Group (A)

Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 01

Program- A-1 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: -

1. List of students who play both cricket and badminton
2. List of students who play either cricket or badminton but not both
3. Number of students who play neither cricket nor badminton
4. Number of students who play cricket and football but not badminton.

(Note- While realizing the group, duplicate entries should be avoided, Do not use SET built- in functions)

Code:

def removeDuplicate(d):

    lst=[]

    for i in d:

        if i not in lst:

            lst.append(i)

    return lst

# Function for finding intersection between two sets (A&B)

def intersection(lst1,lst2):

    lst3=[]

    for val in lst1:

        if val in lst2:

            lst3.append(val)

    return lst3

# Function for finding union of two sets (A|B)

def union(lst1,lst2):

    lst3=lst1.copy()

    for val in lst2:

        if val not in lst3:

            lst3.append(val)

    return lst3

# Function for finding difference between two sets (A-B)

def diff(lst1,lst2):

    lst3=[]

    for val in lst1:

        if val not in lst2:

            lst3.append(val)

    return lst3

# Function for finding symmetric difference of two sets (A^B)

def sym\_diff(lst1,lst2):

    lst3=[]

    D1=diff(lst1,lst2)

    print("Difference between Cricket and Badminton (C-B) is : ", D1)

    D2=diff(lst2,lst1)

    print("Difference between Badminton and Cricket (B-C) is : ", D2)

    lst3=union(D1,D2)

    return lst3

# Functon for finding List of students who play both cricket and badminton

def CB(lst1,lst2):

    lst3=intersection(lst1,lst2)

    print("\n\nList of students who play both cricket and badminton is : ", lst3)

    return len(lst3)

# Function for finding List of students who play either cricket or badminton but not both

def eCeB(lst1,lst2):

    lst3=sym\_diff(lst1,lst2)

    print("\nList of students who play either cricket or badminton but not both is : ",lst3)

    return len(lst3)

# Function for finding Number of students who play neither cricket nor badminton

def nCnB(lst1,lst2,lst3):

    lst4=diff(lst1,union(lst2,lst3))

    print("\n\nList of students who play neither cricket nor badminton is : ",lst4)

    return len(lst4)

# Function for finding Number of students who play cricket and football but not badminton

def CBnF(lst1,lst2,lst3):

    lst4=diff(intersection(lst1,lst2),lst3)

    print("\n\nList of students who play cricket and football but not badminton is : ",lst4)

    return len(lst4)

# Main function

# Creating an empty list for SE COMP

SEComp = []

n = int(input("\nEnter number of students in SE COMP: "))

print("Enter the names of",n,"students (Please press ENTER after entering each students name) :")

for i in range(0, n):

    ele = input()

    SEComp.append(ele)  # adding the element

print("Original list of students in SEComp : " + str(SEComp))

# Creating an empty list for Cricket

Cricket = []

n = int(input("\n\nEnter number of students who play cricket : "))

print("Enter the names of",n,"students who play cricket (Please press ENTER after entering each students name) :")

for i in range(0, n):

    ele = input()

    Cricket.append(ele)  # adding the element

print("Original list of students playing cricket is :" +str(Cricket))

Cricket=removeDuplicate(Cricket)

print("The list of students playing cricket after removing duplicates : " +str(Cricket))

# Creating an empty list for Football

Football = []

n = int(input("\n\nEnter number of students who play football : "))

print("Enter the name of",n,"students who play football (Please press ENTER after entering each students name) :")

for i in range(0, n):

    ele = input()

    Football.append(ele)  # adding the element

print("Original list of students playing football :" +str(Football))

Football=removeDuplicate(Football)

print("The list of students playing football after removing duplicates : " +str(Football))

# Creating an empty list for Badminton

Badminton = []

n = int(input("\n\nEnter number of students who play badminton : "))

print("Enter the name of",n,"students who play badminton (Please press ENTER after entering each students name) :")

for i in range(0, n):

    ele = input()

    Badminton.append(ele)  # adding the element

print("Original list of students playing badminton :" +str(Badminton))

Badminton=removeDuplicate(Badminton)

print("The list of students playing badminton after removing duplicates : " +str(Badminton))

flag=1

while flag==1:

    print("\n\n--------------------MENU--------------------\n")

    print("1. List of students who play both cricket and badminton")

    print("2. List of students who play either cricket or badminton but not both")

    print("3. List of students who play neither cricket nor badminton")

    print("4. Number of students who play cricket and football but not badminton")

    print("5. Exit\n")

    ch=int(input("Enter your Choice (from 1 to 5) :"))

    if ch==1:

        print("Number of students who play both cricket and badminton : ", CB(Cricket,Badminton))

        a = input("\n\nDo you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==2:

        print("Number of students who play either cricket or badminton but not both : ", eCeB(Cricket, Badminton))

        a = input("\n\nDo you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==3:

        print("Number of students who play neither cricket nor badminton : ", nCnB(SEComp,Cricket,Badminton))

        a = input("\n\nDo you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==4:

        print("Number of students who play cricket and football but not badminton : ", CBnF(Cricket,Football,Badminton))

        a = input("\n\nDo you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==5:

        flag=0

        print("Thanks for using this program!")

    else:

        print("!!Wrong Choice!! ")

        a=input("\n\nDo you want to continue (yes/no) :")

        if a=="yes":

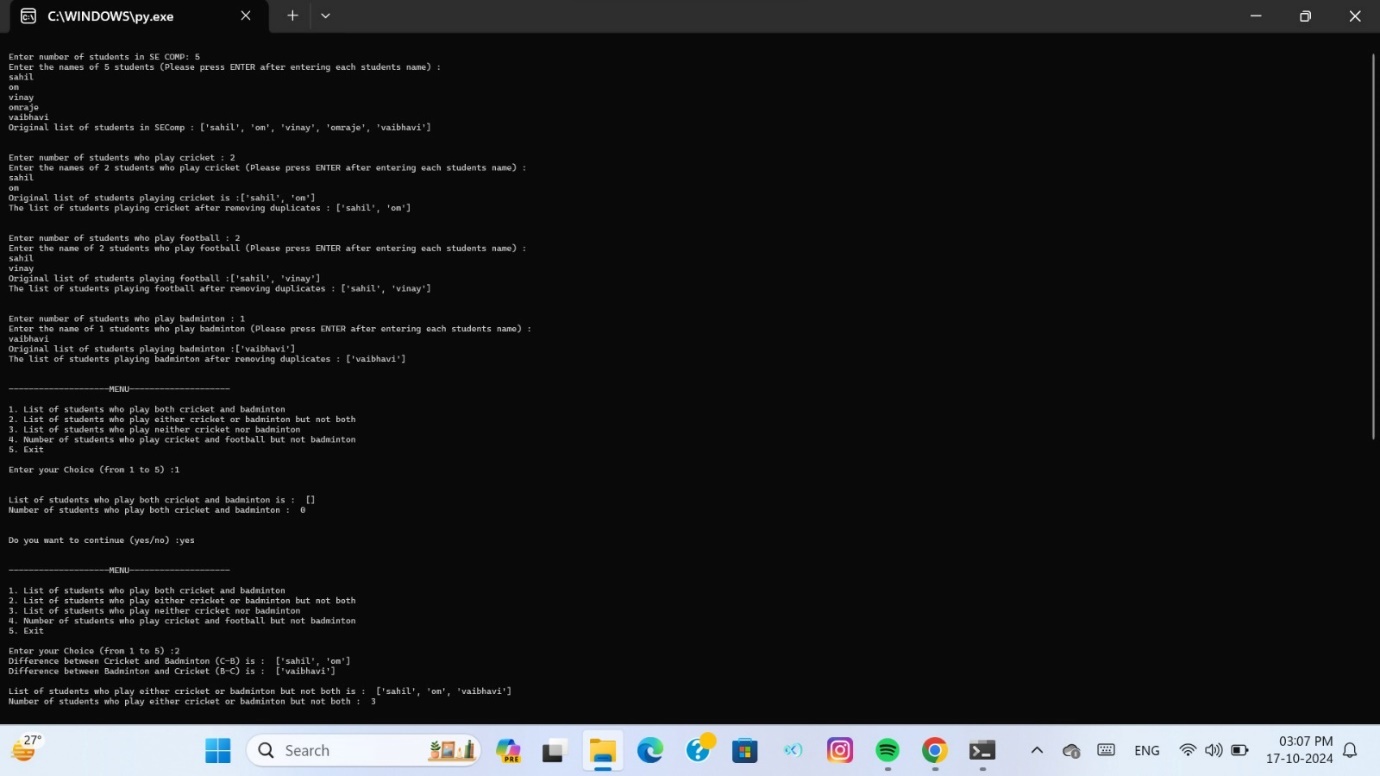
            flag=1

        else:

            flag=0

            print("Thanks for using this program!")

Output:



Name: Tejaswini Sachin Shewale

Roll No.:37

Assignment No,:02

A-2 Write a Python program to store marks scored in subject “Fundamental of Data Structure” by N students in the class. Write functions to compute following:

1. The average score of class
2. Highest score and lowest score of class
3. Count of students who were absent for the test

Display mark with highest frequency

Code:

# Function for average score of the class

def average(listofmarks):

    sum=0

    count=0

    for i in range(len(listofmarks)):

        if listofmarks[i]!=-999:

            sum+=listofmarks[i]

            count+=1

    avg=sum/count

    print("Total Marks : ", sum)

    print("Average Marks : {:.2f}".format(avg))

# Function for Highest score in the test for the class

def Maximum(listofmarks):

    for i in range(len(listofmarks)):

        if listofmarks[i]!=-999:

            Max=listofmarks[0]

            break

    for i in range(1,len(listofmarks)):

        if listofmarks[i]>Max:

            Max=listofmarks[i]

    return(Max)

# Function for Lowest score in the test for the class

def Minimum(listofmarks):

    for i in range(len(listofmarks)):

        if listofmarks[i]!=-999:

            Min=listofmarks[0]

            break

    for i in range(1,len(listofmarks)):

        if listofmarks[i]<Min:

            Min=listofmarks[i]

    return(Min)

# Function for counting the number of students absent for the test

def absentcount(listofmarks):

    count=0

    for i in range(len(listofmarks)):

        if listofmarks[i]==-999:

            count+=1

    return(count)

# Function for displaying marks with highest frequency

def maxFrequency(listofmarks):

    i=0

    Max=0

    print("Marks  |  Frequency")

    for j in listofmarks:

        if (listofmarks.index(j)==i):

            print(j,"    |  ",listofmarks.count(j))

            if listofmarks.count(j)>Max:

                Max=listofmarks.count(j)

                mark=j

        i=i+1

    return(mark,Max)

