**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

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 substraction\_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) :

substraction\_matrix(M1,M2,M3,r1,c1)

print("\tSubstraction ")

display\_matrix(M3,r1,c1)

else :

print("substraction 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 :

20

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

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

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:$ python pr6.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 : 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

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