std;
#define MAX 50
class Stack
{
private:
   char data[MAX],str[MAX];
   int top,length,count;

   void pushData(char);
   char popData();

public:
   Stack()
   {
      top=-1;
      length=0;
```

```
        count=0;
    }
    void getString();
    void checkPalindrome();
    void extractString();
    void displayReverse();
};

int main()
{
    Stack obj;
    obj.getString();
    cout<<"\n Extracted string: ";
    obj.extractString();
    cout<<"\n Reverse of entered string: ";
    obj.displayReverse();
    obj.checkPalindrome();
    return 0;
}

void Stack::getString()
{
```

```
    cout<<"\n Enter a String: ";
    cin.getline(str,MAX);

    length=strlen(str);
}

void Stack::extractString()
{
    char temp[MAX];
    int i,j;
    for(i=0; i<length; i++)
    {
        temp[i]=str[i];
    }
    j=0;
    for(i=0; i<length; i++ )
    {
        if(isalpha(temp[i]))
        {
            str[j]=tolower(temp[i]);
            j++;
        }
```

```
    }

    length=j;   //update length with new str length
    for(int i=0; i<length; i++)
            cout<<str[i];
}

void Stack::checkPalindrome()
{
    for(int i=0; i<length; i++)
        pushData(str[i]);

    for(int i=0; i<length; i++)
    {
        if(str[i]==popData())
            count++;
    }

    if(count==length) {
        cout<<"\n Entered string is a Palindrome. \n";
    }
    else  cout<<"\n Entered string is not a Palindrome. \n";
```

```
}

void Stack::displayReverse()
{
   for(int i=length-1; i>=0; i--)
      cout<<str[i];
}

void Stack::pushData(char temp)
{
   if(top==MAX-1)
   {
      cout<<"\n Stack Overflow!!!";
      return;
   }

   top++;
   data[top]=temp;

}

char Stack::popData()
```

```
{
   if(top==-1)
   {
      cout<<"\n Stack Underflow!!!";
      return 0;
   }
   char temp=data[top];
   top--;
   return temp;
}
```

OUTPUT:-

Case1:--

Enter a String: Was it a car or a cat I saw

Extracted string: wasitacaroracatisaw

Reverse of entered string: wasitacaroracatisaw

Entered string is a Palindrome.

Case2:---

Enter a String: My name is Kaustubh.

Extracted string: mynameiskaustubh

Reverse of entered string: hbutsuaksiemanym

Entered string is not a Palindrome.

* Data Structure Lab (DSL) - Practical Number - 10 (Group - D)

Name :- Kaustubh Shrikant Kabra.
Class :- Second Year Engineering
Div :- A Roll Number :-
Batch :-
Department :- Computer Department
College :- AISSMS's IOIT.

Title :-

Write a C++ program using stack to check whether given expression is well parenthesized or not.

Aim :-

In any language program, mostly syntax error occurs due to unbalancing delimiter such as (), {}, []. Write a C++ program using stack to check whether given expression is well parenthesized or not.

Objective :-

1) To study the stack data structure.
2) To study the operations on stack.

Theory :-

A stack is an ordered list in which all the insertions and deletions are made at one end. It possesses the property of LIFO i.e. last in first out.

Stack Operations :-

1) Basically there are two important stack operations a) Push
b) Pop.

2) Performing push operation means we are inserting the element onto the stack. While, pop operation means we are removing the element from the stack.

3) Before pushing, we need to check stack full condition and before performing pop operation we need to check stack empty condition.

Algorithm :-

Step 1 - Start

Step 2 - Declare a class to create a stack, constructor and methods to perform operations on stack.

Step 3 - Accept an expression from the user.

Step 4 - Traverse to the expression and push the opening parentheses in stack.

Step 5 - If no parenthesis are present i.e. stack is empty, then display well balanced and stop.

Step 6 - Pop an element from the stack if for every opening parenthesis ('(', '[', '{'), there is a corresponding closing parenthesis

Step 7 - Repeat step 6 until stack becomes empty.

Step 8 - If stack becomes empty i.e. there is a closing parethesis for every opening parenthesis, return true.

Step 9 - If the return value is true, then the equation is well balanced.

Step 10 - Go to step 3, if user wants to check another expression.

Step 11 - Stop.

Analysis:-

Time complexity -

1) Display → O(n)
2) Push and pop → O(1)
3) Check parenthesis → O(n).

Conclusion:-

Hence, we have checked whether a given function's expression is well parenthesized or not.

GROUP D-26:--- KAUSTUBH SHRIKANT KABRA SE COMP-1 20

Program:---