# Main function

marksinFDS=[]

numberofstudents=int(input("Enter total number of students : "))

for i in range(numberofstudents):

    marks=int(input("Enter marks of student "+str(i+1)+" : "))

    marksinFDS.append(marks)

flag=1

while flag==1:

    print("\n\n--------------------MENU--------------------\n")

    print("1. Total and Average Marks of the Class")

    print("2. Highest and Lowest Marks in the Class")

    print("3. Number of Students absent for the test")

    print("4. Marks with Highest Frequency")

    print("5. Exit\n")

    ch=int(input("Enter your Choice (from 1 to 5) :"))

    if ch==1:

        average(marksinFDS)

        a = input("Do you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==2:

        print("Highest Score in Class : ", Maximum(marksinFDS))

        print("Lowest Score in Class : ", Minimum(marksinFDS))

        a = input("Do you want to continue (yes/no) :")

        if a == "yes":  flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==3:

        print("Number of Students absent in the test : ", absentcount(marksinFDS))

        a = input("Do you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==4:

        mark,fr = maxFrequency(marksinFDS)

        print("Highest frequency is of marks {0} that is {1} ".format(mark,fr))

        a = input("Do you want to continue (yes/no) :")

        if a == "yes":

            flag = 1

        else:

            flag = 0

            print("Thanks for using this program!")

    elif ch==5:

        flag=0

        print("Thanks for using this program!")

    else:

        print("!!Wrong Choice!! ")

        a=input("Do you want to continue (yes/no) :")

        if a=="yes":

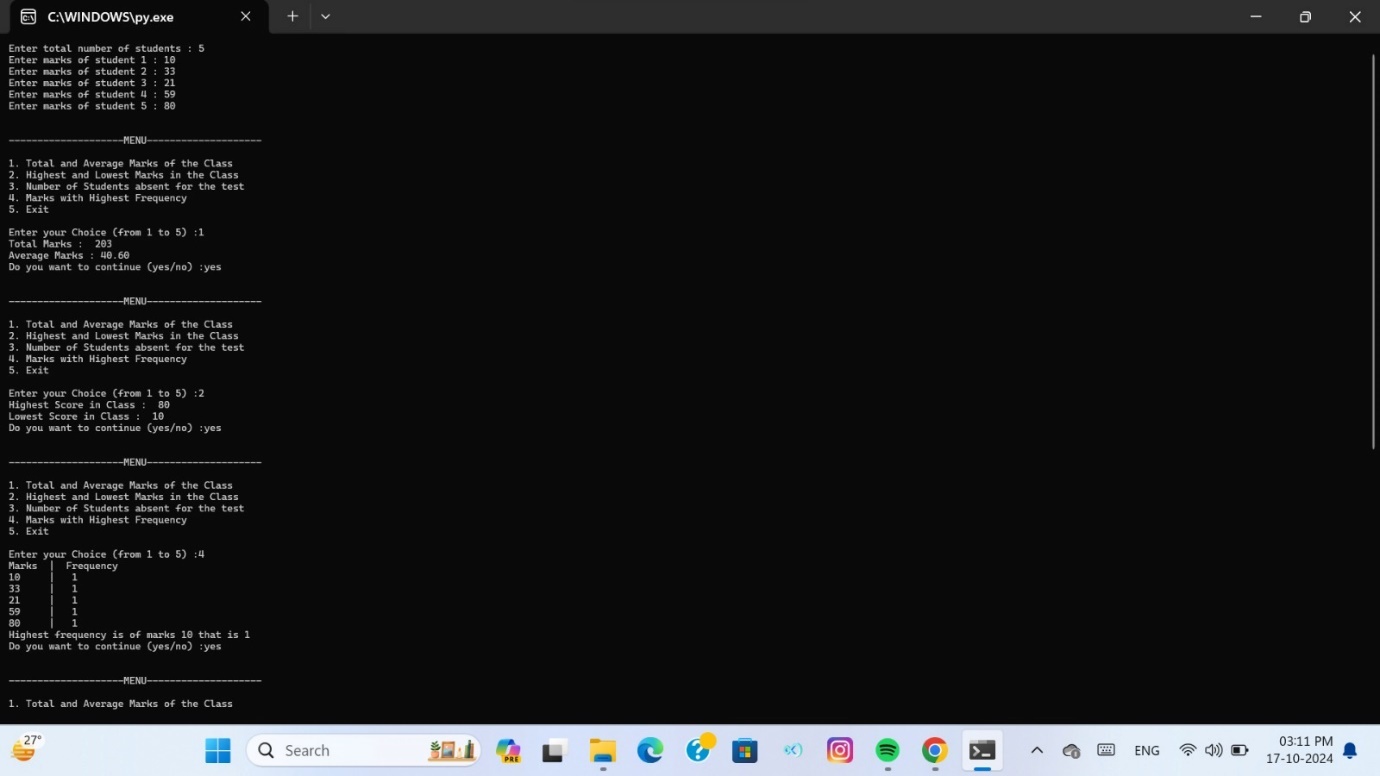
            flag=1

        else:

            flag=0

            print("Thanks for using this program!")

Output:



Name: Tejaswini Sachin Shewale

Roll no.: 37

Assignment No:03

A-5 Write a Python program to compute following operations on String:

1. To display word with the longest length
2. To determines the frequency of occurrence of particular character in the string
3. To check given string is palindrome or not
4. To display index of first substring

To count the occurrences of each word in string

Code:

def longest\_word(s):

    words = s.split()  # Split the string into words

    if not words:  # Handle empty strings

        return ""

    longest = max(words, key=len)  # Find the longest word

    return longest

def char\_frequency(s, char):

    if len(char) != 1:  # Ensure 'char' is a single character

        return "Please enter a valid single character."

    return s.count(char)  # Count occurrences of the character in the string

def is\_palindrome(s):

    s = s.replace(" ", "").lower()  # Ignore spaces and make the check case-insensitive

    return s == s[::-1]  # Check if the string is the same forwards and backwards

def first\_substring\_index(s, substring):

    return s.find(substring)  # Return the index of the first occurrence of the substring

def word\_occurrences(s):

    words = s.split()  # Split the string into words

    frequency = {}  # Create a dictionary to store word frequencies

    for word in words:

        frequency[word] = frequency.get(word, 0) + 1  # Increment word count

    return frequency

def main():

    input\_string = input("Enter a string: ")

    # a) Display the word with the longest length

    longest = longest\_word(input\_string)

    if longest:

        print("Longest word:", longest)

    else:

        print("The string is empty or contains no words.")

    # b) Determine the frequency of occurrence of a particular character

    char = input("Enter a character to find its frequency: ")

    char\_freq = char\_frequency(input\_string, char)

    print(f"Frequency of '{char}':", char\_freq)

    # c) Check if the string is a palindrome

    print("Is palindrome:", is\_palindrome(input\_string))

    # d) Display index of the first occurrence of a substring

    substring = input("Enter a substring to find its first index: ")

    index = first\_substring\_index(input\_string, substring)

    if index != -1:

        print(f"First index of '{substring}':", index)

    else:

        print(f"'{substring}' not found in the string.")

    # e) Count the occurrences of each word in the string

    word\_freq = word\_occurrences(input\_string)

    print("Word occurrences:", word\_freq)

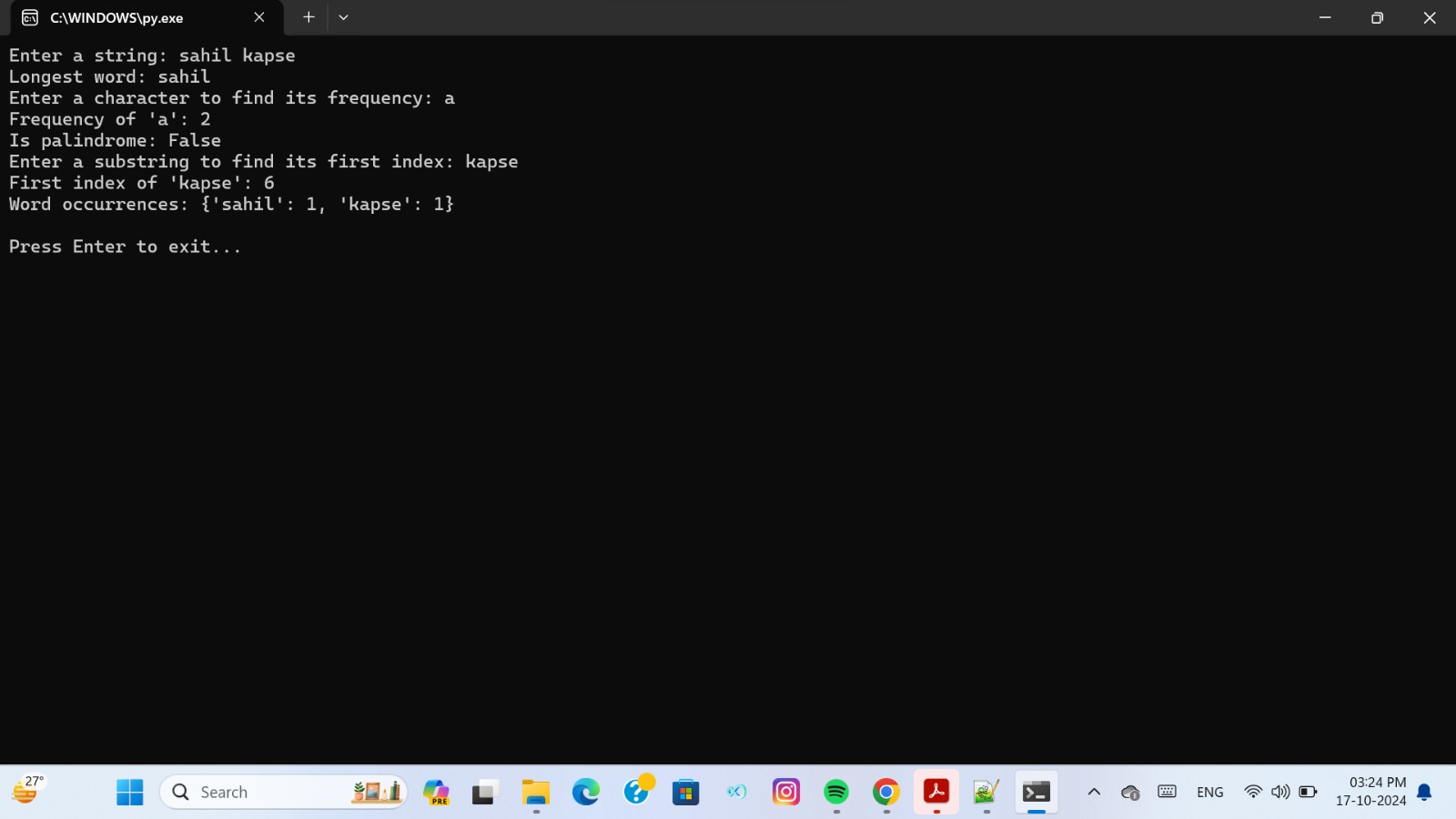
    # Pause the program to keep it open

    input("\nPress Enter to exit...")  # Wait for the user to press Enter before exiting

