

Practical No: 01 To perform following operations on the matrix.

a) Addition of two matrices b) Subtraction of two matrices

c) Multiplication of two matrices d) Transpose of a matrix

- **Program:**

```
def accept_marks(A):
    n = int(input("Enter the total no. of student : "))
    for i in range(n) :
        while True :
            x = input("Enter the marks scored in FDS for student %d : "%(i+1))
            if(x == "AB"):
                x = -1 # indicates Absent students
                break
            x = int(x)
            if(x >= 0 and x <= 30) :
                break
            else :
                print("Plz enter valid marks out of 30")
        A.append(x)
    print("Marks accepted & stored successfully");

def display_marks(A) :
    print("\nMarks Scored in FDS")
    for i in range(len(A)):
        if(A[i] == -1) :
            print("\tStudent %d : AB"%(i+1))
        else :
            print("\tStudent %d : %d"%(i+1,A[i]))

def search_set(A,X) :
    n = len(A)
    for i in range(n):
        if(A[i] == X) :
            return (1)
```

```
return (0)
```

```
def find_average_score_of_class(A) :  
    sum = 0  
    for i in range(len(A)) :  
        if(A[i] != -1) :  
            sum = sum + A[i]  
    avg = sum / len(A)  
    display_marks(A)  
    print("\nAverage score of class is %.2f\n\n"%avg)
```

```
def find_highest_and_lowest_score_of_class(A) :  
    max = -1  
    min = 31  
    for i in range(1,len(A)) :  
        if(max < A[i]) :  
            max = A[i]  
            max_ind = i  
        if(min > A[i] and A[i] != -1) :  
            min = A[i]  
            min_ind = i  
    display_marks(A)  
    print("Highest Mark Score of class is %d scored by student %d"%(max,max_ind+1))  
    print("Lowest Mark Score of class is %d scored by student %d"%(min,min_ind+1))
```

```
def find_count_of_absent_students(A) :  
    count = 0  
    for i in range(len(A)):  
        if(A[i] == -1) :  
            count += 1  
    display_marks(A)  
    print("\tAbsent Student Count = %d"%count)
```

```
def display_mark_with_highest_frequency(A) :  
    freq = 0  
    for i in range(len(A)) :  
        count = 0  
        if(A[i] != -1) :  
            for j in range(len(A)):  
                if(A[i] == A[j]) :  
                    count += 1  
            if(freq < count) :  
                Marks = A[i]  
                freq = count  
    display_marks(A)  
    print("\nMarks with highest frequency is %d (%d)"%(Marks,freq))
```

```
def main():  
    FDS_Marks = []  
    while True :  
        print ("\t\t1 : Accept FDS Marks")  
        print ("\t\t2 : Average score of class")  
        print ("\t\t3 : Highest score and lowest score of class")  
        print ("\t\t4 : Count of students who were absent for the test")  
        print ("\t\t5 : Display mark with highest frequency")  
        print ("\t\t6 : Exit")  
        ch = int(input("Enter your choice : "))  
        if (ch == 6):  
            print ("End of Program")  
            quit()  
        elif (ch == 1) :  
            accept_marks(FDS_Marks)  
            display_marks(FDS_Marks)  
        elif (ch == 2) :  
            find_average_score_of_class(FDS_Marks)  
        elif (ch == 3) :
```

```
        find_highest_and_lowest_score_of_class(FDS_Marks)
    elif (ch == 4) :
        find_count_of_absent_students(FDS_Marks)
    elif (ch == 5) :
        display_mark_with_highest_frequency(FDS_Marks)
    else :
        print ("Wrong choice entered !! Try again")

main()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:~$ python pr1.python
```

```
jaihind@jaihind-ThinkCentre-M60e:~$ ./a.out
```

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 1

Enter the total no. of student : 2

Enter the marks scored in FDS for student 1 : 70

Plz enter valid marks out of 30

Enter the marks scored in FDS for student 1 : 29

Enter the marks scored in FDS for student 2 : 25

Marks accepted & stored successfully

Marks Scored in FDS

Student 1 : 29

Student 2 : 25

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

Roll No. 65

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 2

Marks Scored in FDS

Student 1 : 29

Student 2 : 25

Average score of class is 27.00

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 3

Marks Scored in FDS

Student 1 : 29

Student 2 : 25

Highest Mark Score of class is 25 scored by student 2

Lowest Mark Score of class is 25 scored by student 2

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 4

Marks Scored in FDS

Student 1 : 29

Student 2 : 25

Absent Student Count = 0

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 5

Marks Scored in FDS

Student 1 : 29

Student 2 : 25

Marks with highest frequency is 29 (1)

1 : Accept FDS Marks

2 : Average score of class

3 : Highest score and lowest score of class

4 : Count of students who were absent for the test

5 : Display mark with highest frequency

6 : Exit

Enter your choice : 6

End of Program

Practical No: 02 Write Python program to compute following operation on string:

- a) To display word with the longest length.**
- b) To determine the frequency of occurrence of particular character in the string.**
- c) To check whether given string is palindrome or not.**
- d) To display index of first appearance of the substring.**
- e) To count the occurrences of each word in given string.**

- **Program:**

```
def Display_word_with_longest_length() :  
    Str = input("Enter the main string : ") # Its assumed that string contain only characters  
    and spaces (multile spaces are allowed)  
    M_str = ""  
    i = 0  
    while( i < len(Str)) :  
        word = ""  
        while(Str[i] != ' '):  
            word += Str[i]  
            i = i + 1  
        if( i == len(Str)) :  
            break  
        if(i != len(Str)) :  
            while(Str[i] == ' '):  
                i = i + 1  
            if(len(M_str) < len(word)) :  
                M_str = word  
    print("\tWord with longest length is %s having lenght %d\n\n"%(M_str,len(M_str)))
```

```
def Determine_frequency_of_occurrence_of_particular_character_in_string() :  
    Str = input("Enter the string : ")  
    C = input("Enter the character : ")  
    print("\tString : %s"%Str)
```

```
print("\tCharacter : %s"%C)
count = 0
for i in range(len(Str)) :
    if(Str[i] == C) :
        count += 1
print("\tFrequency of occurrence of character(%s) in string(%s) is
%d\n\n"%(C,Str,count))
```

```
def Check_for_palindrome() :
    Str = input("Enter the string to be checked : ")
    b = 0
    e = len(Str) - 1
    while( b < e) :
        if(Str[b] != Str[e]) :
            break
        b += 1
        e -= 1
    if(b < e) :
        print("\t%s string is not an palindrome string\n\n"%Str)
    else :
        print("\t%s string is an palindrome string\n\n"%Str)
```

```
def display_index_of_first_appearance_of_the_substring() :
    M = input("Enter the main string : ")
    S = input("Enter the sub string to check : ")
    print("Main String : %s"%M)
    print("Substring String : %s"%S)
    L1 = len(M)
    L2 = len(S)
    if(L1 >= L2) :
        for i in range((L1 - L2 + 1)) :
            flag = 1
```



```
for j in range(L2):
    if(M[i+j] != S[j]) :
        flag = 0
        break
    if(flag == 1) :
        print("Substring %s found at index %d\n\n"%(S,i))
        break;
    if(flag == 0) :
        print("Substring not found in the main string\n\n")
else :
    print("Substring is greater than main string\n\n")
```

```
def Count__occurrences_of_each_word_in_given_string() :
    Str = input("Enter the main string : ") # Its assumed that string contain only characters
and spaces (multile spaces are allowed)
    i = 0
    Word_array = []
    Count = []
    while( i < len(Str)) :
        word = ""
        while(Str[i] != ' ') :
            word += Str[i]
            i = i + 1
        if( i == len(Str)) :
            break
        if(i != len(Str)) :
            while(Str[i] == ' ') :
                i = i + 1
        if(len(Word_array) == 0) :
            Word_array.append(word)
            Count.append(1)
    else :
```

```
flag = 1
for j in range(len(Word_array)) :
    if(Word_array[j] == word) :
        Count[j] += 1
        flag = 0
        break
if (flag == 1) :
    Word_array.append(word)
    Count.append(1)
for i in range(len(Word_array)) :
    print("\t%15s : %d"%(Word_array[i],Count[i]))
```

```
def main():
    while True :
        print ("\t\t ***** STRING OPERATIONS *****")
        print ("\t\t1 : Display word with longest length")
        print ("\t\t2 : Determine the frequency of occurrence of particular character in the
string")
        print ("\t\t3 : Check whether given string is palindrome or not ")
        print ("\t\t4 : Display index of first appearance of the substring")
        print ("\t\t5 : Count the occurrences of each word in a given string")
        print ("\t\t6 : Exit")
        ch = int(input("Enter your choice : "))
        if (ch == 6):
            print ("End of Program")
            quit()
        elif (ch == 1) :
            Display_word_with_longest_length()
        elif (ch == 2) :
            Determine_frequency_of_occurrence_of_particular_character_in_string()
        elif (ch == 3) :
            Check_for_palindrome()
        elif (ch == 4) :
```

```
        display_index_of_first_appearance_of_the_substring()
    elif (ch == 5) :
        Count__occurrences_of_each_word_in_given_string()
    else :
        print ("Wrong choice entered !! Try again")

main()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:$ python pr2.python
```

```
jaihind@jaihind-ThinkCentre-M60e:$ ./a.out
```

```
**** STRING OPERATIONS ****
```