```
#include <iostream>
#include<cstdio>
#include<cstdlib>
using namespace std;

#define MAX 50        /* Size of Stack */

class Stack
{
   char s[MAX];
   int top;
   public:
      Stack()
      {
         top=-1;
      }
      void push(char ch);
      char pop();
      bool isEmpty();
      bool isFull();
      bool checkParenthesis(char expr[]);
};
```

```
bool Stack::isEmpty()
{
   if(top==-1)
      return 1;
   else
      return 0;
}

bool Stack::isFull()
{
   if(top==MAX-1)
      return 1;
   else
      return 0;
}

void Stack::push(char ch)
{
   if(!isFull())
   {
      top++;
      s[top]=ch;
   }
}
```

```
char Stack::pop()
{
   if(!isEmpty())
   {
      char ch=s[top];
      top--;
      return ch;
   }
   else
      return '\0';

}

bool Stack::checkParenthesis(char expr[])
{
   char x;

   // Traversing the Expression
   for (int i=0; expr[i]!='\0'; i++)
   {
      if (expr[i]=='('||expr[i]=='['||expr[i]=='{')
      {
         // Push the element in the stack
         push(expr[i]);
         continue;
```

```
}

// IF current current character is not opening
// bracket, then it must be closing. So stack
// cannot be empty at this point.
if (isEmpty())
  return false;

switch (expr[i])
{
case ')':

   // Store the top element in a
   x = pop();
   if (x=='{' || x=='[')
      return false;
   break;

case '}':

   // Store the top element in b
   x = pop();

   if (x=='(' || x=='[')
      return false;
```

```
            break;


        case ']':


            // Store the top element in c
            x = pop();


            if (x =='(' || x == '{')
                return false;
            break;
        }
    }


    // Check Empty Stack
    return (isEmpty());
}


// Driver program to test above function


int main()
{
    char expr[50];
    int i=0,k=0;
    Stack st;
    cout<<"\nEnter  Expression: ";
```

```
    cin>>expr;
    if (st.checkParenthesis(expr))
        cout << "Balanced";
    else
        cout << "Not Balanced";

    return 0;
}
```

Output:-

Enter Expression: (){]

Not Balanced

Enter Expression: (){}

Balanced

Enter Expression: [{(a+b+c)*(a+b-c)}-{(a-b-c)*(a+c-b)}]

Balanced

* Data Structure Lab (DSL):- Practical Number - 12 (Group - E)

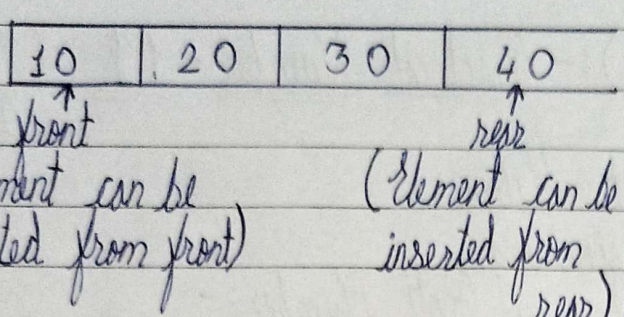
Name:- Kaustubh Shrikant Kabra.
Class:- Second Year Engineering. Roll Number:-
Div:- A
Batch:-
Department:- Computer Department
College:- AISSMS's IOIT.

Title:- Simulating Job Queue.

Aim:- Write a C++ program for simulating job queues. Write functions to add job and delete job from queue.

Objective:-
1. To study the concepts of Queues
2. To understand operations on Queues.

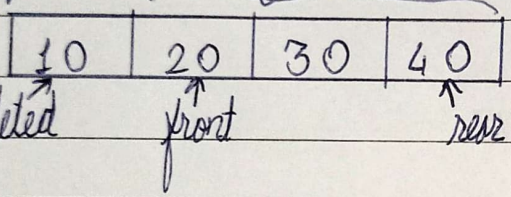
Theory:-
- Definition Queue:- It is an ordered collection of elements that has two named ends → front and rear, from the front end one can delete elements and from rear end can insert elements.
- Queue is also called as FIFO i.e. First In First Out.
- All elements are stored sequentially.
- Insertion of element:- Insertion of any element in the queue will always take place from the rear end.



Order of Insertion
① 10
② 20
③ 30
④ 40

- Deletion of element -

Deletion of any element in the queue takes by the front end always



Number 10 gets deleted logically

Before performing delete operation check whether the queue is empty or not.

Algorithm :-

Step 1: Start
Step 2: Define structure for queue.
Step 3: Read choice.
Step 4: If choice = insert
i) read the element
ii) create a data structure.
iii) if empty queue then front of queue pointer points to newly created data structure.
iv) otherwise end of the queue points to newly created data structure
Step 5: If choice = remove
i) check if queue is empty, if so, print queue is empty.
ii) front of queue points next element.
iii) free element pointed
iv) by temp. pointer
v) return the element
vi) Print the element.

Step 6: If choice = display
i) check of empty queue if so print queue empty.
ii) otherwise print the elements from front of the queue until the end of the queue.

Step 7: If choice = exit
Stop.

Program :-

Output :-

Conclusion :-
Performed adding and removing from Queue.

Group E1:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:---