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



Name: Tejaswini Sachin Shewale

Roll no.: 37

Assignment no.:04

A-9 Write a **Python** program to compute following computation on matrix: a)Addition of two matrices

1. Subtraction of two matrices
2. Multiplication of two matrices

Transpose of a matrix

Code:

# Function to add two matrices

def add\_matrices(matrix1, matrix2):

    result = []

    for i in range(len(matrix1)):

        row = []

        for j in range(len(matrix1[0])):

            row.append(matrix1[i][j] + matrix2[i][j])

        result.append(row)

    return result

# Function to subtract two matrices

def subtract\_matrices(matrix1, matrix2):

    result = []

    for i in range(len(matrix1)):

        row = []

        for j in range(len(matrix1[0])):

            row.append(matrix1[i][j] - matrix2[i][j])

        result.append(row)

    return result

# Function to multiply two matrices

def multiply\_matrices(matrix1, matrix2):

    result = []

    for i in range(len(matrix1)):

        row = []

        for j in range(len(matrix2[0])):

            sum = 0

            for k in range(len(matrix2)):

                sum += matrix1[i][k] \* matrix2[k][j]

            row.append(sum)

        result.append(row)

    return result

# Function to transpose a matrix

def transpose\_matrix(matrix):

    result = []

    for i in range(len(matrix[0])):

        row = []

        for j in range(len(matrix)):

            row.append(matrix[j][i])

        result.append(row)

    return result

# Function to display a matrix

def display\_matrix(matrix):

    for row in matrix:

        print(row)

def main():

    # Input matrices

    matrix1 = [

        [1, 2, 3],

        [4, 5, 6],

        [7, 8, 9]

    ]

    matrix2 = [

        [9, 8, 7],

        [6, 5, 4],

        [3, 2, 1]

    ]

    print("Matrix 1:")

    display\_matrix(matrix1)

    print("\nMatrix 2:")

    display\_matrix(matrix2)

    # a) Addition of two matrices

    print("\nMatrix Addition:")

    addition\_result = add\_matrices(matrix1, matrix2)

    display\_matrix(addition\_result)

    # b) Subtraction of two matrices

    print("\nMatrix Subtraction:")

    subtraction\_result = subtract\_matrices(matrix1, matrix2)

    display\_matrix(subtraction\_result)

    # c) Multiplication of two matrices

    print("\nMatrix Multiplication:")

    multiplication\_result = multiply\_matrices(matrix1, matrix2)

    display\_matrix(multiplication\_result)

    # d) Transpose of a matrix

    print("\nTranspose of Matrix 1:")

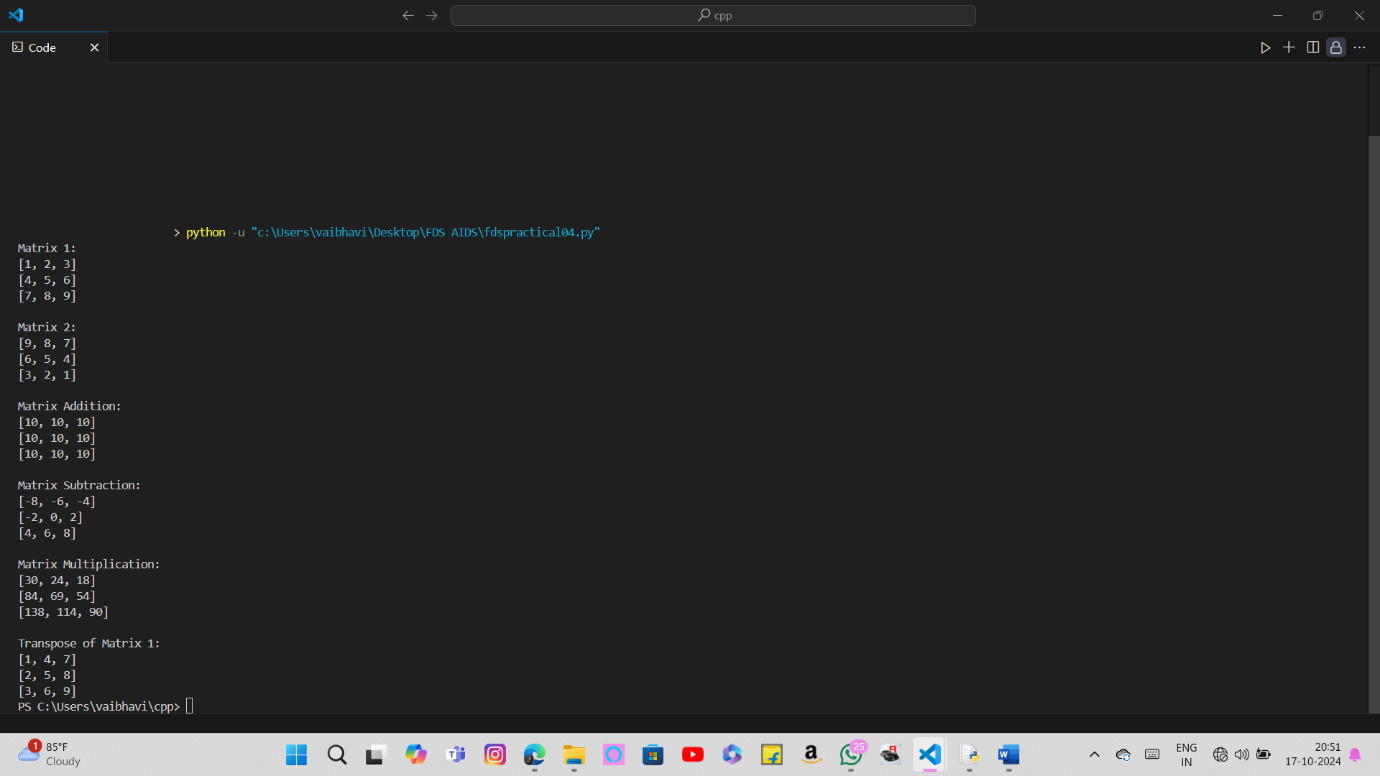
    transpose\_result = transpose\_matrix(matrix1)

    display\_matrix(transpose\_result)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



Group (B)

Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment no.: 05

Program- Write a Python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a)SelectionSort b) Bubble sort and display top five scores

Code:

def selection\_sort(arr):

    n = len(arr)

    for i in range(n):

        min\_idx = i

        for j in range(i + 1, n):

            if arr[j] < arr[min\_idx]:

                min\_idx = j

        arr[i], arr[min\_idx] = arr[min\_idx], arr[i]

def bubble\_sort(arr):

    n = len(arr)

    for i in range(n):

        for j in range(0, n - i - 1):

            if arr[j] > arr[j + 1]:

                arr[j], arr[j + 1] = arr[j + 1], arr[j]

def display\_top\_scores(arr, top\_n=5):

    print("Top {} scores:".format(top\_n))

    for score in arr[-top\_n:]:

        print(f"{score:.2f}")

def main():

    # Example percentages of students

    percentages = [85.5, 92.3, 76.4, 88.0, 67.5, 95.1, 84.2, 73.8, 91.0, 78.6]

    # Sort using Selection Sort

    sorted\_percentages\_selection = percentages[:]

    selection\_sort(sorted\_percentages\_selection)

    display\_top\_scores(sorted\_percentages\_selection)

    # Sort using Bubble Sort

    sorted\_percentages\_bubble = percentages[:]

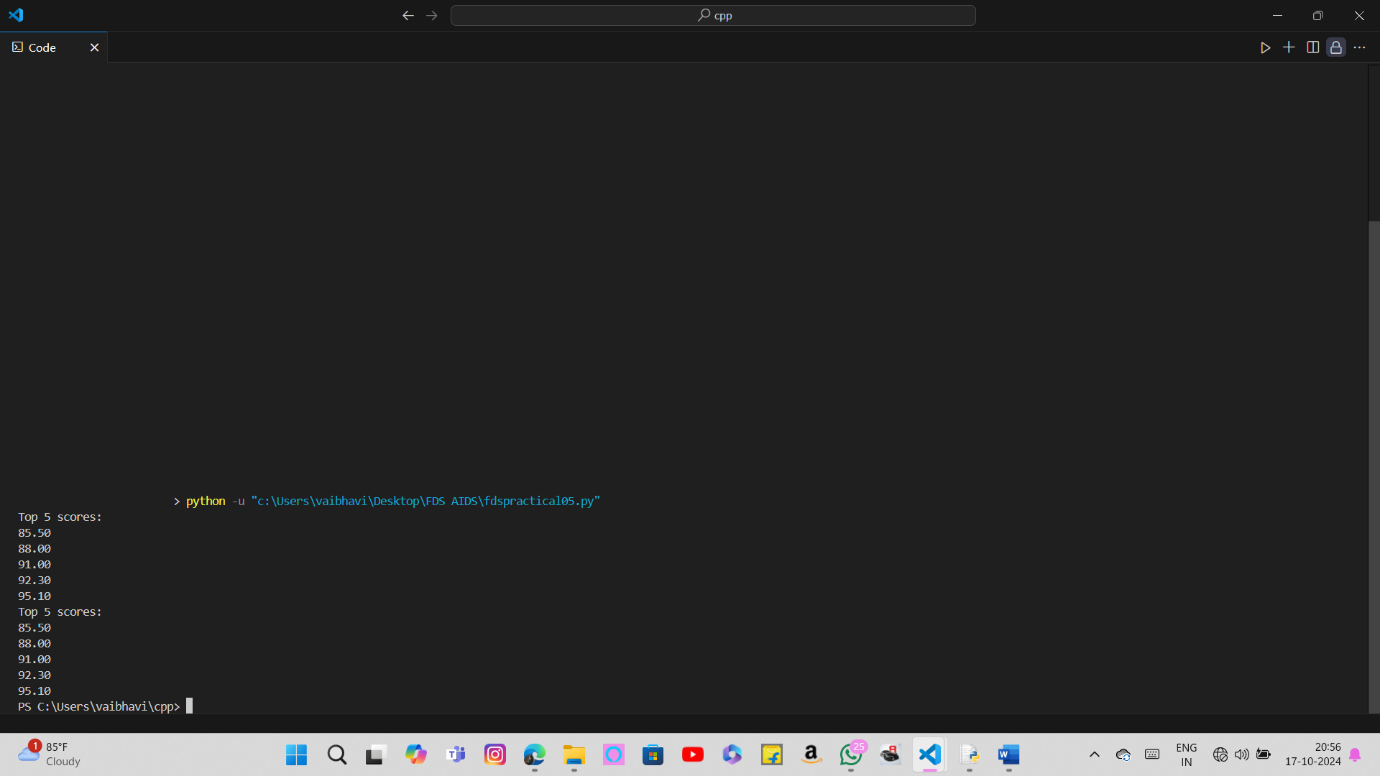
    bubble\_sort(sorted\_percentages\_bubble)

    display\_top\_scores(sorted\_percentages\_bubble)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment no.: 06