- 1 : Display word with longest length
- 2 : Determine the frequency of occurrence of particular character in the string
- 3 : Check whether given string is palindrome or not
- 4 : Display index of first appearance of the substring
- 5 : Count the occurrences of each word in a given string
- 6 : Exit

Enter your choice : 1

Enter the main string : python

Word with longest length is python having lenght 6

```
**** STRING OPERATIONS ****
```

- 1 : Display word with longest length
- 2 : Determine the frequency of occurrence of particular character in the string
- 3 : Check whether given string is palindrome or not
- 4 : Display index of first appearance of the substring
- 5 : Count the occurrences of each word in a given string
- 6 : Exit

Enter your choice : 2

Enter the string : python

Enter the character : t

String : python

Character : t

Frequency of occurrence of character(t) in string(python) is 1

**** STRING OPERATIONS ****

1 : Display word with longest length

2 : Determine the frequency of occurrence of particular character in the string

3 : Check whether given string is palindrome or not

4 : Display index of first appearance of the substring

5 : Count the occurrences of each word in a given string

6 : Exit

Enter your choice : 3

Enter the string to be checked : aba

aba string is an palindrome string

**** STRING OPERATIONS ****

1 : Display word with longest length

2 : Determine the frequency of occurrence of particular character in the string

3 : Check whether given string is palindrome or not

4 : Display index of first appearance of the substring

5 : Count the occurrences of each word in a given string

6 : Exit

Enter your choice : 4

Enter the main string : python

Enter the sub string to check : on

Main String : python

Substring String : on

Substring on found at index 4

**** STRING OPERATIONS ****

1 : Display word with longest length

2 : Determine the frequency of occurrence of particular character in the string

- 3 : Check whether given string is palindrome or not
- 4 : Display index of first appearance of the substring
- 5 : Count the occurrences of each word in a given string
- 6 : Exit

Enter your choice : 5

Enter the main string : python

python : 1

**** STRING OPERATIONS ****

- 1 : Display word with longest length
- 2 : Determine the frequency of occurrence of particular character in the string
- 3 : Check whether given string is palindrome or not
- 4 : Display index of first appearance of the substring
- 5 : Count the occurrences of each word in a given string
- 6 : Exit

Enter your choice : 6

End of Program

Practical No: 03 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 students who play cricket and football but not badminton.**

- **Program:**

```
def accept_matrix(M) :  
    print("\nEnter the order of the Matrix (row,col) : ")  
    r = int(input("\trow = "))  
    c = int(input("\tcol = "))  
    print("Enter the elements of the Matrix : \n")  
    for i in range(r) :  
        A = []  
        for j in range(c) :  
            A.append(int(input()))  
        M.append(A)  
    print("\nMatrix accepted successfully\n")
```

```
def display_matrix(M,r,c):  
    print("Matrix (%d,%d) : "%(r,c))  
    for i in range(r) :  
        print("\t\t",end=' ')  
        for j in range(c):  
            print("%3d"%M[i][j],end=' ')  
        print("")
```

```
def addition_matrix(M1,M2,M3,r,c) :  
    for i in range(r) :  
        A = []  
        for j in range(c):  
            A.append(M1[i][j] + M2[i][j])  
        M3.append(A)  
  
def subtraction_matrix(M1,M2,M3,r,c) :  
    for i in range(r) :  
        A = []  
        for j in range(c):  
            A.append(M1[i][j] - M2[i][j])  
        M3.append(A)  
  
def multiplication_matrix(M1,M2,M3,r1,c1,c2) :  
    for i in range(r1) :  
        A = []  
        for j in range(c2) :  
            sum = 0  
            for k in range(c1) :  
                sum = sum + (M1[i][k] * M2[k][j])  
            A.append(sum)  
        M3.append(A)  
  
def find_transpose_matrix(M,r,c,T) :  
    for i in range(c):  
        A = []  
        for j in range(r):  
            A.append(M[j][i])  
        T.append(A)  
  
def main():  
    while True :
```

```
print("\t\t\t1: Accept Matrix");
print("\t\t\t2: Display Matrix");
print("\t\t\t3: Addition of Matrices");
print("\t\t\t4: Substraction of Matrices");
print("\t\t\t5: Multiplication of Matrices");
print("\t\t\t6: Transpose Matrix");
print("\t\t\t7: Exit");
```

```
ch = int(input("Enter your choice : "))
```

```
M3 = []
```

```
if (ch == 7):
```

```
    print ("End of Program")
```

```
    break
```

```
elif (ch==1):
```

```
    M1 = []
```

```
    M2 = []
```

```
    print("Input First Matrix ")
```

```
    accept_matrix(M1)
```

```
    r1 = len(M1)
```

```
    c1 = len(M1[0])
```

```
    print("Input Second Matrix ")
```

```
    accept_matrix(M2)
```

```
    r2 = len(M2)
```

```
    c2 = len(M2[0])
```

```
elif (ch==2):
```

```
    print("\tFirst ",end=' ')
```

```
    display_matrix(M1,r1,c1)
```

```
    print("\tSecond ",end=' ')
```

```
    display_matrix(M2,r2,c2)
```

```
elif (ch==3):
```

```
    print("\tFirst ",end=' ')
```

```
    display_matrix(M1,r1,c1)
```

```
    print("\tSecond ",end=' ')
```

```
    display_matrix(M2,r2,c2)
```



```
if(r1 == r2 and c1 == c2) :
    addition_matrix(M1,M2,M3,r1,c1)
    print("\tAddition ")
    display_matrix(M3,r1,c1)
else :
    print("Addition not possible (order not same)")

elif (ch==4):
    print("\tFirst ",end=' ')
    display_matrix(M1,r1,c1)
    print("\tSecond ",end=' ')
    display_matrix(M2,r2,c2)
    if(r1 == r2 and c1 == c2) :
        subtraction_matrix(M1,M2,M3,r1,c1)
        print("\tSubtraction ")
        display_matrix(M3,r1,c1)
    else :
        print("subtraction not possible (order not same)")

elif (ch==5):
    print("\tFirst ",end=' ')
    display_matrix(M1,r1,c1)
    print("\tSecond ",end=' ')
    display_matrix(M2,r2,c2)
    if(c1 == r2) :
        multiplication_matrix(M1,M2,M3,r1,c1,c2)
        print("\tMultiplication ")
        display_matrix(M3,r1,c2)
    else :
        print("Multiplication not possible ")
elif (ch==6):
    print("\tFirst ",end=' ')
    display_matrix(M1,r1,c1)
    find_transpose_matrix(M1,r1,c1,M3);
```

```
print("\tTranspose ",end=' ');
display_matrix(M3,c1,r1)
print("\tSecond ",end=' ')
display_matrix(M2,r2,c2)
M3 = []
find_transpose_matrix(M2,r2,c2,M3);
print("\tTranspose ",end=' ');
display_matrix(M3,c2,r2)

else :
    print ("Wrong choice entered !! Try again")

main()
quit()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:$ python pr3.python
jaihind@jaihind-ThinkCentre-M60e:$ ./a.out
```