```
#include <iostream>
#define MAX 10
using namespace std;
struct queue
{      int data[MAX];
int front,rear;
};
class Queue
{    struct queue q;
  public:
     Queue(){q.front=q.rear=-1;}
     int isempty();
     int isfull();
     void enqueue(int);
     int delqueue();
     void display();
};
int Queue::isempty()
{
```

```
return(q.front==q.rear)?1:0;
}
int Queue::isfull()
{   return(q.rear==MAX-1)?1:0;}
void Queue::enqueue(int x)
{q.data[++q.rear]=x;}
int Queue::delqueue()
{return q.data[++q.front];}
void Queue::display()
{  int i;
   cout<<"\n";
   for(i=q.front+1;i<=q.rear;i++)
   cout<<q.data[i]<<" ";
}
int main()
{     Queue obj;
int ch,x;
do{   cout<<"\n 1. insert job\n 2.delete job\n 3.display\n 4.Exit\n Enter your
choice:";
     cin>>ch;
switch(ch)
{  case 1: if (!obj.isfull())
```

```
        {   cout<<"\n Enter data:";
                cin>>x;
                obj.enqueue(x);
        }
      else
           cout<< "Queue is overflow";
       break;
case 2: if(!obj.isempty())
                 cout<<"\n Deleted Element="<<obj.delqueue();
          else
                {   cout<<"\n Queue is underflow";  }
          cout<<"\nremaining jobs :";
          obj.display();
       break;
case 3: if (!obj.isempty())
    {  cout<<"\n Queue contains:";
           obj.display();
    }
    else
            cout<<"\n Queue is empty";
   break;
case 4: cout<<"\n Exit";
```

```
      }
    }while(ch!=4);
return 0;
}
```

Output:-

```
/************************************OUTPUT*****************************
 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:1
 Enter data:34

 1. insert job
 2.delete job
 3.display
 4.Exit
 Enter your choice:1
 Enter data:64