Write a Python program to store second year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using

1. Insertion sort

Shell Sort and display top five scores

Code:

def insertion\_sort(arr):

    for i in range(1, len(arr)):

        key = arr[i]

        j = i - 1

        while j >= 0 and key < arr[j]:

            arr[j + 1] = arr[j]

            j -= 1

        arr[j + 1] = key

def shell\_sort(arr):

    n = len(arr)

    gap = n // 2

    while gap > 0:

        for i in range(gap, n):

            temp = arr[i]

            j = i

            while j >= gap and arr[j - gap] > temp:

                arr[j] = arr[j - gap]

                j -= gap

            arr[j] = temp

        gap //= 2

def display\_top\_scores(arr, top\_n=5):

    print("Top {} scores:".format(top\_n))

    for score in arr[-top\_n:]:

        print(f"{score:.2f}")

def main():

    # Example percentages of students

    percentages = [88.5, 74.3, 91.2, 85.7, 66.5, 94.8, 79.0, 82.4, 89.1, 73.6]

    # Sort using Insertion Sort

    sorted\_percentages\_insertion = percentages[:]

    insertion\_sort(sorted\_percentages\_insertion)

    display\_top\_scores(sorted\_percentages\_insertion)

    # Sort using Shell Sort

    sorted\_percentages\_shell = percentages[:]

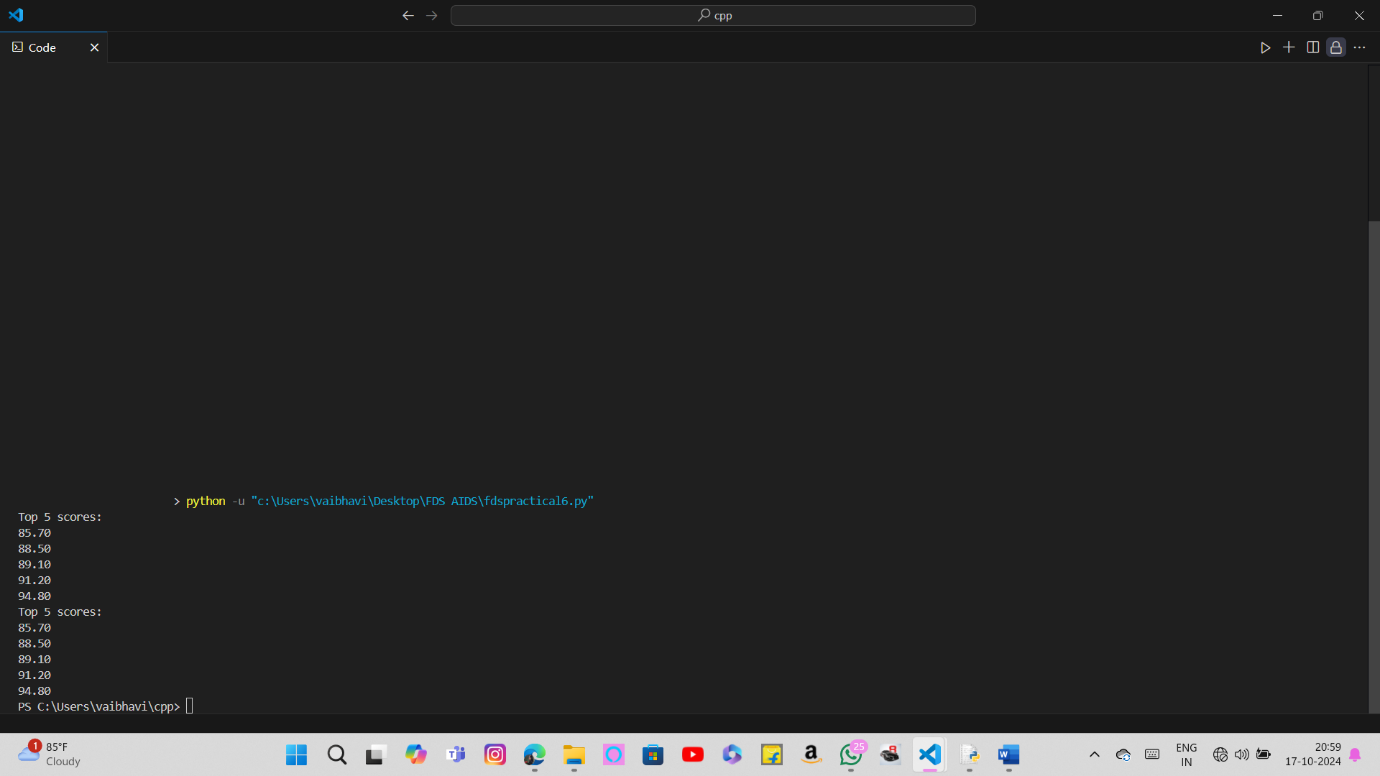
    shell\_sort(sorted\_percentages\_shell)

    display\_top\_scores(sorted\_percentages\_shell)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment no.: 07

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.

Code:

def quick\_sort(arr):

    if len(arr) <= 1:

        return arr

    else:

        pivot = arr[len(arr) // 2]

        left = [x for x in arr if x < pivot]

        middle = [x for x in arr if x == pivot]

        right = [x for x in arr if x > pivot]

        return quick\_sort(left) + middle + quick\_sort(right)

def display\_top\_scores(arr, top\_n=5):

    print("Top {} scores:".format(top\_n))

    for score in arr[-top\_n:]:

        print(f"{score:.2f}")

def main():

    # Input the number of students

    n = int(input("Enter the number of students: "))

    percentages = []

    # Input percentages from user

    for i in range(n):

        score = float(input(f"Enter percentage for student {i + 1}: "))

        percentages.append(score)

    # Sort using Quick Sort

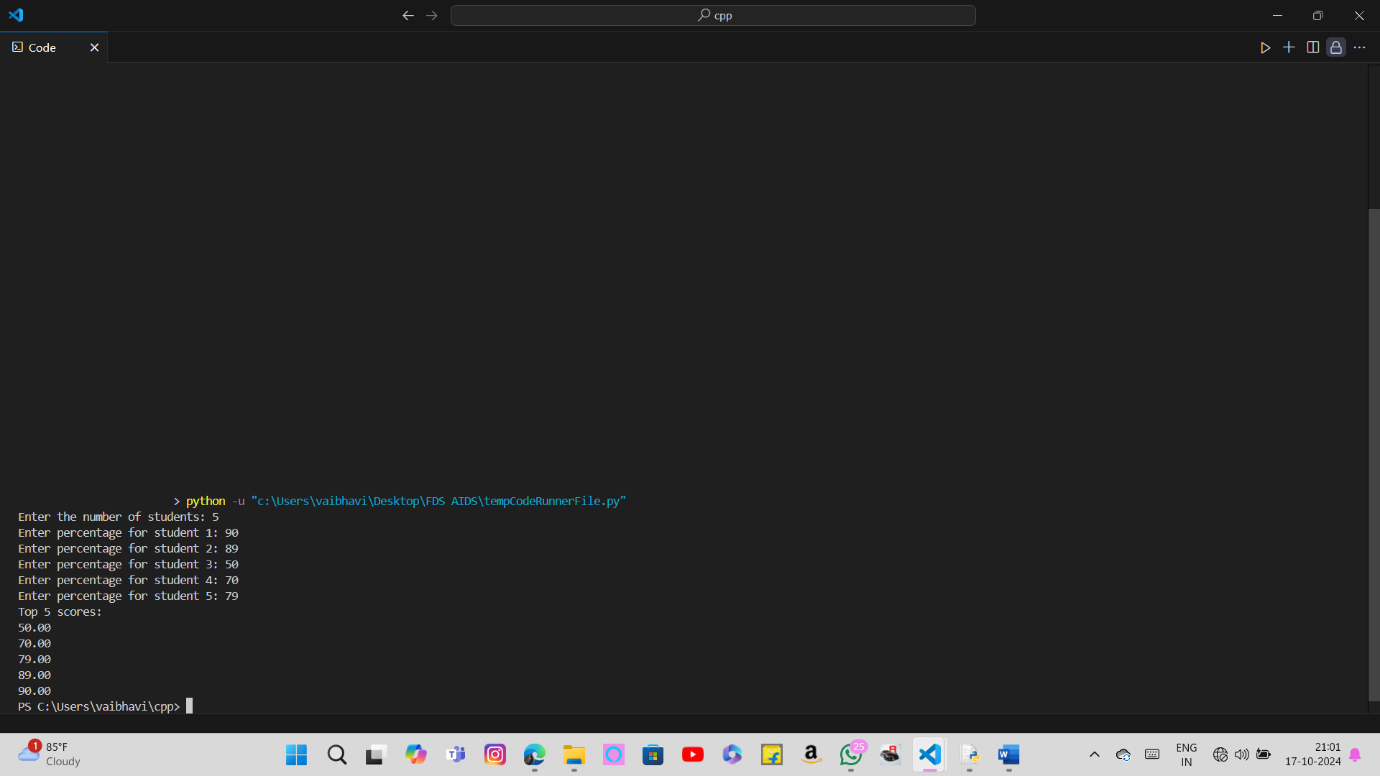
    sorted\_percentages = quick\_sort(percentages)

    display\_top\_scores(sorted\_percentages)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



Group (C)

Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment no.: 08

Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member‘s information using singly linked list. Store student PRN and Name. Write functions to:

1. Add and delete the members as well as president or even secretary.
2. Compute total number of members of club
3. Display members
4. Two linked lists exists for two divisions. Concatenate two lists.