- 1: Accept Matrix
- 2: Display Matrix
- 3: Addition of Matrices
- 4: Substraction of Matrices
- 5: Multiplication of Matrices
- 6: Transpose Matrix
- 7: Exit

Enter your choice : 1

Input First Matrix

Enter the order of the Matrix (row,col) :

row = 1

col = 1

Enter the elements of the Matrix :

Matrix accepted successfully

Input Second Matrix

Enter the order of the Matrix (row,col) :

row = 1

col = 1

Enter the elements of the Matrix :

30

Matrix accepted successfully

- 1: Accept Matrix
- 2: Display Matrix
- 3: Addition of Matrices
- 4: Substraction of Matrices
- 5: Multiplication of Matrices
- 6: Transpose Matrix
- 7: Exit

Enter your choice : 2

First Matrix (1,1) :

20

Second Matrix (1,1) :

30

- 1: Accept Matrix
- 2: Display Matrix
- 3: Addition of Matrices
- 4: Substraction of Matrices
- 5: Multiplication of Matrices
- 6: Transpose Matrix
- 7: Exit

Enter your choice : 3

First Matrix (1,1) :

20

Second Matrix (1,1) :

30

Addition

Matrix (1,1) :

50

1: Accept Matrix

2: Display Matrix

3: Addition of Matrices

4: Substraction of Matrices

5: Multiplication of Matrices

6: Transpose Matrix

7: Exit

Enter your choice : 4

First Matrix (1,1) :

20

Second Matrix (1,1) :

30

Substraction

Matrix (1,1) :

-10

1: Accept Matrix

2: Display Matrix

3: Addition of Matrices

4: Substraction of Matrices

5: Multiplication of Matrices

6: Transpose Matrix

7: Exit

Enter your choice : 5

First Matrix (1,1) :

20

Second Matrix (1,1) :

30

Multiplication

Matrix (1,1) :

600

- 1: Accept Matrix
- 2: Display Matrix
- 3: Addition of Matrices
- 4: Substraction of Matrices
- 5: Multiplication of Matrices
- 6: Transpose Matrix
- 7: Exit

Enter your choice : 6

First Matrix (1,1) :

20

Transpose Matrix (1,1) :

20

Second Matrix (1,1) :

30

Transpose Matrix (1,1) :

30

- 1: Accept Matrix
- 2: Display Matrix
- 3: Addition of Matrices
- 4: Substraction of Matrices
- 5: Multiplication of Matrices
- 6: Transpose Matrix
- 7: Exit

Enter your choice : 7

End of Program

Practical No: 04 a. Write a Python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search.

b. Write a Python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended

training program or not, using Binary search and Fibonacci search

- **Program a:**

```
def accept_array(A):  
    n = int(input("Enter the total no. of student : "))  
    for i in range(n):  
        x = int(input("Enter the roll no of student %d :"%(i+1)))  
        A.append(x)  
    print("Student Info accepted successfully\n\n")  
    return n
```

```
def display_array(A,n):  
    if(n == 0) :  
        print("\nNo records in the database")  
    else :  
        print("Students Array : ",end=' ' )  
        for i in range(n) :  
            print("%d "%A[i],end=' ' )  
        print("\n");
```

```
def Linear_Search(A,n,X) :  
    for i in range(n) :  
        if(A[i] == X) :  
            return i    # found so returning the position i.e index
```

```
return -1    # Not found
```

```
def Sentinel_Search(A,n,X) :  
    last = A[n-1]  
    i = 0  
    A[n-1] = X    # Here X is the roll_no to be searched.  
    while(A[i] != X) :  
        i = i + 1  
    A[n-1] = last  
    if( (i < n-1) or (X == A[n-1]) ) :  
        return i    #roll_no found at location i  
    else :  
        return -1    # roll_no not found"
```

```
def Main() :  
    A = []  
    while True :  
        print ("\t1 : Accept & Display Students info ")  
        print ("\t2 : Linear Search")  
        print ("\t3 : Sentinel Search")  
        print ("\t4 : Exit")  
        ch = int(input("Enter your choice : "))  
        if (ch == 4):  
            print ("End of Program")  
            quit()  
        elif (ch==1):  
            A = []  
            n = accept_array(A)  
            display_array(A,n)  
        elif (ch==2):  
            X = int(input("Enter the roll_no to be searched : "))  
            flag = Linear_Search(A,n,X)
```

```
    if(flag == -1) :
        print("\tRoll no to be Searched not Found\n")
    else :
        print("\tRoll no found at location %d"%(flag + 1))
elif (ch==3):
    X = int(input("Enter the roll_no to be searched : "))
    flag = Sentinel_Search(A,n,X)
    if(flag == -1) :
        print("\tRoll no to be Searched not Found\n")
    else :
        print("\tRoll no found at location %d"%(flag + 1))
else :
    print ("Wrong choice entered !! Try again")
Main()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:$ python pr4a.python
```

```
jaihind@jaihind-ThinkCentre-M60e:$ ./a.out
```

```
1 : Accept & Display Students info
```

```
2 : Linear Search
```

```
3 : Sentinel Search
```

```
4 : Exit
```

```
Enter your choice : 1
```

```
Enter the total no. of student : 2
```

```
Enter the roll no of student 1 : 1
```

```
Enter the roll no of student 2 : 2
```

```
Student Info accepted successfully
```

```
Students Array : 1 2
```

```
1 : Accept & Display Students info
```

```
2 : Linear Search
```

```
3 : Sentinel Search
```