 1. insert job
 2.delete job
 3.display
```

4.Exit

Enter your choice:1

Enter data:84

1. insert job

2.delete job

3.display

4.Exit

Enter your choice:1

Enter data:93

1. insert job

2.delete job

3.display

4.Exit

Enter your choice:3

Queue contains:

34 64 84 93

1. insert job

2.delete job

3.display

4.Exit

Enter your choice:2

Deleted Element=34

remaining jobs :

64 84 93

1. insert job

2.delete job

3.display

4.Exit

Enter your choice:3

Queue contains:

64 84 93

1. insert job

2.delete job

3.display

4.Exit

Enter your choice:4

Exit*/

* Data Structure Lab (DSL):- Practical Number - 12 (Group - E)

Name:- Kaustubh Shrikant Kabra.
Class:- Second Year Engineering.
Div:- A Roll Number:-
Batch:-
Department:- Computer Department.
College:- AISSMS's IOIT.

Title:- To simulate deque with function to add and delete elements.

Aim:- Write a C++ program to simulate deque with function to add and delete elements.

Objective:-

- To study the concepts of deque.
- To understand the operation to insert and delete element from front and rear end.

Theory:-

In doubly-ended queue we can make use of both the ends for insertion of elements as well as use both ends for deletion of the elements.

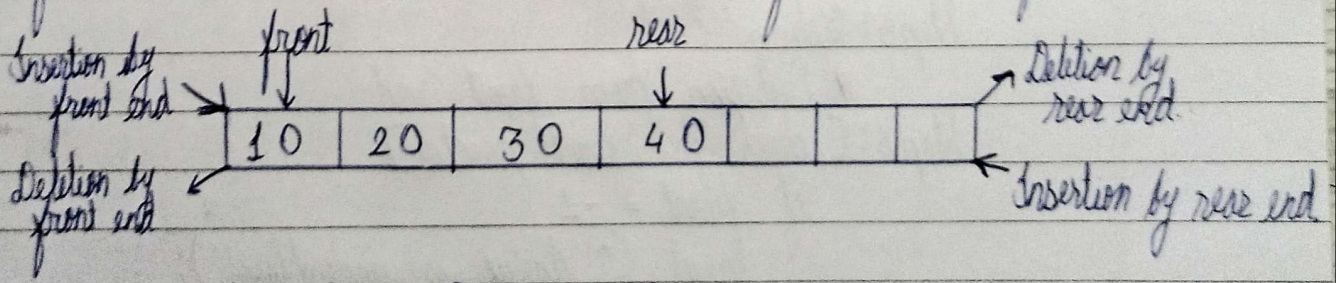


fig:- Doubly ended queue.

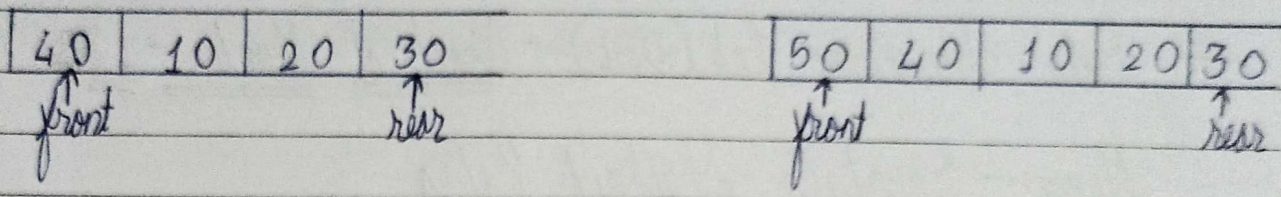


fig:- Insertion by front end.

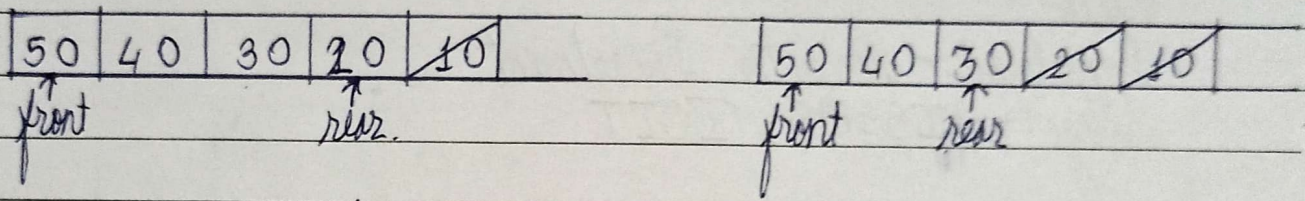


fig:- Deletion by rear end

Algorithm:-

Step 1: Start

Step 2: Initialize size of queue as Max = 50.

// Insertion at front end.

Step 3: [check for the front position]

if (front <= 1)

print ("cannot add item at front")

return;

Step 4: [Insert at front]

else:

front = front - 1;

q[front] = no.;

Step 5: Return

// Deletion from front end.

Step 6: [check for front pointer]

if front = -1

print ("queue is underflow");

return;

Step 7: [Perform deletion]

```
else
    no = q[front]
print("Deletion element is", no);
if front = rear
    front = 0;
    rear = 0;
else:
    front = front + 1;
```

Step 8: [check for overflow]

```
if (rear == Max]
print("Queue is overflow");
```

Step 9:- [Insert element]