Code:

#include<iostream>

#include<string.h>

using namespace std;

 struct node

{    int prn,rollno;

      char name[50];

      struct node \*next;

};

class info

{        node

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

        int b,c,i,j,ct;

        char a[20];

        public:

            node \*create();

            void insertp();

            void insertm();

            void delm();

            void delp();

            void dels();

            void display();

            void count();

            void reverse();

            void rev(node \*p);

            void concat();

}  ;

node \*info::create()

{   node \*p=new(struct node);

     cout<<"enter name of student ";

     cin>>a;

     strcpy(p->name,a);

       cout<<"\n  enter prn no. of student \n";

     cin>>b;

     p->prn=b;

     cout<<"enter student rollno";

     cin>>c;

     p->rollno=c;

     p->next=NULL;

     return  p;

  }

  void info::insertm()

  {

       node \*p=create();

     if(head==NULL)

     {    head=p;

     }

    else

    {      temp=head;

          while(temp->next!=NULL)

          {    temp=temp->next;   }

              temp->next=p;

     }

   }

     void info::insertp()

  {

       node \*p=create();

     if(head==NULL)

     {    head=p;

     }

    else

    {      temp=head;

            head=p;

              head->next=temp->next;

     }

   }

   void info::display()

   {          if(head==NULL)

               {    cout<<"linklist is empty";

                }

              else

              {

                temp=head;

                          cout<<"     prn     rolln0    NAME   \n";

                          while(temp->next!=NULL)

                          {     cout<<"    \n"<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;

                                temp=temp->next;

                          }

                         cout<<"    "<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;

               }

  }

  void info::delm()

  {  int m,f=0;

     cout<<"\n enter the prn no. of student whose data you want to delete";

      cin>>m;

      temp=head;

      while(temp->next!=NULL)

      {

           if(temp->prn==m)

            {           s->next=temp->next;

                         delete(temp);        f=1;

            }

            s=temp;

            temp=temp->next;

       }      if(f==0)

             {   cout<<"\n sorry memeber not deleted ";   }

   }

   void info::delp()

  {     temp=head;

      head=head->next;

         delete(temp);

      }

      void info::dels()

  {

      temp=head;

      while(temp->next!=NULL)

      {   s=temp;

      temp=temp->next;

      }     s->next=NULL;

         delete(temp);

   }

   void info::count()

   {      temp=head;    ct=0;

          while(temp->next!=NULL)

          {    temp=temp->next; ct++;   }

             ct++;

             cout<<"  Count of members is:"<<ct;

     }

     void info::reverse()

     {     rev(head);   }

     void info::rev(node \*temp)

     {     if(temp==NULL)

           { return;   }

           else

           {   rev(temp->next); }

           cout<<"    "<<temp->prn<<"    "<<temp->rollno<<"    "<<temp->name;

     }

     void info::concat()

    {  int k,j;

       cout<<"enter no. of members in list1";

       cin>>k;

        head=NULL;

       for(i=0;i<k;i++)

       { insertm();

         head1=head;

       } head=NULL;

      cout<<"enter no. of members in list2";

       cin>>j;

       for(i=0;i<j;i++)

       { insertm();

         head2=head;

       } head=NULL;

        temp1=head1;

      while(temp1->next!=NULL)

      {   temp1=temp1->next;   }

            temp1->next=head2;

                              temp2=head1;

                             cout<<"     prn     rolln0    NAME   \n";

                          while(temp2->next!=NULL)

                          {

                             cout<<"\n    "<<temp2->prn<<"    "<<temp2->rollno<<"      "<<temp2->name<<"\n";;

                             temp2=temp2->next;

                          }

                             cout<<"\n    "<<temp2->prn<<"    "<<temp2->rollno<<"      "<<temp2->name<<"\n";

     }

  int main()

  { info s;

  int i;

          char ch;

       do{

          cout<<"\n choice the options";

          cout<<"\n  1. To insert president   ";

          cout<<"\n  2. To insert member   ";

          cout<<"\n  3. To insert secretary ";

          cout<<"\n  4. To delete president   ";

          cout<<"\n  5. To delete member  ";

          cout<<"\n  6. To delete secretary ";

          cout<<"\n  7. To display data   ";

          cout<<"\n  8. Count of members";

          cout<<"\n  9. To display reverse of string ";

          cout<<"\n  10.To concatenate two strings ";

          cin>>i;

         switch(i)

         {        case 1:   s.insertp();

                                  break;

                  case 2:   s.insertm();

                                  break;

                  case 3:   s.insertm();

                                  break;

                  case 4:   s.delp();

                                  break;

                  case 5:   s.delm();

                                  break;

                  case 6:   s.dels();

                                  break;

                  case 7:   s.display();

                                  break;

                  case 8:   s.count();

                                  break;

                  case 9:   s.reverse();

                                  break;

                  case 10:  s.concat();

                                  break;

                  default:  cout<<"\n unknown choice";

          }

            cout<<"\n do you want to continue enter y/Y \n";

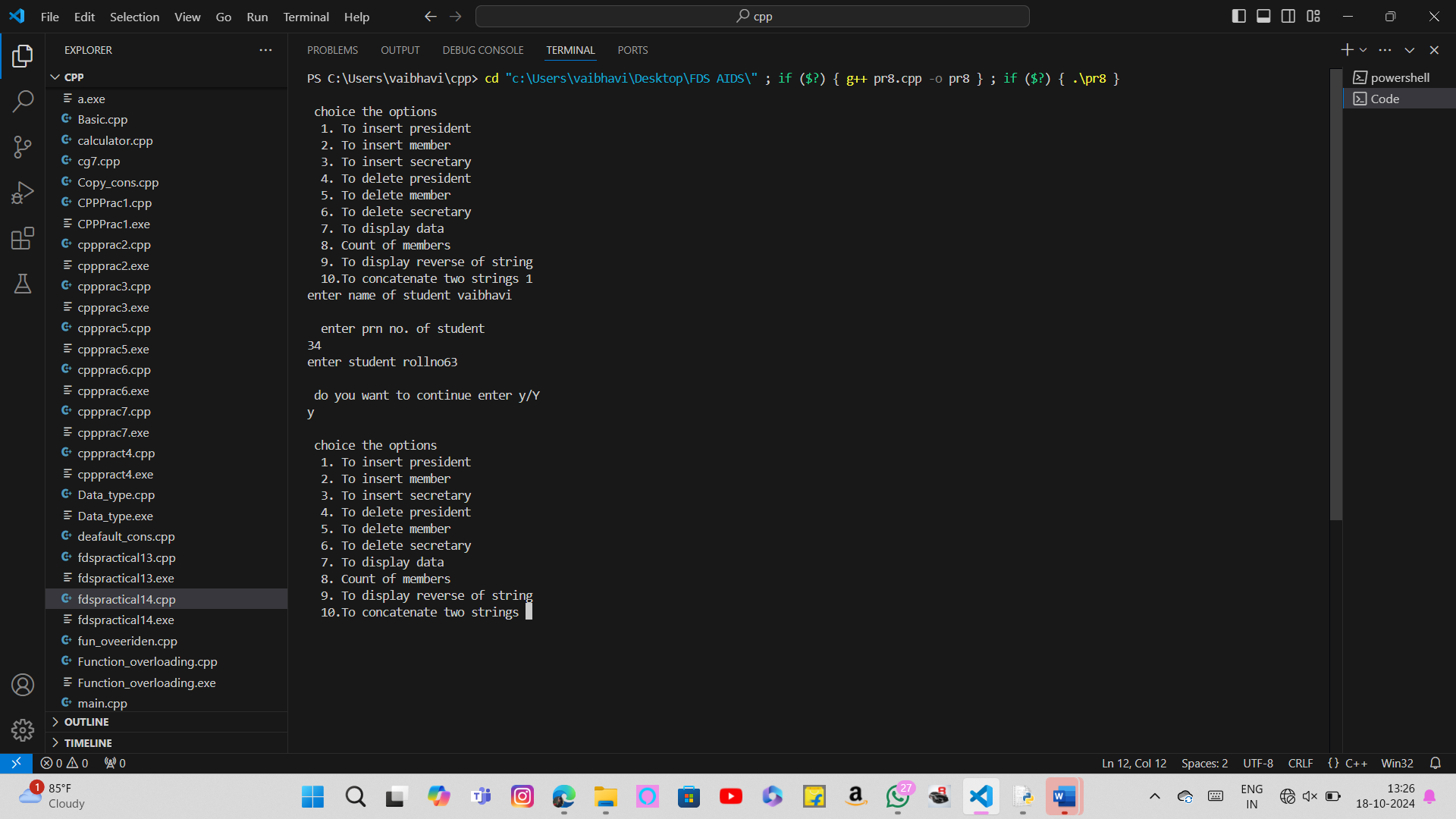
            cin>>ch;

       }while(ch=='y'||ch=='Y');

   return 0;

 }

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 09

The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use array to store pointers (Head pointer) to each row. On demand

1. The list of available seats is to be displayed
2. The seats are to be booked

The booking can be cancelled.

Code: //Name : Shubham Ghodke

//Roll No:56

//Practical 9-GROUP [C-C2]

#include <iostream>

#include<stdlib.h>

using namespace std;

class node

{ public:

node\* next;

node\* prev;

int seat;

string id;

int status;

};

class cinemax

{

public:

node\* head,\* tail ,\* temp;

cinemax()

{

head=NULL;

}

void create\_list();

void display();

void book();

void cancel();

void avail();

};

void cinemax::create\_list()

{

int i=1;

temp=new node;

temp->seat=1;

temp->status=0;

temp->id="null";

tail=head=temp;

for(int i=2;i<=70;i++)

{

node \*p;

p= new node;

p->seat=i;

p->status=0;

p->id="null";

tail->next=p;

p->prev=tail;

tail=p;

tail->next=head;

head->prev=tail;

}

}

void cinemax::display()

{

{ int r=1;

node\* temp;

temp=head;

int count=0;

cout<<"\n------------------------------------------------------------------------------------\n";

cout<<" Screen";

this ;"\n";

cout<<"------------------------------------------------------------------------------------\n";

while(temp->next!=head)

{

if(temp->seat/10==0)

cout<<"S0"<<temp->seat<<" :";

else

cout<<"S"<<temp->seat<<" :";

if(temp->status==0)

cout<<"|\_| ";

else

cout<<"|B| ";

count++;

if(count%7==0)

{

cout<<endl;

r++;

}

temp=temp->next;

}

cout<<"S"<<temp->seat<<" :";

if(temp->status==0)

cout<<"|\_| ";

else

cout<<"|B| ";

}

}

void cinemax::book()

{ int x;

string y;

label:

cout<<"\n\n\nEnter seat number to be booked\n";

cin>>x;

cout<<"Enter your ID number\n";

cin>>y;

if(x<1||x>70)

{

cout<<"Enter correct seat number to book (1-70)\n";

goto label;

}

node \*temp;

temp=new node;

temp=head;

while(temp->seat!=x)

{

temp=temp->next;

}

if(temp->status==1)

cout<<"Seat already booked!\n";

else{

temp->status=1;

temp->id=y;

cout<<"Seat "<<x<<" booked!\n";

}

}

void cinemax::cancel()

{

int x;

string y;

label1:

cout<<"Enter seat number to cancel booking\n";

cin>>x;

cout<<"Enter you ID\n";

cin>>y;

if(x<1||x>70)

{

cout<<"Enter correct seat number to cancel (1-70)\n";

goto label1;