4 : Exit

Enter your choice : 2

Enter the roll_no to be searched : 2

Roll no found at location 2

1 : Accept & Display Students info

2 : Linear Search

3 : Sentinel Search

4 : Exit

Enter your choice : 3

Enter the roll_no to be searched : 1

Roll no found at location 1

1 : Accept & Display Students info

2 : Linear Search

3 : Sentinel Search

4 : Exit

Enter your choice : 4

End of Program

- **Program b:**

```
def accept_array(A):
```

```
    n = int(input("Enter the total no. of student : "))
```

```
    print("Input roll numbers in sorted order")
```

```
    for i in range(n):
```

```
        x = int(input("Enter the roll no of student %d :"%(i+1)))
```

```
        A.append(x)
```

```
    print("Student Info accepted successfully\n\n")
```

```
    return n
```

```
def display_array(A,n):
```

```
    if(n == 0) :
```

```
        print("\nNo records in the database")
```

```
    else :
```

```
        print("Students Array : ",end=' ')
```

```
        for i in range(n) :
```

```
print("%d "%A[i],end=' ')
print("\n");
```

```
def Recursive_Binary_Search(A,s,l,X) :
    if(s <= l ) :
        mid = int((s + l) / 2)
        if(A[mid] == X) :
            return mid    # Found
        else :
            if(X < A[mid] ) :
                return Recursive_Binary_Search(A,s,mid-1,X)
            else :
                return Recursive_Binary_Search(A,mid+1,l,X)
    return -1 # NOT FOUND
```

```
def Iterative_Binary_Search(A,n,X) :
    s = 0
    l = n-1
    while(s <= l ) :
        mid = int((s + l) / 2)
        if(A[mid] == X) :
            return mid    # Found
        else :
            if (X < A[mid] ) :
                l = mid-1
            else :
                s = mid+1
    return -1; #NOT FOUND
```

#Returns index of x if present, else returns -1

```
def Fibonacci_Search(A,n,X) :  
    f1 = 0  
    f2 = 1  
    f3 = f1 + f2  
    offset = -1  
    while (f3 < n) :  
        f1 = f2  
        f2 = f3  
        f3 = f1 + f2  
    while (f3 > 1) :  
        i = min(offset+f1, n-1)  
        if(A[i] == X) :  
            return i      #Found  
        else :  
            if (X < A[i] ) : # left substudent (66 % or 2/3 student)  
                f3 = f1  
                f2 = f2 - f1  
                f1 = f3 - f2  
            else : # right substudent ( 33 % or 1/3 student)  
                f3 = f2  
                f2 = f1  
                f1 = f3 - f2  
            offset = i  
    if(f2 == 1 and (offset+1) < n and A[offset + 1] == X) :  
        return offset+1    # Found  
    return -1    #NOT FOUND
```

```
def Main() :  
    A = []  
    while True :  
        print ("\t1 : Accept & Display Students info ")  
        print ("\t2 : Recursive Binary Search")  
        print ("\t3 : Iterative Binary Search")
```

```
print ("\t4 : Fibonacci Search")
print ("\t5 : Exit")
ch = int(input("Enter your choice : "))
if (ch == 5):
    print ("End of Program")
    quit()
elif (ch==1):
    A = []
    n = accept_array(A)
    display_array(A,n)
elif (ch==2):
    X = int(input("Enter the roll_no to be searched : "))
    flag = Recursive_Binary_Search(A,0,n-1,X)
    if(flag == -1) :
        print("\tRoll no to be Searched not Found\n")
    else :
        print("\tRoll no found at location %d"%(flag + 1))
elif (ch==3):
    X = int(input("Enter the roll_no to be searched : "))
    flag = Iterative_Binary_Search(A,n,X)
    if(flag == -1) :
        print("\tRoll no to be Searched not Found\n")
elif (ch==4):
    X = int(input("Enter the roll_no to be searched : "))
    flag = Fibonacci_Search(A,n,X)
    if(flag == -1) :
        print("\tRoll no to be Searched not Found\n")
    else :
        print("\tRoll no found at location %d"%(flag + 1))
else :
    print ("Wrong choice entered !! Try again")
Main()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:$ python pr4b.python
```

```
jaihind@jaihind-ThinkCentre-M60e:$ ./a.out
```

```
1 : Accept & Display Students info
```

```
2 : Recursive Binary Search
```

```
3 : Iterative Binary Search
```

```
4 : Fibonacci Search
```

```
5 : Exit
```

```
Enter your choice : 1
```

```
Enter the total no. of student : 2
```

```
Input roll numbers in sorted order
```

```
Enter the roll no of student 1 : 1
```

```
Enter the roll no of student 2 : 2
```

```
Student Info accepted successfully
```

```
Students Array : 1 2
```

```
1 : Accept & Display Students info
```

```
2 : Recursive Binary Search
```

```
3 : Iterative Binary Search
```

```
4 : Fibonacci Search
```

```
5 : Exit
```

```
Enter your choice : 2
```

```
Enter the roll_no to be searched : 2
```

```
Roll no found at location 2
```

```
1 : Accept & Display Students info
```

```
2 : Recursive Binary Search
```

```
3 : Iterative Binary Search
```

```
4 : Fibonacci Search
```

```
5 : Exit
```

```
Enter your choice : 3
```

```
Enter the roll_no to be searched : 1
```

```
1 : Accept & Display Students info
```

Roll No. 65

2 : Recursive Binary Search

3 : Iterative Binary Search

4 : Fibonacci Search

5 : Exit

Enter your choice : 4

Enter the roll_no to be searched : 2

Roll no found at location 2

1 : Accept & Display Students info

2 : Recursive Binary Search

3 : Iterative Binary Search

4 : Fibonacci Search

5 : Exit

Enter your choice : 5

End of Program

Practical No: 05 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.

- **Program:**

```
def accept_array(A):
    n = int(input("Enter the total no. of student : "))
    for i in range(n):
        x = float(input("Enter the first year percentage of student %d :"%(i+1)))
        A.append(x)
    print("Array accepted successfully\n\n");
```

```
def display_array(A):
    n = len(A)
    if(n == 0) :
        print("\nNo records in the database")
    else :
        print("Array of FE Marks : ",end=' ')
        for i in range(n) :
            print("%.2f "%A[i],end=' ')
        print("\n");
```

```
def partition(A,s,l) :
    b=s+1
    e=l
    while(e>=b) :
        while(b<=l and A[s] >= A[b]) :
            b = b + 1
        while(A[s] <A[e]) :
            e = e - 1
        if(e>b) :
```

```
    temp = A[e]
    A[e] = A[b]
    A[b] = temp
temp = A[s]
A[s] = A[e]
A[e] = temp
return e
```

```
def Quicksort(A,s,l) :
    if(s<l) :
        mid = partition(A,s,l)
        Quicksort(A,s,mid-1)
        Quicksort(A,mid+1,l)
```

```
def Main() :
    A = []
    while True :
        print ("\t1 : Accept & Display the FE Marks")
        print ("\t2 : Quick sort ascending order and display top five scores")
        print ("\t3 : Exit")
        ch = int(input("Enter your choice : "))
        if (ch == 3):
            print ("End of Program")
            quit()
        elif (ch==1):
            A = []
            accept_array(A)
            display_array(A)
        elif (ch==2):
            print("Marks before sorting")
            display_array(A)
            n =len(A)
```



```
Quicksort(A,0,n-1)
print("Marks after sorting")
display_array(A)
if(n >= 5) :
    print("Top Five Scores : ")
    for i in range(n-1,n-6,-1) :
        print("\t%.2f"%A[i])
    else :
        print("Top Scorers : ")
        for i in range(n-1,-1,-1) :
            print("\t%.2f"%A[i])
    else :
        print ("Wrong choice entered !! Try again")
Main()
```

- **Output:**

```
jaihind@jaihind-ThinkCentre-M60e:$ python pr5.python
```

```
jaihind@jaihind-ThinkCentre-M60e:$ ./a.out
```

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 1

Enter the total no. of student : 2

Enter the first year percentage of student 1 : 75

Enter the first year percentage of student 2 : 80

Array accepted successfully

Array of FE Marks : 75.00 80.00

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 2

Roll No. 65

Marks before sorting

Array of FE Marks : 75.00 80.00

Marks after sorting

Array of FE Marks : 75.00 80.00

Top Scorers :

80.00

75.00

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 3

End of Program

Practical No: 06 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.