```
else:
    rear = rear + 1;
    q[rear] = no;
[Set rear and front pointer]
if rear = 0
    rear = 1;
if front = 0
    front = 1;
```

Step 10: [check for the rear pointer]

```
if rear = 0
    print("Cannot delete value at rear");
    return;
```

Step 11: [check for and perform deletion]

```
else:
    no = q[rear];
if front = rear
    front = rear = 0;
```

else rear = rear - 1;
print ("Deleted element is", no.);
Return.

Program:-

Output:-

Conclusion:-
Performed enque and deque operation on double-ended Queue.

Group E31:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:---

```
#include<iostream>
//#include
//#include
using namespace std;
#define SIZE 5
// ERROR HANDLINH NOT DOne

//    program is not working correct.
//
class dequeue
{

        int a[10],front,rear,count;

public:
        dequeue();
        void add_at_beg(int);
        void add_at_end(int);
        void delete_fr_front();
        void delete_fr_rear();
        void display();
};


dequeue::dequeue()
{
        front=-1;
        rear=-1;
        count=0;
}
```

```
void dequeue::add_at_beg(int item)
{
	int  i;
	if(front==-1)
	{
		front++;
		rear++;
		a[rear]=item;
		count++;
	}
	else if(rear>=SIZE-1)
	{
		cout<<"\nInsertion is not possible,overflow!!!!";
	}
	else
	{
		for(i=count;i>=0;i--)
		{
			a[i]=a[i-1];
		}
		a[i]=item;
		count++;
		rear++;
	}
}



void dequeue::add_at_end(int item)
{

	if(front==-1)
	{
		front++;
```

```
            rear++;
            a[rear]=item;
            count++;
        }
        else if(rear>=SIZE-1)
        {
            cout<<"\nInsertion is not possible,overflow!!!";
            return;
        }
        else
        {
            a[++rear]=item;
        }


}



void dequeue::display()
{

        for(int i=front;i<=rear;i++)
        {
            cout<<a[i]<<" ";     }
}


void dequeue::delete_fr_front()
{
        if(front==-1)
        {
            cout<<"Deletion is not possible:: Dequeue is empty";
            return;
        }
```

```
        else
        {
                if(front==rear)
                {
                        front=rear=-1;
                        return;
                }
                cout<<"The deleted element is "<<a[front];
                front=front+1;
        }


}

void dequeue::delete_fr_rear()
{
        if(front==-1)
        {
                cout<<"Deletion is not possible:Dequeue is empty";
                return;
        }
        else
        {
                if(front==rear)
                {
                        front=rear=-1;
                }
                cout<<"The deleted element is "<< a[rear];
                rear=rear-1;
        }


}
```

```
int main()
{
	int c,item;
	dequeue d1;

	do
	{
		cout<<"\n\n****DEQUEUE OPERATION****\n";
		cout<<"\n1-Insert at beginning";
		cout<<"\n2-Insert at end";
		cout<<"\n3_Display";
		cout<<"\n4_Deletion from front";
		cout<<"\n5-Deletion from rear";
		cout<<"\n6_Exit";
		cout<<"\nEnter your choice<1-4>:";
		cin>>c;

		switch(c)
		{
		case 1:
			cout<<"Enter the element to be inserted:";
			cin>>item;
			d1.add_at_beg(item);
			break;

		case 2:
			cout<<"Enter the element to be inserted:";
			cin>>item;
			d1.add_at_end(item);
			break;

		case 3:
			d1.display();
			break;
```

```
            case 4:
                d1.delete_fr_front();
                break;
            case 5:
                d1.delete_fr_rear();
                break;

            case 6:
                exit(1);
                break;

            default:
                cout<<"Invalid choice";
                break;
            }

        }while(c!=7);
        return 0;