}

node \*temp;

temp=new node;

temp=head;

while(temp->seat!=x)

{

temp=temp->next;

}

if(temp->status==0)

{

cout<<"Seat not booked yet!!\n";

}

else

{

if(temp->id==y)

{

temp->status=0;

cout<<"Seat Cancelled!\n";

}

else

cout<<"Wrong User ID !!! Seat cannot be cancelled!!!\n";

}

}

void cinemax::avail()

{

int r=1;

node\* temp;

temp=head;

int count=0;

cout<<"\n\n\n\n";

cout<<"\n------------------------------------------------------------------------------------\n";

cout<<" Screen this way \n";

cout<<"------------------------------------------------------------------------------------\n";

while(temp->next!=head)

{

{

if(temp->seat/10==0)

cout<<"S0"<<temp->seat<<" :";

else

cout<<"S"<<temp->seat<<" :";

if(temp->status==0)

cout<<"|\_| ";

else if(temp->status==1)

cout<<" ";

count++;

if(count%7==0)

{

cout<<endl;

}

}

temp=temp->next;

}

if(temp->status==0)

{

cout<<"S"<<temp->seat<<" :";

if(temp->status==0)

cout<<"|\_| ";

}

}

int main()

{ cinemax obj;

obj.create\_list();

int ch;

char c='y';

while(c=='y')

{ obj.display();

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<" CINEMAX MOVIE THEATRE\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\nEnter Choice\n1.Current SeatStatus\n2.Book Seat \n3.Available Seat\n4.CancelSeat\n";

cin>>ch;

switch(ch)

{

case 1:obj.display();

break;

case 2: obj.book();

break;

case 3:obj.avail();

break;

case 4: obj.cancel();

break;

default: cout<<"Wrong choice input\n";

}

cout<<"\nDo you want to perform any other operation : (y/n)\n";

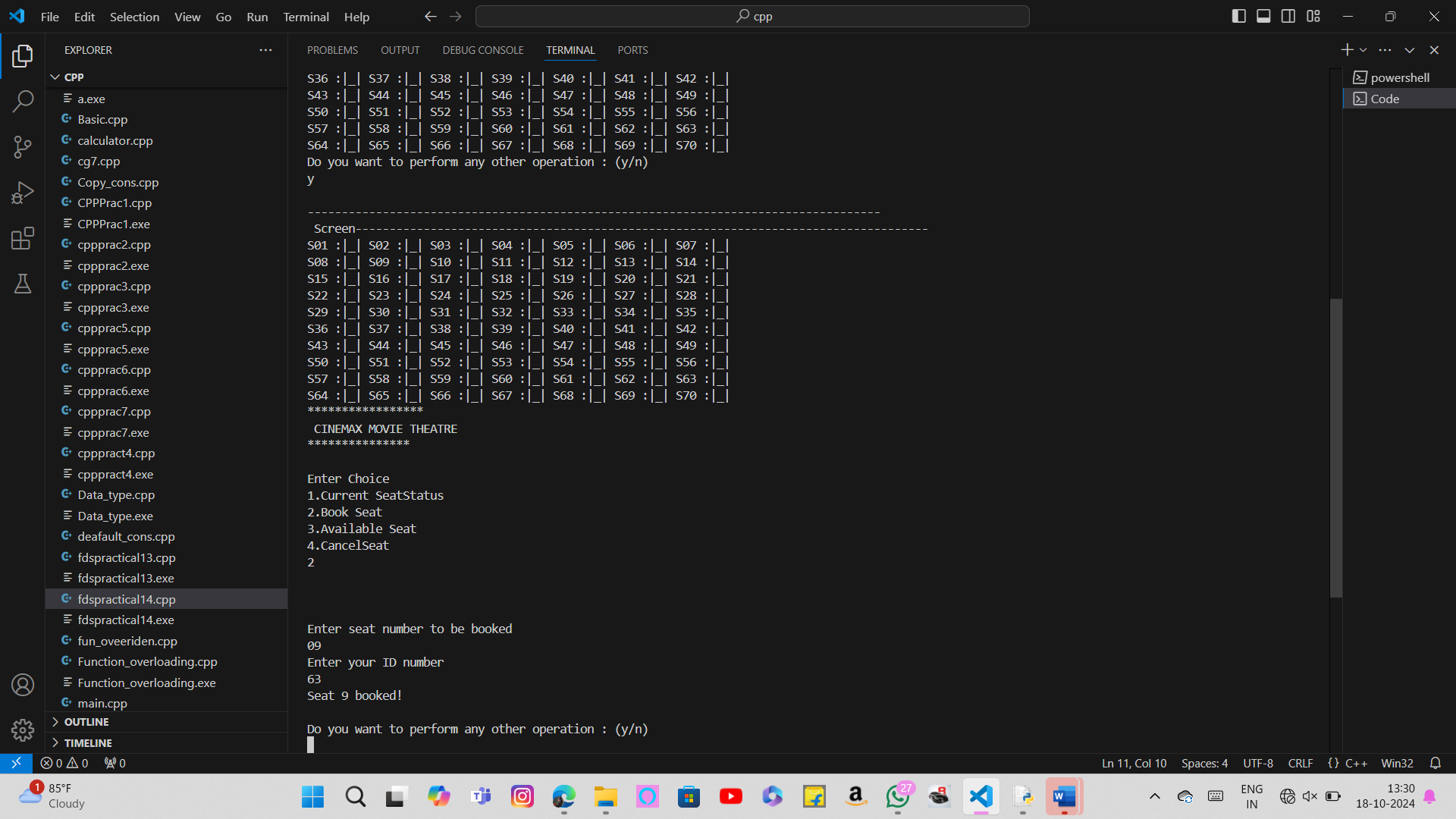
cin>>c;

}

return 0;

}

Output:



Group (D)

Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 10

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 danisina 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 functions-

1. To print original string followed by reversed string using stack

To check whether given string is palindrome or not.

Code:

#include <iostream>

#include <stack>

#include <cctype>

#include <string>

void printOriginalAndReversed(const std::string& input) {

    std::stack<char> charStack;

    // Push characters onto the stack

    for (char c : input)

     {

        if (std::isalnum(c))

         { // Check if the character is alphanumeric

            charStack.push(std::tolower(c)); // Push lowercase characters

        }

    }

    // Print original string

    std::cout << "Original string: " << input << std::endl;

    // Print reversed string

    std::cout << "Reversed string: ";

    while (!charStack.empty()) {

        std::cout << charStack.top();

        charStack.pop();

    }

    std::cout << std::endl;

}

bool isPalindrome(const std::string& input) {

    std::stack<char> charStack;

    std::string cleanedInput;

    // Clean the input and push characters onto the stack

    for (char c : input) {

        if (std::isalnum(c)) {

            char lowerChar = std::tolower(c);

            cleanedInput += lowerChar; // Store cleaned input

            charStack.push(lowerChar);

        }

    }

    // Check for palindrome

    for (char c : cleanedInput) {

        if (c != charStack.top()) {

            return false; // Not a palindrome

        }

        charStack.pop();

    }

    return true; // It is a palindrome

}

int main() {

    std::string input;

    std::cout << "Enter a string: ";

    std::getline(std::cin, input);

    printOriginalAndReversed(input);

    if (isPalindrome(input)) {

        std::cout << "The string is a palindrome." << std::endl;

    } else {

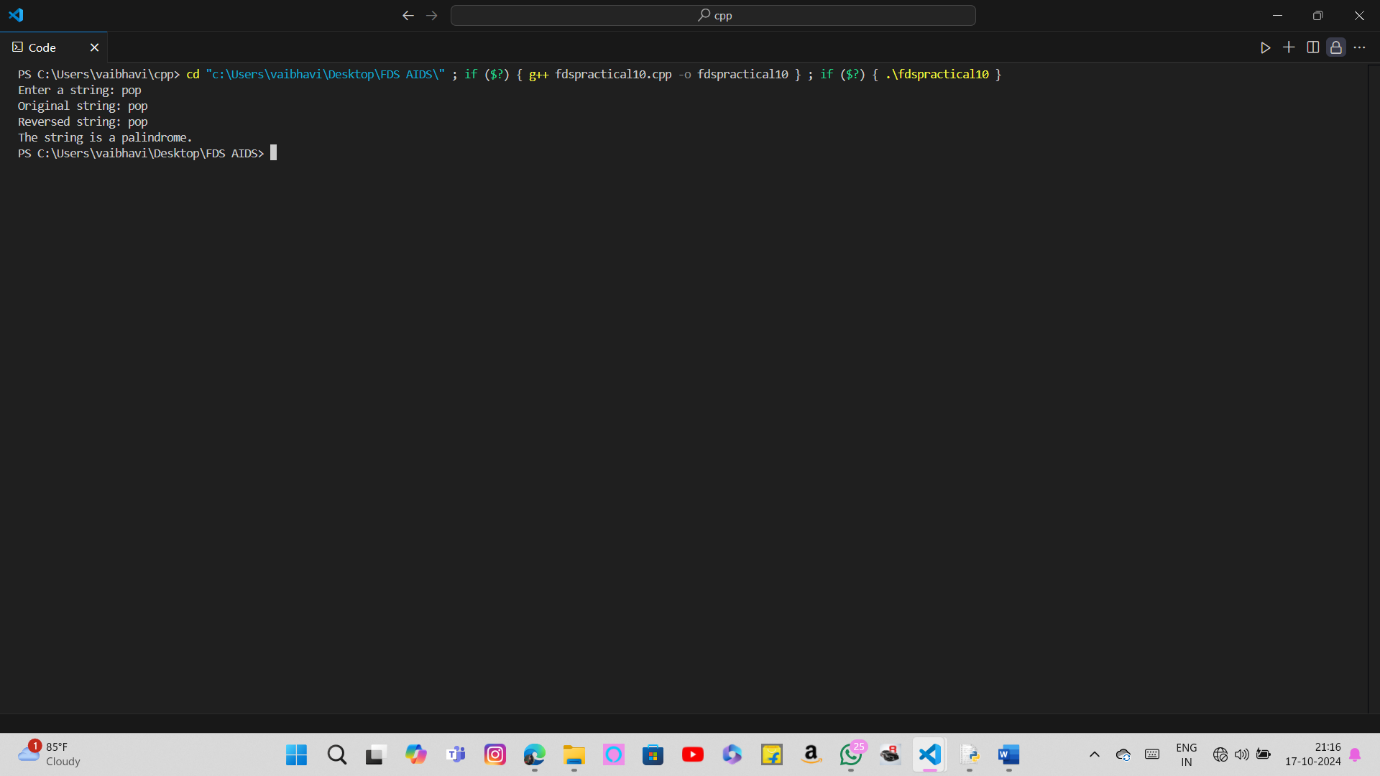
        std::cout << "The string is not a palindrome." << std::endl;

    }

    return 0;

}

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 11

Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions:

1. Operands and operator, both must be single character.
2. Input Postfix expression must be in a desired format.

Only '+', '-', '\*' and '/ ' operators are expected.