- **Program:**

```
def accept_array(A):
    n = int(input("Enter the total no. of student : "))
    for i in range(n):
        x = float(input("Enter the first year percentage of student %d : "%(i+1)))
        A.append(x)
    print("Array accepted successfully\n\n");
```

```
def display_array(A):
    n = len(A)
    if(n == 0) :
        print("\nNo records in the database")
    else :
        print("Array of FE Marks : ",end=' ')
        for i in range(n) :
            print("%.2f "%A[i],end=' ')
        print("\n");
```

```
def partition(A,s,l) :
    b=s+1
    e=l
    while(e>=b) :
        while(b<=l and A[s] >= A[b]) :
            b = b + 1
        while(A[s] <A[e]) :
            e = e - 1
        if(e>b) :
            temp = A[e]
            A[e] = A[b]
            A[b] = temp
```

```
temp = A[s]
A[s] = A[e]
A[e] = temp
return e
```

```
def Quicksort(A,s,l) :
    if(s<l) :
        mid = partition(A,s,l)
        Quicksort(A,s,mid-1)
        Quicksort(A,mid+1,l)
```

```
def Main() :
    A = []
    while True :
        print ("\t1 : Accept & Display the FE Marks")
        print ("\t2 : Quick sort ascending order and display top five scores")
        print ("\t3 : Exit")
        ch = int(input("Enter your choice : "))
        if (ch == 3):
            print ("End of Program")
            quit()
        elif (ch==1):
            A = []
            accept_array(A)
            display_array(A)
        elif (ch==2):
            print("Marks before sorting")
            display_array(A)
            n =len(A)
            Quicksort(A,0,n-1)
            print("Marks after sorting")
            display_array(A)
```

```
if(n >= 5) :
    print("Top Five Scores : ")
    for i in range(n-1,n-6,-1) :
        print("\t%.2f"%A[i])
    else :
        print("Top Scorers : ")
        for i in range(n-1,-1,-1) :
            print("\t%.2f"%A[i])
    else :
        print ("Wrong choice entered !! Try again")
Main()
```

- **Output:**

jaihind@jaihind-ThinkCentre-M60e:\$ g++ pr6.cpp

jaihind@jaihind-ThinkCentre-M60e:\$./a.out

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 1

Enter the total no. of student : 5

Enter the first year percentage of student 1 : 70

Enter the first year percentage of student 2 : 75

Enter the first year percentage of student 3 : 80

Enter the first year percentage of student 4 : 85

Enter the first year percentage of student 5 : 90

Array accepted successfully

Array of FE Marks : 70.00 75.00 80.00 85.00 90.00

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 2

Roll No. 65

Marks before sorting

Array of FE Marks : 70.00 75.00 80.00 85.00 90.00

Marks after sorting

Array of FE Marks : 70.00 75.00 80.00 85.00 90.00

Top Five Scores :

90.00

85.00

80.00

75.00

70.00

1 : Accept & Display the FE Marks

2 : Quick sort ascending order and display top five scores

3 : Exit

Enter your choice : 3

End of Program

Practical No 07: Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C++ program to store two sets using linked list. Compute and display-

- a) Set of students who like both vanilla and butterscotch**
- b) Set of students who like either vanilla or butterscotch or not both**
- c) Number of students who like neither vanilla nor butterscotch**

```
#include <iomanip>
#include <iostream>
using namespace std;

class Student
{
    int rollNo;
    string name;
    Student *next;
    friend class SecondYear;
public:
    Student()
    {
        rollNo=-1;
        name="";
        next=NULL;
    }
    Student(int rollNo,string name)
    {
        next=NULL;
        this->rollNo=rollNo;
        this->name=name;
    }
    void printStudentData()
    {
        cout<<endl<<setw(7)<<rollNo<<setw(20)<<name;
```

```
    }  
};  
class SecondYear  
{  
    Student *start;  
public:  
    SecondYear()  
{  
    start=NULL;  
}  
    Student* getStart()  
{  
    return start;  
}  
    void AddStudent(int rollNo,string name)  
{  
        Student *s=new Student(rollNo,name);  
        if(start==NULL)  
            start=s;  
        else  
        {  
            s->next=start;  
            start=s;  
        }  
    }  
    bool deleteStudent(int rollNo)  
{  
        if(start==NULL)  
            return false;  
        else if(start->next==NULL)  
        {  
            Student *temp=start;  
            start=NULL;  
            delete temp;  
        }  
    }  
}
```



```
        return true;
    }
    else
    {
        Student *prev=NULL,*temp=start;
        while(temp!=NULL)
        {
            if(temp->rollNo==rollNo)
                break;
            prev=temp;
            temp=temp->next;
        }
        if(temp==NULL)
            return false;
        if(temp==start)
            start=start->next;
        else
            prev->next=temp->next;
        delete temp;
        return true;
    }
}

void onlySet(SecondYear *list2)
{
    Student *temp2=list2->start;
    Student *temp1=this->start;
    while(temp1!=NULL)
    {
        bool isPresent=false;
        temp2=list2->start;
        while(temp2!=NULL)
        {
            if(temp1->rollNo==temp2->rollNo)
            {
```

```
                isPresent=true;
                break;
            }
            temp2=temp2->next;
        }
        if(!isPresent)
            temp1->printStudentData();
        temp1=temp1->next;
    }
}

SecondYear* unionSet(SecondYear *list2)
{
    Student *temp1=start;
    Student *temp2=list2->start;
    SecondYear *temp3=new SecondYear;
    while(temp1!=NULL)
    {
        //temp1->printStudentData();
        temp3->AddStudent(temp1->rollNo,temp1->name);
        temp1=temp1->next;
    }
    while(temp2!=NULL)
    {
        bool isPresent=false;
        temp1=start;
        while(temp1!=NULL)
        {
            if(temp1->rollNo==temp2->rollNo)
            {
                isPresent=true;
                break;
            }
            temp1=temp1->next;
        }
    }
}
```

```
        if(!isPresent)
        {
            // temp2->printStudentData();
            temp3->AddStudent(temp2->rollNo,temp2->name);
        }
        temp2=temp2->next;
    }
    return temp3;
}

void displayNone(SecondYear *unionSetAB)
{
    Student *temp1=start;
    Student *temp2=unionSetAB->start;

    while(temp1!=NULL)
    {
        bool isPresent=false;
        temp2=unionSetAB->start;
        while(temp2!=NULL)
        {
            if(temp1->rollNo==temp2->rollNo)
            {
                isPresent=true;
                break;
            }
            temp2=temp2->next;
        }
        if(!isPresent)
            temp1->printStudentData();
        temp1=temp1->next;
    }

}

void intersection(SecondYear *list2)
```

```
{
    Student *temp1=start;
    Student *temp2=NULL;
    while(temp1!=NULL)
    {
        temp2=list2->start;
        while(temp2!=NULL)
        {
            if(temp1->rollNo==temp2->rollNo)
            {
                temp1->printStudentData();
                break;
            }
            temp2=temp2->next;
        }
        temp1=temp1->next;
    }
}

void displayStudents()
{
    Student *temp=start;
    cout<<endl<<setw(7)<<"RollNos" <<setw(20)<<"Names";
    while(temp!=NULL)
    {
        temp->printStudentData();
        temp=temp->next;
    }
}

};

int main()
{
```

```

SecondYear *A,*B,*universalSet,*temp=NULL;
A=new SecondYear;
B=new SecondYear;
universalSet=new SecondYear;
int x=-1,opt,flav;
int rollNo;
string name;
bool del=false;
do
{
    cout<<endl;
    cout<<"*****      Second      Year      Students      Volatile      Database
*****";
    cout<<endl<<"1.Enter      student\n2.Delete      Student\n3.Display
students\n4.Display Students(like-Vanilla)";
    cout<<"\n5.Display      Students(like-Butterscotch)"<<"\n6.Display
Students(like-both Vanilla and ButterScotch)";
    cout<<"\n7.Display Students(like-none of them)\n8.Display Students(like-
either Vanilla or ButterScotch or Both)\n9.Exit\nEnter your choice :";
    cin>>opt;
    switch(opt)
    {
    case 1:
        cout<<"\nEnter the name of student :";
        cin>>name;
        cout<<"\nEnter RollNo :";
        cin>>rollNo;
        cout<<"\nEnter      the      flavour
liked:\n1.ButterScotch\n2.Vanilla\n3.Both\n4.None\n Enter your choice :";
        cin>>flav;
        switch(flav)
        {
        case 1:B->AddStudent(rollNo,name);
        break;