}
```

Output:-

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:1

Enter the element to be inserted:25

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:1

Enter the element to be inserted:26

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:2

Enter the element to be inserted:65

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:3

0 25 65

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:4

The deleted element is 0

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:3

25 65

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:5

The deleted element is 65

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:3

25

****DEQUEUE OPERATION****

1-Insert at beginning

2-Insert at end

3_Display

4_Deletion from front

5-Deletion from rear

6_Exit

Enter your choice<1-4>:6

* Data Structure Lab (DSL):- Practical Number - 12 (Group - E)

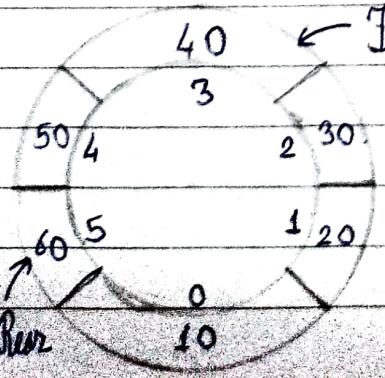
Name:- Kaustubh Shrikant Kabra
Class:- Second Year Engineering
Div:- A Roll Number:-
Batch:-
Department:- Computer Department
College:- AISSMS's IOIT.

Title:- To simulate Pizza order system using circular queue using array.

Aim:- Write a C++ program to simulate system using circular queue using array.

Objective:-
- To study the concepts of circular queue.
- To perform operations on circular queue using arrays.

Theory:-
A circular queue is a linear data structure in which the operations are performed based on FIFO principal and the last position is connected back to the first position to make a circle. It is also called 'Ring Buffer'.



formula for setting the front and rear pointer for circular queue.

rear = (rear + 1) % size.
front = (front + 1) % size.

Algorithm:-

Step 1: Start

Step 2: Intialize class Pizza
initialize max, f, r;

Step 3: public class pizza()
f = -1 , r = -1
print ("Enter Maximum order");

Step 4: full()
if (((f == 0) && (r == (max - 1))) || (f == (r + 1) % max
return 1;
else
return 0

Step 5:- empty()
if (f = -1)
return 1;
else
return 0;
add (int s)

Step 6: ^ check if (full())
print ("Order is full");
elseif (f == -1)
f = r = 0
else
r = (r + 1) % max
order [r] = s.

Step 7: remove ()
int i;
i = order [f]
if (f == r)
f = r = -1;

```
else
    f = (f+1) % max;
print ("Order Deleted")
```

Step 8: display()

```
int temp
    temp = f
if (empty())
    print ("No more currently");
else
    print ("The order are:")
        print (order [temp])
        temp = (temp+1) % max
```

Step 9: main()

```
int ch;
```

Step 10: Create a menu drive code using do while to operate on above function

Return 0.

Program:-

Output:-

Conclusion:-

Thus, the Pizza order system simulated using circular queue array operation.

Group E32:--- Kaustubh Shrikant Kabra SE COMP-1 20

Program:---

```
#include<iostream>

using namespace std;

class pizza{
int order[10];
int max;
int f,r;

public :
pizza(){
f=-1,r=-1;

cout<<"\nEnter Maximum order : ";
cin>>max;
}

int full(){
  if(((f==0)&&(r==(max-1)))||(f==(r+1)%max))
```

```
        return 1;
    else
        return 0;
}
int qempty(){
       if(f==-1)
          return 1;
       else
          return 0;
   }

void add(int a){
if(full()){
cout<<"\nOrder is Full!!!";
}
else{
if(f==-1){
f=r=0;
}
else{
r=(r+1)%max;
}
```

```
order[r]=a;
}
}

void remove(){
int i;
i=order[f];
     if(f==r){
f=r=-1;
}
else{
f=(f+1)%max;
}
cout<<"\n Order deleted : "<<i;



}

void display(){
int temp;
temp=f;
if(qempty())
```

```
   {
      cout<<"\nNo orders currently\n";

   }
else{
cout<<"\nThe oders are : \n\n";
while(temp!=r){
cout<<"  "<<order[temp];
temp=(temp+1)%max;
}
cout<<"  "<<order[temp];
}}

};

int main(){

int ch;
pizza p;
do{
cout<<"\n1. Order \n2. Remove order \n3.Display orders \n4. Exit";
cin>>ch;
```

```
switch(ch){
case 1:int o;
    cout<<"\nEnter Order number : ";
    cin>>o;
    p.add(o);
    break;
case 2:p.remove();
    break;
case 3:p.display();
    break;

}

}while(ch!=4);
return 0;

}
```

Output:-

```
Enter Maximum order : 3
```

1. Order

2. Remove order

3.Display orders

4. Exit1

Enter Order number : 123

1. Order

2. Remove order

3.Display orders

4. Exit2

Order deleted : 123

1. Order

2. Remove order

3.Display orders

4. Exit3

No orders currently

1. Order

2. Remove order

3.Display orders

4. Exit4