Code:

#include <iostream>

#include <stack>

#include <string>

#include <cctype>

int precedence(char op) {

    if (op == '+' || op == '-') return 1;

    if (op == '\*' || op == '/') return 2;

    return 0;

}

bool isOperator(char c) {

    return c == '+' || c == '-' || c == '\*' || c == '/';

}

// Function to convert infix expression to postfix

std::string infixToPostfix(const std::string& infix) {

    std::stack<char> opStack;

    std::string postfix;

    for (char c : infix) {

        if (std::isspace(c)) continue; // Ignore spaces

        if (std::isalnum(c)) {

            postfix += c; // Append operand to postfix

        } else if (isOperator(c)) {

            while (!opStack.empty() && precedence(opStack.top()) >= precedence(c)) {

                postfix += opStack.top();

                opStack.pop();

            }

            opStack.push(c);

        }

    }

    // Pop all the operators from the stack

    while (!opStack.empty()) {

        postfix += opStack.top();

        opStack.pop();

    }

    return postfix;

}

// Function to evaluate postfix expression

int evaluatePostfix(const std::string& postfix) {

    std::stack<int> valueStack;

    for (char c : postfix) {

        if (std::isdigit(c)) {

            valueStack.push(c - '0'); // Convert char to int

        } else if (isOperator(c)) {

            int rightOperand = valueStack.top(); valueStack.pop();

            int leftOperand = valueStack.top(); valueStack.pop();

            switch (c) {

                case '+':

                    valueStack.push(leftOperand + rightOperand);

                    break;

                case '-':

                    valueStack.push(leftOperand - rightOperand);

                    break;

                case '\*':

                    valueStack.push(leftOperand \* rightOperand);

                    break;

                case '/':

                    valueStack.push(leftOperand / rightOperand);

                    break;

            }

        }

    }

    return valueStack.top(); // Final result

}

int main() {

    std::string infix;

    std::cout << "Enter an infix expression (single characters only): ";

    std::getline(std::cin, infix);

    // Convert infix to postfix

    std::string postfix = infixToPostfix(infix);

    std::cout << "Postfix expression: " << postfix << std::endl;

    // Evaluate the postfix expression

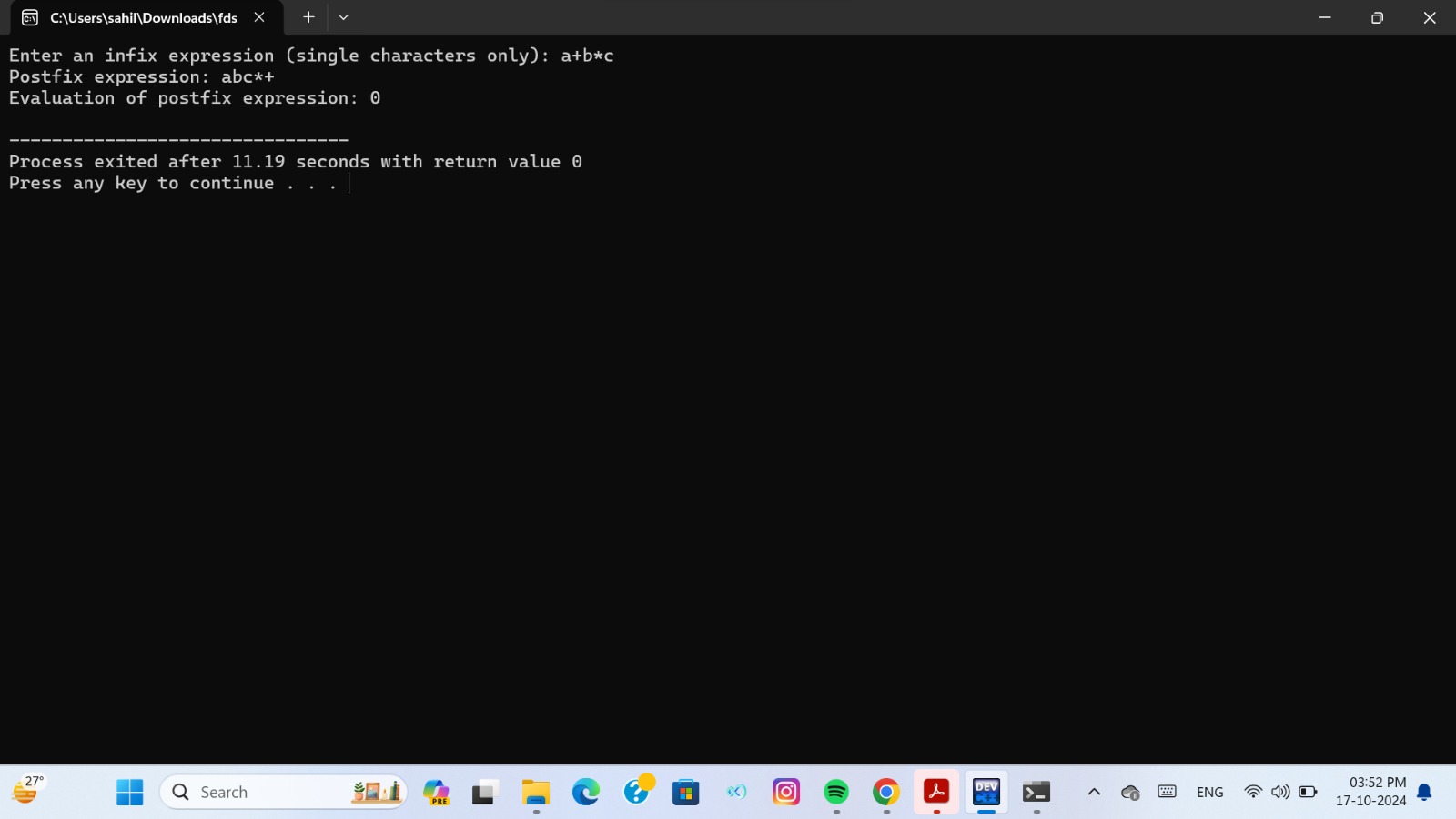
    int result = evaluatePostfix(postfix);

    std::cout << "Evaluation of postfix expression: " << result << std::endl;

    return 0;

}

Output:



Group(E)

Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 12

E-29 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.

Code:

#include <iostream>

#define MAX 10

using namespace std;

struct queue

{       int data[MAX];

    int front,rear;

};

class Queue

{    struct queue q;

   public:

      Queue(){q.front=q.rear=-1;}

      int isempty();

      int isfull();

      void enqueue(int);

      int delqueue();

      void display();

};

int Queue::isempty()

{

    return(q.front==q.rear)?1:0;

}

int Queue::isfull()

{    return(q.rear==MAX-1)?1:0;}

void Queue::enqueue(int x)

{q.data[++q.rear]=x;}

int Queue::delqueue()

{return q.data[++q.front];}

void Queue::display()

{   int i;

    cout<<"\n";

    for(i=q.front+1;i<=q.rear;i++)

         cout<<q.data[i]<<" ";

}

int main()

{      Queue obj;

    int ch,x;

    do{    cout<<"\n 1.Insert Job\n 2.Delete Job\n 3.Display\n 4.Exit\n Enter your choice : ";

           cin>>ch;

    switch(ch)

    {  case 1: if (!obj.isfull())

           {   cout<<"\n Enter data : \n";

            cin>>x;

            obj.enqueue(x);

            cout<<endl;

           }

              else

              cout<< "Queue is overflow!!!\n\n";

               break;

       case 2: if(!obj.isempty())

                cout<<"\n Deleted Element = "<<obj.delqueue()<<endl;

            else

            {   cout<<"\n Queue is underflow!!!\n\n";  }

            cout<<"\nRemaining Jobs : \n";

            obj.display();

               break;

      case 3: if (!obj.isempty())

            {  cout<<"\n Queue contains : \n";

               obj.display();

            }

            else

                 cout<<"\n Queue is empty!!!\n\n";

           break;

      case 4: cout<<"\n Exiting Program.....";

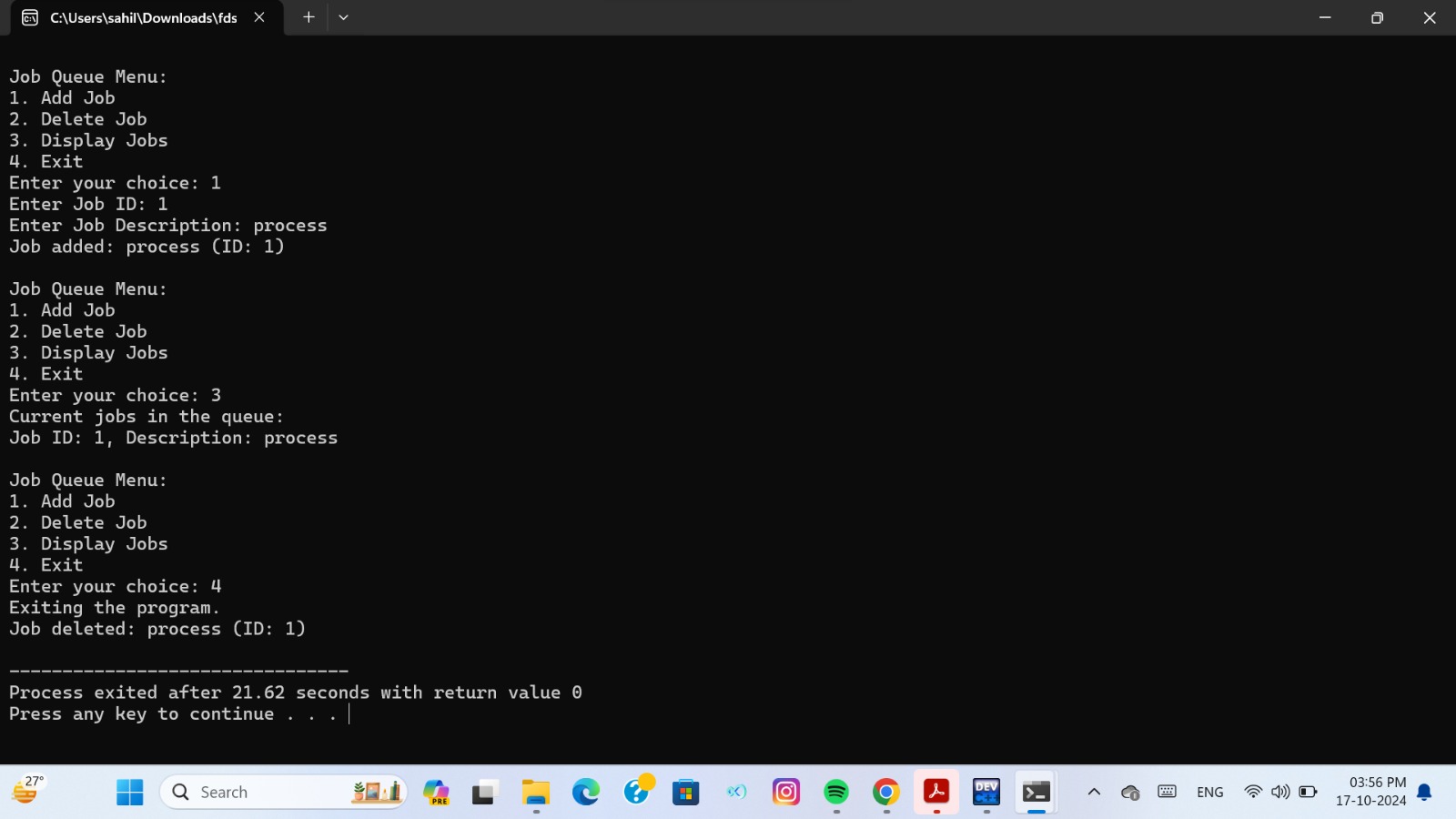
        }

      }while(ch!=4);

return 0;