```

```
case 2:A->AddStudent(rollNo,name);
break;
case 3:A->AddStudent(rollNo,name);
B->AddStudent(rollNo,name);
break;
case 4:break;
default:"Wrong option";
}
if(flav>=1 && flav<=4)
    universalSet->AddStudent(rollNo,name);
break;
case 2:
    cout<<"Enter rollNo :";
    cin>>rollNo;
    del=false;
    if(A->deleteStudent(rollNo))
    {
        universalSet->deleteStudent(rollNo);
        del=true;
    }
    if(B->deleteStudent(rollNo))
    {
        universalSet->deleteStudent(rollNo);
        del=true;
    }
    if(universalSet->deleteStudent(rollNo))
        del=true;
    if(!del)
        cout<<endl<<"No such student";
    else
        cout<<endl<<"Deleted from database";
    break;
case 3:
    universalSet->displayStudents();
```

```
        break;
    case 4:
        A->onlySet(B);
        break;
    case 5:
        B->onlySet(A);
        break;
    case 6:
        A->intersection(B);
        break;
    case 7:
        temp=A->unionSet(B);
        universalSet->displayNone(temp);
        delete temp;
        break;
    case 8:temp=A->unionSet(B);
        temp->displayStudents();
        delete temp;
        break;
    case 9:break;
    default:cout<<"Wrong option";
    break;
}
}while(x!=9);

}
```

Output:

/tmp/FfQC890wxU.o

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)

5.Display Students(like-Butterscotch)

6.Display Students(like-both Vanilla and ButterScotch)7.Display Students(like-none of them)

8.Display Students(like-either Vanilla or ButterScotch or Both)

9.Exit

Enter your choice :1

Enter the name of student :Omkar

Enter RollNo :22

Enter the flavour liked:

1.ButterScotch

2.Vanilla

3.Both

4.None

Enter your choice :1

***** Second Year Students Volatile Database *****

1.Enter student

2.Delete Student

3.Display students

4.Display Students(like-Vanilla)

5.Display Students(like-Butterscotch)

6.Display Students(like-both Vanilla and ButterScotch)

7.Display Students(like-none of them)

8.Display Students(like-either Vanilla or ButterScotch or Both)

9.Exit

Enter your choice :2

Enter rollNo :22

Deleted from database

***** Second Year Students Volatile Database *****

1.Enter student

2.Delete Student

3.Display students

4.Display Students(like-Vanilla)

5.Display Students(like-Butterscotch)

6.Display Students(like-both Vanilla and ButterScotch)

- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :3

RollNos	Names
---------	-------

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)
- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :4

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)
- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :5

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)

- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :6

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)
- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :7

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)
- 7.Display Students(like-none of them)
- 8.Display Students(like-either Vanilla or ButterScotch or Both)
- 9.Exit

Enter your choice :8

RollNos	Names
---------	-------

***** Second Year Students Volatile Database *****

- 1.Enter student
- 2.Delete Student
- 3.Display students
- 4.Display Students(like-Vanilla)
- 5.Display Students(like-Butterscotch)
- 6.Display Students(like-both Vanilla and ButterScotch)

7.Display Students(like-none of them)

8.Display Students(like-either Vanilla or ButterScotch or Both)

9.Exit

Enter your choice :9

***** Second Year Students Volatile Database *****

1.Enter student

2.Delete Student

3.Display students

4.Display Students(like-Vanilla)

5.Display Students(like-Butterscotch)

6.Display Students(like-both Vanilla and ButterScotch)

7.Display Students(like-none of them)

8.Display Students(like-either Vanilla or ButterScotch or Both)

9.Exit

Enter your choice :

Practical No 08: Write C++ program for storing binary number using doubly linked lists. Write functions-

- a) To compute 1's and 2's complement**
- b) Add two binary numbers**

Practical No 9: A palindrome is a string of character that,,s the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, “Poor Dan is in a droop” is a palindrome, as can be seen by examining the characters “poor dan is in a droop” and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functionsa) To print original string followed by reversed string using stack b) To check whether given string is palindrome or not.

```
#include<iostream>
```

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

```
#define SIZE 20
```

```
using namespace std;
```

```
class mystack
```

```
{
```

```
    private :
```

```
        char ST[SIZE];
```

```
        int top;
```

```
    public :
```

```
        mystack();
```

```
        void push(char X);
```

```
        char pop();
```

```
        int isEmpty();
```

```
        int isFull();
```

```
};
```

```
mystack :: mystack()
```

Roll No. 65

```
{  
    top = -1;  
}
```

int mystack :: isEmpty()

```
{  
    if(top == -1)  
        return 1;  
    else  
        return 0;  
}
```

int mystack :: isFull()

```
{  
    if(top == SIZE-1)  
        return 1;  
    else  
        return 0;  
}
```

void mystack :: push(char X)

```
{  
    if(!isFull())  
    {  
        top++;  
        ST[top] = X;  
    }  
}
```

Roll No. 65

```
        else

            cout<<"\nStack Overflow !! Error!!";

    }
```

```
char mystack :: pop()
```

```
{

    char X = '\0';

    X = ST[top];

    top--;

    return X;

}
```

```
void convert_string(char Str[],char Str1[])
```

```
{

    int i,j = 0;

    for(i=0;Str[i] != '\0';i++)

    {

        if(Str[i] >= 'a' && Str[i] <= 'z')

            Str1[j++] = Str[i];

        if(Str[i] >= 'A' && Str[i] <= 'Z')

            Str1[j++] = Str[i] + 32;

    }

    Str1[j] = '\0';

}
```

```
int main()
```

```
{  
  
    int ch,flag,i;  
    char Str[80],Str1[80];  
    mystack S;  
    system("clear");  
    do  
    {  
  
        cout<<"\n\t\t\t1 : Check for Palindrome";  
  
        cout<<"\n\t2 : Find Reverse";  
  
        cout<<"\n\t3 : Exit";  
  
        cout<<"\n\nEnter your choice : ";  
        cin>>ch;  
        switch(ch)  
        {  
  
            case 1 : cout<<"\n\nEnter the string to be checked for palindrome : ";  
                     cin.ignore();  
                     cin.getline(Str,79);  
                     cout<<"\nEntered String is "<<Str;  
                     convert_string(Str,Str1);  
                     cout<<"\nconverted String is : "<<Str1;  
                     for(i = 0; Str1[i] != '\0';i++)  
                         S.push(Str1[i]);  
                     i = 0; flag = 1;  
                     while(!S.isEmpty())  
                     {  
                         if(Str1[i++] != S.pop())  
                             flag = 0;
```



```
        }
        if(flag == 1)
            cout<<"\nGiven string is a palindrome\n";
        else
            cout<<"\nGiven String is not a palindrome\n";
        break;
    case 2 : cout<<"\nEnter the string to be reversed : ";
            cin.ignore();
            cin.getline(Str,79);
            cout<<"\nString entered is "<<Str;

            for(i = 0; Str[i] != '\0';i++)
                S.push(Str[i]);
            cout<<"\nReverse String = ";
            while(!S.isEmpty())
            {
                cout<<S.pop();
            }
            break;
    case 3 : cout<<"\nEnd of Program\n";
            break;
    default: cout<<"\nInvalid choice !! Try again\n\n";

    }
} while(ch != 3);
return 0;
}
```

Output :

Roll No. 65

1 : Check for Palindrome

2 : Find Reverse

3 : Exit

Enter your choice : 1

Enter the string to be checked for palindrome : Racecar

Entered String is Racecar

converted String is : racecar

Given string is a palindrome

1 : Check for Palindrome

2 : Find Reverse

3 : Exit

Enter your choice : 2

Enter the string to be reversed : hii hello hey

String entered is hii hello hey

Reverse String = yyeh olleh iih

1 : Check for Palindrome

2 : Find Reverse

3 : Exit

Enter your choice : 3

End of Program

Practical 10 : In any language program mostly syntax error occurs due to unbalancing delimiter such as (), {}, []. Write C++ program using stack to check whether given expression is well parenthesized or not.

(Code 1)

```
#include <iostream>

using namespace std;

#define size 10

class stackexp
{
    int top;
    char stk[size];
public:
    stackexp()
    {
        top=-1;
    }
    void push(char);
    char pop();
    int isfull();
    int isempty();
};

void stackexp::push(char x)
{
    top=top+1;
    stk[top]=x;
```

Roll No. 65

```
}
```

```
char stackexp::pop()
```

```
{
```

```
    char s;
```

```
    s=stk[top];
```

```
    top=top-1;
```

```
    return s;
```

```
}
```

```
int stackexp::isfull()
```

```
{
```

```
    if(top==size)
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
int stackexp::isempty()
```

```
{
```

```
    if(top==-1)
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
int main()
```

```
{
    stackexp s1;
    char exp[20],ch;
    int i=0;
    cout << "\n\t!!Well Formness of Parenthesis..!!!" << endl; // prints !!!Hello World!!!
    cout<<"\nEnter the expression to check whether it is in well form or not : ";
    cin>>exp;
    if((exp[0]=='')||(exp[0]=='')||(exp[0]==''))
    {
        cout<<"\n Invalid Expression.....\n";
        return 0;
    }
    else
    {
        while(exp[i]!='\0')
        {
            ch=exp[i];
            switch(ch)
            {
                case '(':s1.push(ch);break;
                case '[':s1.push(ch);break;
                case '{':s1.push(ch);break;
                case ')':s1.pop();break;
                case ']':s1.pop();break;
                case '}':s1.pop();break;
            }
            i=i+1;
        }
    }
}
```

Roll No. 65

```
    }  
}  
if(s1.isEmpty())  
{  
    cout<<"\nExpression is well parenthesis...\n";  
}  
else  
{  
    cout<<"\nSorry !!! Invalid Expression or not in well parenthesized....\n";  
}  
return 0;  
}
```

Output :

!!Well Formness of Parenthesis..!!!!

Enter the expression to check whether it is in well form or not : (a+b)(c-d)

Expression is well parenthesis...

(Code 2)

```
#include<stdio.h>  
#include<stdlib.h>  
#include<iostream>  
using namespace std;  
#define bool int  
  
struct sNode
```

Roll No. 65

```
{  
    char data;  
    struct sNode *next;  
};
```

```
void push(struct sNode** top_ref, int new_data);
```

```
int pop(struct sNode** top_ref);
```

```
bool isMatchingPair(char character1, char character2)
```

```
{  
    if (character1 == '(' && character2 == ')')  
        return 1;  
    else if (character1 == '{' && character2 == '}')  
        return 1;  
    else if (character1 == '[' && character2 == ']')  
        return 1;  
    else  
        return 0;  
}
```

```
bool areParenthesisBalanced(char exp[])
```

```
{  
    int i = 0;  
  
    struct sNode *stack = NULL;
```

Roll No. 65

```
while (exp[i])
{

    if (exp[i] == '{' || exp[i] == '(' || exp[i] == '[')
        push(&stack, exp[i]);

    if (exp[i] == '}' || exp[i] == ')' || exp[i] == ']')
    {
/
        if (stack == NULL)
            return 0;

        else if ( !isMatchingPair(pop(&stack), exp[i]) )
            return 0;
    }
    i++;
}

if (stack == NULL)
    return 1;
else
    return 0;
}

int main()
{
    char exp[100] ;
```


Roll No. 65

```
cout<<"enter:";
cin>>exp;
if (areParenthesisBalanced(exp))
    printf("\n Balanced ");
else
    printf("\n Not Balanced ");
return 0;
}

void push(struct sNode** top_ref, int new_data)
{
    struct sNode* new_node =
        (struct sNode*) malloc(sizeof(struct sNode));

    if (new_node == NULL)
    {
        printf("Stack overflow \n");
        getchar();
        exit(0);
    }

    new_node->data = new_data;

    new_node->next = (*top_ref);

    (*top_ref) = new_node;
}
```

Roll No. 65

```
int pop(struct sNode** top_ref)
{
    char res;

    struct sNode *top;

    if (*top_ref == NULL)
    {
        printf("Stack overflow \n");
        getchar();
        exit(0);
    }
    else
    {
        top = *top_ref;
        res = top->data;
        *top_ref = top->next;
        free(top);
        return res;
    }
}
```

Output :

enter:(a+b)(c-d)

Balanced

Practical 11 : Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue.

```
#include <iostream>

using namespace std;

#define size 5

class spq
{
    int f,r,job,djob;
    int simpq[size],prioq[size];
public:
    spq()
    {
        f=r=-1;
        job=djob=0;
        prioq[-1]=0;
    }

    int isQfull()
    {
        if(r==size-1)
            return 1;
        else
            return 0;
    }
};
```

Roll No. 65

```
}
```

```
int isQempty()
```

```
{
```

```
    if((f==1)|| (f>r))
```

```
        return 1;
```

```
    else
```

```
        return 0;
```

```
}
```

```
void simpqadd();
```

```
void showsimpleQ();
```

```
void delsimpleQ();
```

```
void prioqadd();
```

```
void delprioQ();
```

```
void showprioQ();
```

```
};
```

```
void spq::simpqadd()
```

```
{
```

```
    cout<<"\nEnter the Job: ";
```

```
    cin>>job;
```

```
    if(isQfull())
```

```
        cout<<"\nSorry !! Queue is full....\n";
```

```
    else
```

```
    {
```

```
        if(f==1)
```

```
        {
```

Roll No. 65

```
        f=r=0;
        simpq[r]=job;
    }
    else
    {
        r=r+1;
        simpq[r]=job;
    }
}

}

void spq::delsimpleQ()
{
    if(isQempty())
        cout<<"\nSorry Q is empty...\n";
    else
    {
        djob=simpq[f];
        f=f+1;
        cout<<"\nDeleted job is: "<<djob;
    }
}
```

```
void spq::showsimpleQ()
{
    cout<<"\nThe simple Queue job are as follows....\n";
```

Roll No. 65

```
int temp;
for(temp=f;temp<=r;temp++)
{
    cout<<"\t"<<simpq[temp];
}
}
```

```
void spq::delprioQ()
{
    if(isQempty())
        cout<<"\nSorry Q is empty...\n";
    else
    {
        djob=prioq[f];
        f=f+1;
        cout<<"\nDeleted job is: "<<djob;
    }
}
```

```
void spq::showprioQ()
{
    cout<<"\nThe priority Queue job are as follows....\n";
    int temp;
    for(temp=f;temp<=r;temp++)
    {
        cout<<"\t"<<prioq[temp];
    }
}
```

```
}
```

```
void spq::prioqadd()
```

```
{
```

```
    int t=0;
```

```
    cout<<"\nEnter the job: ";
```

```
    cin>>job;
```

```
    if(isQfull())
```

```
        cout<<"\nSorry!! Priority Queue is full...\n";
```

```
    else
```

```
    {
```

```
        if(f==-1)
```

```
        {
```

```
            f=r=0;
```

```
            prioq[r]=job;
```

```
        }
```

```
        else if(job<prioq[r])
```

```
        {
```

```
            t=r;
```

```
            while(job<prioq[t])
```

```
            {
```

```
                prioq[t+1]=prioq[t];
```

```
            }
```

```
            t=t+1;
```

```
            prioq[t]=job;
```

```
            r=r+1;
```

```
        }
```

```
        else
        {
            r=r+1;
            prioq[r]=job;
        }
    }
}

int main()
{
    spq s1,s2;
    int ch;
    do
    {
        cout<< "\n\t!!!Operating System Job Queue!!!" << endl; // prints the msg.
        cout<< "\n1.SimpleQ   Add_Job\n2.SimpleQ   Del_Job\n3.Show   SimpleQ\n4.PrioQ
Add_Job\n5.PrioQ Del_Job\n6.Show PrioQ";
        cout<< "\nEnter Your Choice:";
        cin>>ch;
        switch(ch)
        {
            case 1:s1.simpqadd();break;
            case 2:s1.delsimpleQ();break;
            case 3:s1.showsimpleQ();break;
            case 4:s2.prioqadd();break;
            case 5:s2.delprioQ();break;
            case 6:s2.showprioQ();break;
        }
    }
```


Roll No. 65

```
    }while(ch!=7);  
    return 0;  
}
```

Output :

Practical 12 : A double-ended queue (deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one- dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque.

```
#include<iostream>

#include<stdlib.h>

using namespace std;

#define SIZE 5

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:
        cout<<"You exited from program.....";
        exit(1);
        break;
    default:
        cout<<"Invalid choice";
        break;
    }
}while(c!=7);
return 0;
}
```

Output :

/tmp/eLy4kztnKL.o

****DEQUEUE OPERATION****

Roll No. 65

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:22

****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:11

****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

Roll No. 65

The deleted element is 22

****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 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>:6

You exited from program.....

Practical No 13: Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array.

```
#include <iostream>
using namespace std;
#define size 5
class pizza
{
    int porder[size];
    int front,rear;
public:
    pizza()
    {
        front=rear=-1;
    }
    int qfull()
    {
        if((front==0)&&(rear==(size-1))||(front==(rear+1)%size))
            return 1;
        else
            return 0;
    }
    int qempty()
    {
        if(front==-1)
            return 1;
        else
            return 0;
    }
    void accept_order(int);
    void make_payment(int);
    void order_in_queue();
};
```

```
void pizza::accept_order(int item)
{
    if(qfull())
        cout<<"\nVery Sorry !!!! No more orders....\n";
    else
    {
        if(front==-1)
        {
            front=rear=0;
        }
        else
        {
            rear=(rear+1)%size;
        }
        porder[rear]=item;
    }
}
```

```
void pizza::make_payment(int n)
{
    int item;
    char ans;
    if(qempty())
        cout<<"\nSorry !!! order is not there...\n";
    else
    {
        cout<<"\nDeliverd orders as follows...\n";
        for(int i=0;i<n;i++)
        {
            item=porder[front];
            if(front==rear)
            {
                front=rear=-1;
            }
        }
    }
}
```

```
        else
        {
            front=(front+1)%size;
        }
        cout<<"\t"<<item;
    }
    cout<<"\nTotal amount to pay : "<<n*100;
    cout<<"\nThank you visit Again....\n";
}

}

void pizza::order_in_queue()
{
    int temp;
    if(qempty())
    {
        cout<<"\nSorry !! There is no pending order...\n";
    }
    else
    {
        temp=front;
        cout<<"\nPending Order as follows..\n";
        while(temp!=rear)
        {
            cout<<"\t"<<porder[temp];
            temp=(temp+1)%size;
        }
        cout<<"\t"<<porder[temp];
    }
}

int main()
{
    pizza p1;
    int ch,k,n;
```

```
do
{
cout<<"\n\t***** Welcome To Pizza Parlor *****\n";
cout << "\n1.Accept order\n2.Make_payment\n3.Pending Orders\nEnter u r choice: ";
cin>>ch;
switch(ch)
{
case 1:cout<<"\nWhich Pizza do u like most....\n";
        cout<<"\n1.Veg Soya Pizza\n2.Veg butter Pizza\n3.Egg_Pizza";
        cout<<"\nPlease enter u r order: ";
        cin>>k;
        p1.accept_order(k);
        break;
case 2:cout<<"\nNumber of pizza =";
        cin>>n;
        p1.make_payment(n);
        break;
case 3:cout<<"\n Following orders are in queue to deliver....as follows..\n";
        p1.order_in_queue();
        break;
}
}while(ch!=4);

return 0;
}
```

Output :

***** Welcome To Pizza Parlor *****

1.Accept order

2.Make_payment

3.Pending Orders

Enter u r choice: 1

Which Pizza do u like most....

- 1.Veg Soya Pizza
- 2.Veg butter Pizza
- 3.Egg_Pizza

Please enter u r order: 3

***** Welcome To Pizza Parlor *****

- 1.Accept order
- 2.Make_payment
- 3.Pending Orders

Enter u r choice: 2

Number of pizza = 3

3

Deliverd orders as follows...

3 2 2

Total amount to pay : 300

Thank you visit Again....

***** Welcome To Pizza Parlor *****

- 1.Accept order
- 2.Make_payment
- 3.Pending Orders

Enter u r choice: 3

Following orders are in queue to deliver....as follows..

Sorry !! There is no pending order...

***** Welcome To Pizza Parlor *****