}

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 13

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

Code:

#include<iostream>

#include<stdio.h>

#define MAX 10

using namespace std;

struct que

{

    int arr[MAX];

    int front,rear;

};

void init(struct que \*q)

{

    q->front=-1;

    q->rear=-1;

}

void print(struct que q)

{

    int i;

    i=q.front;

    while(i!=q.rear)

    {

        cout<<"\t"<<q.arr[i];

        i=(i+1)%MAX;

    }

    cout<<"\t"<<q.arr[q.rear];

}

int isempty(struct que q)

{

    return q.rear==-1?1:0;

}

int isfull(struct que q)

{

    return (q.rear+1)%MAX==q.front?1:0;

}

void addf(struct que \*q,int data)

{

    if(isempty(\*q))

    {

        q->front=q->rear=0;

        q->arr[q->front]=data;

    }

    else

    {

        q->front=(q->front-1+MAX)%MAX;

        q->arr[q->front]=data;

    }

}

void addr(struct que \*q,int data)

{

    if(isempty(\*q))

    {

        q->front=q->rear=0;

        q->arr[q->rear]=data;

    }

    else

    {

        q->rear=(q->rear+1)%MAX;

        q->arr[q->rear]=data;

    }

}

int delf(struct que \*q)

{

    int data1;

    data1=q->arr[q->front];

    if(q->front==q->rear)

        init(q);

    else

        q->front=(q->front+1)%MAX;

    return data1;

}

int delr(struct que \*q)

{

    int data1;

    data1=q->arr[q->rear];

    if(q->front==q->rear)

        init(q);

    else

        q->rear=(q->rear-1+MAX)%MAX;

    return data1;

}

int main()

{

    struct que q;

    int data,ch;

    init(&q);

    while(ch!=6)

    {

        cout<<"\t\n1.Insert front"

                "\t\n2.Insert rear"

                "\t\n3.Delete front"

                "\t\n4.Delete rear"

                "\t\n5.Print"

                "\t\n6.Exit";

         cout<<"\nEnter your choice : ";

        cin>>ch;

        switch(ch)

        {

           case 1:

              cout<<"\nEnter data to insert front : ";

              cin>>data;

              addf(&q,data);

              break;

           case 2:

               cout<<"\nEnter the data to insert rear : ";

               cin>>data;

               addr(&q,data);

               break;

           case 3:

               if(isempty(q))

                   cout<<"\nDequeue is empty!!!";

               else

               {

                   data=delf(&q);

                   cout<<"\nDeleted data is : "<<data;

               }

               break;

           case 4:

               if(isempty(q))

                   cout<<"\nDequeue is empty!!!";

               else

               {

                   data=delr(&q);

                   cout<<"\nDeleted data is : "<<data;

               }

               break;

           case 5:

                if(isempty(q))

                    cout<<"\nDequeue is empty!!!";

                else

                {

                    cout<<"\nDequeue elements are : ";

                    print(q);

                }

                break;

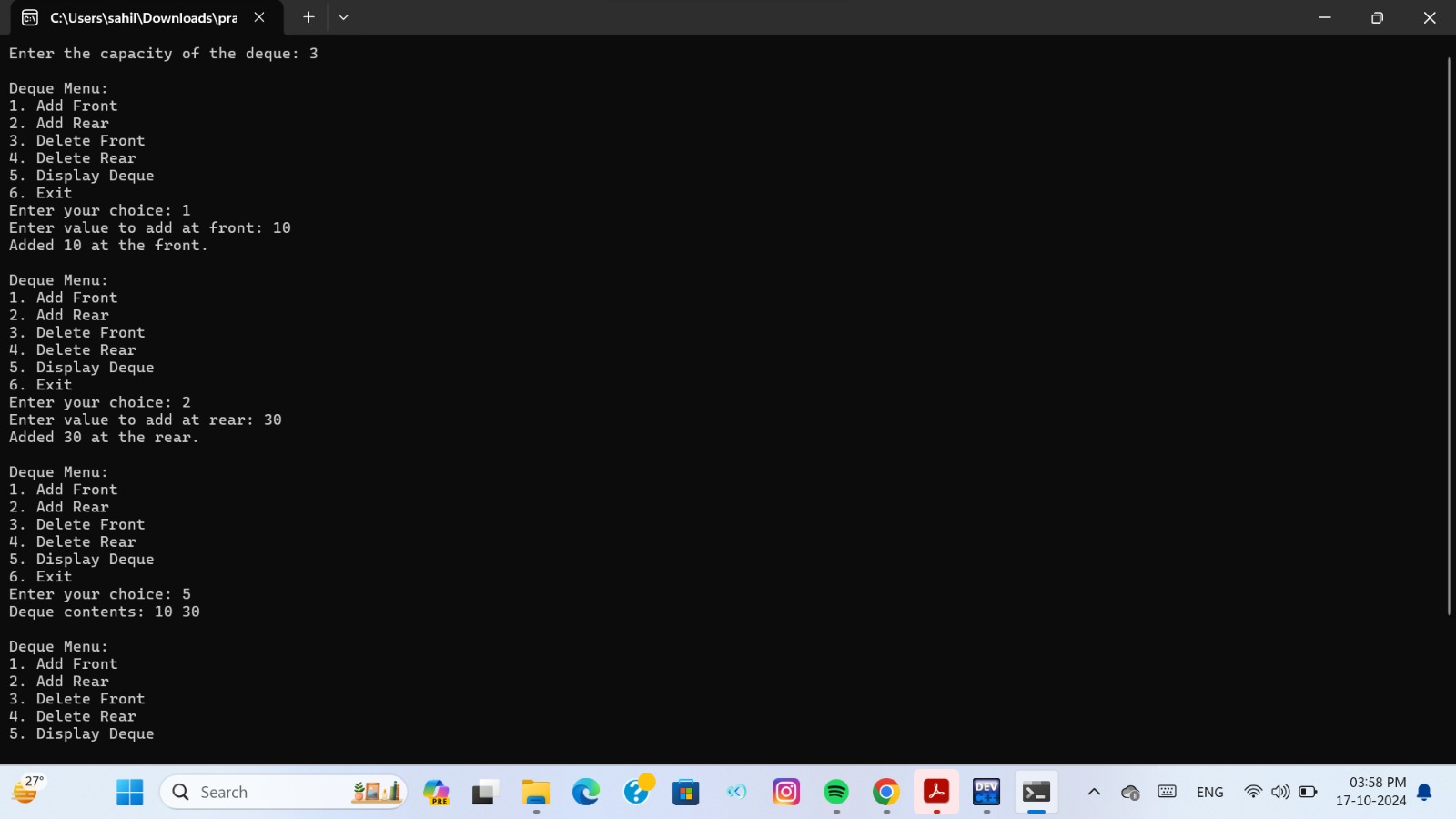
        }

    }

    return 0;

}

Output:



Name : Tejaswini Sachin Shewale

Roll No.:37

Assignment No.: 14

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.

Code:

#include<iostream>

#include<cstdlib>

using namespace std;

class pizza {

    int front, rear, q[5];

public:

    pizza() {

        front = -1;

        rear = -1;

    }

    int isfull() {

        if ((front == 0 && rear == 4) || front == (rear + 1) % 5) {

            return 1;

        }

        else {

            return 0;

        }

    }

    int isempty() {

        if (front == -1 && rear == -1) {

            return 1;

        }

        else {

            return 0;

        }

    }

    void add() {

        if (isfull() == 0) {

            cout << "\n Enter the Pizza ID: ";

            if (front == -1 && rear == -1) {

                front = 0;

                rear = 0;

                cin >> q[rear];

            }

            else {

                rear = (rear + 1) % 5;

                cin >> q[rear];

            }

            char c;

            cout << " Do you want to add another order ? ";

            cin >> c;

            if (c == 'y' || c == 'Y') add();

        }

        else {

            cout << "\n Orders are full ";

        }

    }

    void serve() {

        if (isempty() == 0) {

            if (front == rear) {

                cout << "\n Order served is : " << q[front];

                front = -1;

                rear = -1;

            }

            else {

                cout << "\n Order served is : " << q[front];

                front = (front + 1) % 5;

            }

        }

        else {

            cout << "\n Orders are empty ";

        }

    }

    void display() {

        if (isempty() == 0) {

            if (rear >= front) {

                for (int i = front; i <= rear; i++) {

                    cout << q[i] << " <- ";

                }

            }

            else {

                for (int i = front; i < 5; i++) {

                    cout << q[i] << " <- ";

                }

                for (int i = 0; i <= rear; i++) {

                    cout << q[i] << " <- ";

                }

            }

            cout << q[rear];

        }

        else {

            cout << "\n Orders are empty";

        }

    }

    void check() {

        char ch1;

        int ch;

        do {

            cout << "\n\n \* \* \* \* PIZZA PARLOUR \* \* \* \* \n\n";

            cout << "\n 1. Add a Pizza \n 2. Display the Orders \n 3. Serve a pizza \n 4. Exit \n Enter your choice : ";

            cin >> ch;

            switch (ch) {

            case 1:

                add();

                break;

            case 2:

                display();

                break;

            case 3:

                serve();

                break;

            case 4:

                exit(0);

                break;

            default:

                cout << "Invalid choice ";

            }

            cout << "\n Do you want to continue? ";

            cin >> ch1;

        } while (ch1 == 'y' || ch1 == 'Y');

    }

};

int main() {

    pizza p1;

    p1.check();

    return 0;

}

Output:

