NAME🡪YASH BHAVATHANKR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

#1st array of ckt bdm ftball grpA=[]

grpB=[] grpC=[]

q=int(input("\nEnter number of student in cricket team:")) for i in range(0,q):

p=input("\nEnter Name::") grpA.append(p)

q=int(input("\nEnter number of student in badminton team:")) for i in range(0,q):

p=input("\nEnter Name::") grpB.append(p)

q=int(input("\nEnter number of student in football team:")) for i in range(0,q):

p=input("\nEnter Name::") grpC.append(p)

c1=[ ]

c2=[ ]

c3=[ ]

c4=[ ]

c=[ ]

def f1():

for i in grpA:

for j in grpB

if i==j:

c1.append(i)

return c1 def f2():

for i in grpA:

if i not in grpB:

c2.append(i)

for j in grpB:

if j not in grpA:

c2.append(j)

return c2 def f3():

s=[ ]

for i in grpC:

if i not in grpA:

s.append(i)

for j in s:

if j not in grpB:

c3.append(j)

return c3 def f4():

c=[ ]

for i in grpA:

if i not in grpB:

c.append(i)

for j in grpC:

if j not in grpC:

c4.append(j)

for k in grpC:

if k not in grpB:

c4.append(k)

return c4

while(1):

print("1] students who play cricket and badminton::\n2]students who play either cricket or badminton not both::\n3]students who play neither cricket nor badminton::\n4]students who play cricket or football but not badminton::\n ")

c=int(input("enter the following choices"))

if (c==1):

print(f1()) elif(c==2):

print(f2()) elif(c==3):

print(f3()) elif(c==4):

print(f4())

else:

print("enter valid choice")

NAME🡪YASH BHAVATHANKR

ROLLNO-205A027

DIVISION🡪SE-1

def s1():

str1 = input("Enter the main string :") long = []

s = str1.split() for i in s:

if(len(i) > len(long)): long = i

print("Word with longest length is operations having length:", long)

def s2():

str1 = input("Enter the string: ") d = input("Enter the character :") e = str1.count(d)

print("Frequency of occurrence of character(s) in string (", d, ") is", e)

def s3():

str1 = input("Enter the string to be checked : ") a = str1[::-1]

if(a == str1):

print(a, " string is an palindrome string") else:

print(a, " string is not a palindrome string")

def s4():

str1 = input("Enter the main string : ")

c = input("Enter the sub string to check : ") b = str1.find(c)

print("Main String :", str1); print("Substring String :", c); print("Substring string found at index", b)

def s5():

str1 = input("Enter the main string :") words = str1.split()

words.sort() occ=1

prev = words[0]

for i in range(1, len(words)): if(words[i] != prev):

print(prev, ":", occ) prev = words[i] occ = 1

else:

occ += 1 print(prev, ":", occ)

while(True):

print("1 : Display word with longest length")

print("2 : Determine the frequency of occurrence of particular character in the string") print("3 : Check whether given string is palindrome or not")

print("4 : Display index of first appearance of the substring") print("5 : Count the occurrences of each word in a given string") print("6 : Exit")

func = int(input("Enter your choice: ")) if(func==1):

p=s1() elif(func==2):

q=s2() elif(func==3):

r=s3() elif(func==4):

s=s4() elif(func==5):

t=s5() else:

break

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

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()

NAME🡪YASH BHAVATHANKAR

ROLL-NO🡪205A027

DIVISION🡪SE-1

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

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()

NANE🡪YASH BHAVATHANKAR

ROLL-NO🡪205A027

DIVISION🡪SE-1

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 Selection\_sort(A) : n = len(A)

for pos in range(n-1): min\_ind = pos

for i in range(pos + 1, n) : if(A[i] < A[min\_ind]) :

min\_ind = i temp = A[pos] A[pos] = A[min\_ind]

A[min\_ind] = temp

def Bubble\_sort(A) :

n = len(A)

for Pass in range(1,n) :

for i in range(n-Pass) :

if(A[i] < A[i+1]) :

temp = A[i] A[i] = A[i+1]

A[i+1] = temp

def Main() :

A = []

while True :

print ("\t1 : Accept & Display the FE Marks") print ("\t2 : Selection Sort Ascending order")

print ("\t3 : Bubble sort Descending order and display top five scores") print ("\t4 : Exit")

ch = int(input("Enter your choice : ")) if (ch == 4):

print ("End of Program") quit()

elif (ch==1): accept\_array(A) display\_array(A)

elif (ch==2):

print("Marks before sorting")

display\_array(A) Selection\_sort(A) print("Marks after sorting") display\_array(A)

elif (ch==3):

print("Marks before sorting") display\_array(A) Bubble\_sort(A)

print("Marks after sorting") display\_array(A)

if(len(A) >= 5) :

print("Top Five Scores : ") for i in range(5) :

print("\t%.2f"%A[i]) else :

print("Top Scorers : ") for i in range(len(A)) : print("\t%.2f"%A[i])

else :

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

Main()

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

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()

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISIO🡪SE-1

#include<iostream> #include <string.h> #include<stdlib.h>

using namespace std;

class SNode // NODE CLASS

{

private:

char p[10],n[10];

SNode \*next;

public:

SNode();

char\* getPrn(); char\* getName();

void setData(char val[],char name[]);

SNode \*getNext();

void setNext(SNode \*n);

};

SNode::SNode(){ //DEFAULT CONSTRUCTOR

strcpy(p,"");

strcpy(n,""); next = NULL;

}

char\* SNode::getPrn(){ return p;

}

char\* SNode::getName(){ return n;

}

void SNode::setData(char val[],char na[]){ strcpy(p,val);

strcpy(n,na);

}

SNode \*SNode::getNext(){ return next;

}

void SNode::setNext(SNode \*n){ next = n;

}

class LinkedList //LinkedList CLASS

{

private:

SNode \*start;

public:

LinkedList();

int Insert (char data[],char n[]); int Count ();

int Delete (char data[]); int Search (char data[]); void printList();

void concatenate(LinkedList l);

};

LinkedList::LinkedList(){ //DEFAULT CONSTRUCTOR start = NULL;

}

int LinkedList::Insert (char data[],char n[]){

SNode \*pNew, \*pTemp, \*pPre; int ch;

char key[10];

if(Search(data))

{

cout<<"\nItem already present !!!"; return 0;

}

cout<<"\n1. Insert President at start :\n2. Insert member after a specific member :\n3.

Insert Secretary at end:\n"; cout<<"\nEnter your choice:"; cin>>ch;

pNew = new SNode(); //Allocate memory

pNew->setData(data,n); //Store data

if(start == NULL) //List is empty now

{

start = pNew; //Attach it as a start list

}

else

{

switch(ch)

{

case 1:

pNew->setNext(start); //Attach new node to first node start=pNew; //Make start point to first node

break;

case 2: cout<<"Enter the item after which u have to insert:\n"; cin>>key;

//find the correct position pTemp = start; pPre = NULL;

while (pTemp != NULL && strcmp(pTemp->getPrn(),key)!=0)

{

pPre = pTemp;

pTemp= pTemp->getNext();

}//Insert after that if(pTemp!=NULL)

{

pNew->setNext(pTemp->getNext());

pTemp->setNext(pNew);

}

else

{

cout<<"\n Error!! element not found";

}

break;

case 3: pTemp = start;

while (pTemp->getNext() != NULL) pTemp= pTemp->getNext();

pTemp->setNext(pNew); break;

}

}

return 0;

}

int LinkedList::Count(){

SNode \*pTemp = start; if(pTemp==NULL)

{

return 0;

}

else {

int cnt=0;

while(pTemp!=NULL)

{

cnt++;

pTemp = pTemp->getNext();

}

return cnt;

}

}

int LinkedList::Delete (char key[]){ SNode \*pTemp, \*pPre;

//Find the item along with its predecessor pTemp = start; pPre = NULL;

if(pTemp->getNext()==NULL && strcmp(pTemp->getPrn(),key)==0) //only one node in list

{

delete(pTemp); start=NULL; return 1;

}

else

{

while (pTemp != NULL)

{

if(strcmp(pTemp->getPrn(),key)==0) break;

pPre = pTemp;

pTemp= pTemp->getNext();

}

if(pTemp!=NULL) //Item found

{

if(pPre == NULL) //Item found in first node start = start->getNext();

else

pPre->setNext(pTemp->getNext()); delete pTemp;

return 1;

}

}

return 0;

}

int LinkedList::Search (char key[]){ SNode \*pTemp = start;

while(pTemp!=NULL)

{

if (strcmp(pTemp->getPrn(),key)==0)

{

cout<<"\nItem Found : "; cout<<"\t"<<pTemp->getPrn()<<"\t"<<" "; cout<<"\t"<<pTemp->getName()<<"\t"<<" "; return 1;

}

pTemp = pTemp->getNext();

}

return 0;

}

void LinkedList::printList(){ SNode \*pTemp = start;

if(pTemp==NULL)

{

cout<<"\nThe list is empty. \n";

}

else

{

cout<<"\nThe list is : \n"; while(pTemp->getNext()!=NULL)

{

cout<<"|"<<pTemp->getPrn()<<"|"<<pTemp->getName()<<"->"; pTemp = pTemp->getNext();

}

cout<<"|"<<pTemp->getPrn()<<"|"<<pTemp->getName();

}

}

void LinkedList::concatenate(LinkedList l)

{ cout<<"\n in concatenate:"; SNode \*pTemp=start;

while(pTemp->getNext()!=NULL)

{

pTemp=pTemp->getNext();

}

pTemp->setNext(l.start);

}

int main()

{

LinkedList list,list2; char data[10],n1[10]; int n,n2,choice;

do

{

// system("cls");

cout<<"\nWelcome to Pinnacle Club: Where you may-"; cout<<"\n1. Add Student(1.President 2. Any member 3.Secretary"

<<"\n2. Delete memebrs"

<<"\n3. Count number of nodes in list1"

<<"\n4. Search"

<<"\n5. Display club members List-1"

<<"\n6. Read second list"

<<"\n7. Display club members List-2"

<<"\n8. Concatenate SE1 & SE2 list"

<<"\n9. Exit";

cout<<"\n\nEnter your choice : ";cin>>choice; switch(choice)

{

case 1:

//Insert

cout<<"\nInsert new data:"; cout<<"\nData Value : ";cin>>data;

cout<<"\nData Value name : ";cin>>n1;

list.Insert(data,n1);

break;

case 2://Delete

cout<<"\nWhich item to delete?...Enter item name : "; cin>>data;

int ret; ret=list.Delete(data); if(ret==1)

{

cout<<"\n Item deleted successfully!! remaining list is:";

list.printList();

}

else{

cout<<"\n Item not found in list.";

}

break;

case 3://count

cout<<"\nNo. of items in list are: "; cout<<list.Count();

break;

case 4://Search

cout<<"\nWhich item to Search?...\nEnter item name : "; cin>>data;

if(list.Search(data))

cout<<"\nItem Found";

else

cout<<"\nItem Not Found";

break;

case 5://display list 1

list.printList();

break;

case 6:

cout<<"\nInsert new data:"; cout<<"\nData Value : ";cin>>data;

cout<<"\nData Value name : ";cin>>n1;

list2.Insert(data,n1);

list2.printList(); break;

case 7: //display list-2 list2.printList(); break;

case 8: //concatenation

//list -> List 1

//list2 -> List 2 list.concatenate(list2);

cout<<"\n Concatenated list is:\n"; list.printList();

break;

case 9:

cout<<"\nThank you"; exit(0);

break; default:

cout<<"\nWrong Choice";

}

}while (choice != 9); return 0;

}

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

#include<iostream> using namespace std; struct node

{ int roll;

struct node \*next;

};

class info

{ node

\*head1=NULL,\*temp1=NULL,\*head2=NULL,\*temp2=NULL,\*head=NULL,\*temp=NULL,\*h1=NULL,

\*head3=NULL,\*temp3=NULL;

int c,i,f,j,k;

public:

node \*create(); void insert(); void allstud(); void vanila(); void butters(); void uice(); void nice(); void notice(); void ovanila(); void obutters(); void display();

void notboth();

} ;

node \*info::create()

{ node \*p=new(struct node); cout<<"enter student rollno"; cin>>c;

p->roll=c;

p->next=NULL; return p;

}

void info::insert()

{

node \*p=create();

if(head==NULL)

{ head=p;

}

else

{ temp=head;

while(temp->next!=NULL)

{ temp=temp->next; } temp->next=p;

}

}

void info::display()

{ temp=head;

while(temp->next!=NULL)

{ cout<<"\n"<<temp->roll; temp=temp->next;

} cout<<"\n"<<temp->roll;

}

void info::allstud()

{cout<<"enter no. of students"; cin>>k;

head=NULL; for(i=0;i<k;i++)

{ insert(); h1=head;

} display(); head=NULL;

}

void info::vanila()

{

cout<<"enter no. of students who like vanila"; cin>>k;

head=NULL; for(i=0;i<k;i++)

{ insert(); head1=head;

} display(); head=NULL;

}

void info::butters()

{

cout<<"enter no. of students who like butterscotch"; cin>>j;

for(i=0;i<j;i++)

{ insert(); head2=head;

} display(); head=NULL;

}

void info::uice()

{ cout<<"students who like vanila or butterscotch\n"; temp1=head1;

while(temp1!=NULL)

{

node \*p=new(struct node); p->roll=temp1->roll;

p->next=NULL; if(head3==NULL)

{ head3=p;

}

else

{ temp3=head3;

while(temp3->next!=NULL)

{ temp3=temp3->next; } temp3->next=p;

}

temp1=temp1->next;

}

temp2=head2; while(temp2!=NULL)

{ f=0;

temp1=head1; while(temp1!=NULL)

{

if(temp2->roll==temp1->roll)

{ f=1; }

temp1=temp1->next;

}

if(f==0)

{

node \*p=new(struct node); p->roll=temp2->roll;

p->next=NULL; if(head3==NULL)

{ head3=p;

}

else

{ temp3=head3; while(temp3->next!=NULL)

{ temp3=temp3->next; } temp3->next=p;

}

}

temp2=temp2->next;

}

temp3=head3; while(temp3->next!=NULL)

{ cout<<"\n"<<temp3->roll; temp3=temp3->next;

} cout<<"\n"<<temp3->roll;

}

void info::ovanila()

{

cout<<"\nstudents like only vanila \n";

temp1=head1; while(temp1!=NULL)

{ temp2=head2; f=0;

while(temp2!=NULL)

{ if(temp1->roll==temp2->roll)

{ f=1; }

temp2=temp2->next;

}

if(f==0)

{ cout<<"\n"<<temp1->roll; } temp1=temp1->next;

}

}

void info::obutters()

{

cout<<"\nstudents like only butterscotch\n"; temp2=head2;

while(temp2!=NULL)

{ temp1=head1; f=0;

while(temp1!=NULL)

{ if(temp2->roll==temp1->roll)

{ f=1; }

temp1=temp1->next;

}

if(f==0)

{ cout<<"\n"<<temp2->roll; } temp2=temp2->next;

}

}

void info::nice()

{

cout<<"\nstudents who like both vanila and butterscotch\n"; temp1=head1;

while(temp1!=NULL)

{ temp2=head2; while(temp2!=NULL)

{ if(temp1->roll==temp2->roll)

{ cout<<"\n"<<temp1->roll; } temp2=temp2->next;

}

temp1=temp1->next;

}

}

void info::notice()

{

cout<<"\nstudents who like neither vanila nor butterscotch\n"; temp=h1;

while(temp!=NULL)

{ temp3=head3; f=0;

while(temp3!=NULL)

{ if(temp->roll==temp3->roll)

{ f=1; }

temp3=temp3->next;

}

if(f==0)

{ cout<<"\n"<<temp->roll; } temp=temp->next;

}

}

void info::notboth()

{

//cout<<"\nstudents like only vanila \n";

temp1=head1; while(temp1!=NULL)

{ temp2=head2; f=0;

while(temp2!=NULL)

{ if(temp1->roll==temp2->roll)

{ f=1; }

temp2=temp2->next;

}

if(f==0)

{ cout<<"\n"<<temp1->roll; } temp1=temp1->next;

}

// cout<<"\nstudents like only butterscotch\n"; temp2=head2;

while(temp2!=NULL)

{ temp1=head1; f=0;

while(temp1!=NULL)

{ if(temp2->roll==temp1->roll)

{ f=1; }

temp1=temp1->next;

}

if(f==0)

{ cout<<"\n"<<temp2->roll; } temp2=temp2->next;

}

}

int main()

{ info s; int i;

char ch; do{

cout<<"\n choice the options";

cout<<"\n 1. To enter all students rollno ";

cout<<"\n 2. To enter the rollno of student who like vanila"; cout<<"\n 3. To enter the rollno of student who like butterscotch";

cout<<"\n 4. To display the rollno of student who like vanila or butterscotch"; cout<<"\n 5. To display the rollno of student who like only vanila";

cout<<"\n 6. To display the rollno of student who like only butterscotch";

cout<<"\n 7. To display the rollno of student who like both vanila and butterscotch "; cout<<"\n 8. To display the rollno of student who neither like vanila nor butterscotch"; cout<<"\n 9. To display the rollno of student who either like vanila or butterscotch but not

both";

cin>>i; switch(i)

{ case 1: s.allstud();

break;

case 2: s.vanila();

break; case 3: s.butters();

break; case 4: s.uice();

break; case 5: s.ovanila();

break;

case 6: s. obutters();

break; case 7: s.nice();

break; case 8: s.notice();

break;

case 9: s.notboth();

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;

}

NAME🡪YASH BHAVATHANKAR

ROLL-NO🡪205A027

DIVISION🡪SE-1

#include<iostream> #include<conio.h> #include<stdlib.h> #define size 5

using namespace std; class STACK\_CLASS

{

private:

struct stac{ int s[size]; int top;

}st; public:

STACK\_CLASS();

int stfull();

void push(int item); int stempty();

int pop();

void display();

}; STACK\_CLASS::STACK\_CLASS(){

st.top=-1;

for(int i=0;i<size;i++) st.s[i]=0;

}

int STACK\_CLASS::stfull(){ if(st.top>=size-1)

return 1; else

return 0;}

void STACK\_CLASS::push(int item){ st.top++;

st.s[st.top] =item;

}

int STACK\_CLASS::stempty(){ if(st.top ==-1)

return 1; else

return 0;

}

int STACK\_CLASS::pop(){ int item; item=st.s[st.top]; st.top--;

return(item);

}

void STACK\_CLASS::display(){ int i;

if(stempty())

cout<<"\n Stack Is Empty!"; else{

for(i=st.top;i>=0;i--) {cout<<"\n"<<st.s[i];}

};}

int main(){

int item,choice; char ans; STACK\_CLASS obj;

cout<<"\n\t\t Implementation Of Stack"; do

{

cout<<"\n Main Menu"; cout<<"\n1.Push\n2.Pop\n3.Display\n4.exit"; cout<<"\n Enter Your Choice: ";

cin>>choice; switch(choice)

{

case 1:

cout<<"\n Enter The item to be pushed"; cin>>item;

if(obj.stfull())

cout<<"\n Stack is Full"; else

obj.push(item); break;

case 2:if(obj.stempty())

cout<<"\n Empty stack/Underflow II";

else

{item=obj.pop();

cout<<"\n The popped element is "<<item;} break;

case 3:obj.display(); break;

case 4:exit(0);};

cout<<"\n Do You want To Continue?"; ans=getche();

}

while(ans =='Y' || ans =='y'); getch();

return 0;

}

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

#include<iostream> #include<math.h> #include<string.h> using namespace std;

class stack //creating linked list node

{

char p; // operator stack float r; // operand stack stack \*next;

public: stack()

{

r = 0;

next = NULL;

}

void setval(float x)

{

r = x;

}

float getval()

{

return r;

}

void setchar(char x)

{

p = x;

}

char getchar()

{

return p;

}

stack\* getnext()

{

return next;

}

void setnext(stack \*t)

{

next = t;

}

};

class intopos // operations

{

stack \*top; int j;

char \*post; char \*in; public:

intopos()

{

j = 0;

in = new char[20]; post = new char[20]; top = NULL;

}

stack\* gettop()

{

return top;

}

void settop(stack \*w)

{

top = w;

}

void push(char x); char pop();

void push1(float x); float pop1();

void op();

int icp(char x); int isp();

void disp(); void eva(); void eva1();

};

void intopos::push(char x)

{

stack \*nn;

nn = new stack(); nn->setchar(x); if(top == NULL)

{

settop(nn);

}

else

{

nn->setnext(gettop()); settop(nn);

}

}

char intopos::pop()

{

if(top == NULL)

{

cout<<"EMPTY";

return '\0';

}

else

{

stack \*t;

t = gettop(); settop(gettop()->getnext()); char z = t->getchar();

t->setnext(NULL); delete t;

return z;

}

}

int intopos::icp(char x)

{

if(x == '(')

return 4;

else if(x == '+' || x == '-') return 1;

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

else if(x == '^')

return 4;

else if(x == ')')

return -1;

else //if operand

return -2;

}

int intopos::isp()

{

if(gettop() != NULL)

{

char x = gettop()->getchar(); if(x == '(')

return 0;

else if(x == '+' || x == '-') return 1;

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

else if(x == '^')

return 3;

else //operand

return -3;

}

else if(gettop() == NULL)

return -4;

}

void intopos::op()

{

int a, b, l, v = 0; char x;

cout<<"\nLength of the expression : "; cin>>l;

cout<<"\nEnter the infix expression: "; while(v < l) //read infix expression

{

/\*cout<<"\nCharacter "<<v + 1<<" : ";\*/ cin>>x;

in[v] = x; v++;

}

in[v] = '\0';

v = 0;

while(v < l)

{

a = isp();

b = icp(in[v]); if(b == -2)

{

post[j] = in[v]; j++;

}

else if(b == 4 || b == 2 || b == 1)

{

if(b > a) // if greater, the push in stack

{

push(in[v]);

}

else

{

while(b <= a)

{

post[j] = pop(); j++;

a = isp(); // call isp()till icp <= isp

}

push(in[v]);

}

}

else if(b == -1)

{

while(gettop()->getchar() != '(')

{

post[j] = pop(); j++;

}

char c = pop();

}

v++; // read next char in infix expr.

}

while(gettop() != NULL)

{

char c = pop(); if(c != '(')

{

post[j] = c; j++;

}

}

}

void intopos::disp()

{

cout<<"\nInfix expression is: "<<in<<endl; cout<<"\nPostfix expression is: ";

for(int w = 0; w < j; w++)

{

cout<<post[w];

}

cout<<endl;

}

void intopos::push1(float x) // to push operands in stack

{

stack \*nn;

nn = new stack(); nn->setval(x); if(top == NULL)

{

settop(nn);

}

else

{

nn->setnext(gettop()); settop(nn);

}

}

float intopos::pop1() // to pop operands from stack

{

if(gettop() == NULL)

{

cout<<"EMPTY";

return 0;

}

else

{

stack \*t;

t = gettop(); settop(gettop()->getnext()); float z = t->getval();

t->setnext(NULL); delete t;

return z;

}

}

void intopos::eva()

{

float e, f, u; int w = 0, b; while(w < j)

{

b = icp(post[w]); if(b == -2)

{

cout<<"\nEnter a value of "<<post[w]<<" : "; cin>>e;

push1(e);

}

else if(b != -2)

{

if(b == 1)

{

if(post[w] == '+')

{

e = pop1();

f = pop1();

u = e+f; push1(u);

}

if(post[w] == '-')

{

e = pop1();

f = pop1();

u = (f-e);

push1(u);

}

}

else if(b == 2)

{

if(post[w] == '\*')

{

e = pop1();

f = pop1();

u = (e\*f);

push1(u);

}

if(post[w] == '/')

{

e = pop1();

f = pop1();

u = (f/e);

push1(u);

}

}

else if(b == 4)

{

e = pop1();

f = pop1();

u = pow(e,f);

push1(u);

}

} w++;

}

cout<<"\nThe value of the evaluated equation is "<<pop1();

}

void intopos::eva1() //for postfix evaluation

{

float e, f, u; char \*h;

int w = 0, b, l;

cout<<"\nEnter the length of the postfix expression : "; cin>>w;

h = new char[20];

cout<<"\nEnter the postfix expression: "; for(l = 0; l < w; l++)

{

cin>>h[l];

}

l = 0;

while(l < w)

{

b = icp(h[l]); if(b == -2)

{

cout<<"\nEnter a value of "<<h[l]<<" : "; cin>>e;

push1(e);

}

else if(b != -2)

{

if(b == 1)

{

if(h[l] == '+')

{

e = pop1();

f = pop1();

u = e+f; push1(u);

}

if(h[l] == '-')

{

e = pop1();

f = pop1();

u = (f-e);

push1(u);

}

}

else if(b == 2)

{

if(h[l] == '\*')

{

e = pop1();

f = pop1();

u = (e\*f);

push1(u);

}

if(h[l] == '/')

{

e = pop1();

f = pop1();

u = (f/e);

push1(u);

}

}

else if(b == 4)

{

e = pop1();

f = pop1();

u = pow(e,f);

push1(u);

}

} l++;

}

cout<<"\nThe value of the evaluated equation is "<<pop1();

}

int main()

{

int d; //choice

char y; // yes or no to continue do{

cout<<"\nSelect an option\n1: Infix to postfix conversion\t2: Postfix

evaluation\n";

cin>>d;

switch(d)

{

case 1:{

intopos i1;

i1.op(); // for conversion infix to postfix i1.disp();

cout<<"\nDo you want to evaluate the postfix expression(Press y) :

";

cin>>y;

if(y == 'y' || y == 'Y')

{

i1.eva(); // evaluate postfix

}

break;

}

case 2:{

intopos i2;

i2.eva1(); // evaluated input postfix expression break;

}

default:{

cout<<"\nThis option doesn't exist\n"; break;

}

}

cout<<"\nContinue(y/Y): "; cin>>y;

}while(y == 'Y' || y == 'y'); return 0;

}

NAME 🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

#include<iostream> using namespace std;

class job //class declaration

{

int num; //store serial no. of job

job \*next; //points toward next node of job public:

job()

{

next = NULL;

}

void setnum(int x)

{

num = x;

}

void setnext(job \*y)

{

next = y;

}

int getnum()

{

return num;

}

job\* getnext()

{

return next;

}

};

class order

{

job \*first,\*last,\*temp1; //first is front of queue and last is rear of queue, temp1 is taken for verification of data inside queue

public:

order()

{

first = NULL; last = NULL;

temp1=NULL;

}

void create(); void add(); void del(); void display();

};

void order::create() //It creates the initial queue

{

char ans; do

{

int n;

cout<<"\nEnter the job number: "; cin>>n;

job \*nn;

nn = new job(); nn->setnum(n); if(first == NULL)

{

first = nn; last = nn;

cout<<"\nfirst job inserted";

}

else

{

last->setnext(nn); last=last->getnext(); cout<<"\njob inserted";

}

cout<<"\nDo you want to add another job, press Y/y :"; cin>>ans;

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

}

void order::add() //Add job in queue from Rear

{

int n; job \*nn;

nn = new job();

cout<<"\nEnter the job number: "; cin>>n;

nn->setnum(n); if(first == NULL)

{

first = nn; last = nn;

}

else

{

last->setnext(nn); last=last->getnext();

}

cout<<"\nJob Added Successfully\n";

}

void order::del() //Remove job from queue from Front

{

if(first == NULL)

{

cout<<"\nJob Queue is Empty\n";

}

else

{

job \*temp;

temp = first->getnext(); delete(first);

first = temp;

}

cout<<"\nFirst Job Removed Successfully\n";

}

void order::display() //for verification(logically can't be performed) of data available inside queue

{

cout<<"\nJob Queue :\n";

temp1 = first;

cout<<temp1->getnum()<<" "; while(temp1->getnext()!= NULL)

{

temp1 = temp1->getnext(); cout<<temp1->getnum()<<" ";

}

cout<<endl;

}

int main() //main function

{

char ans; order l; do

{

int choice;

cout<<"\nChoice List: \n"; //choice list

cout<<"\n\t1.Create Job Queue\n"<<"\n\t2.Insert Job in Job Queue\n"<<"\n\t3.Remove First Job from Job Queue\n"<<"\n\t4.Display Job Queue\n";

cout<<"\nEnter Choice : "; cin>>choice;

cout<<endl;

switch(choice)

{

case 1:

l.create(); break;

case 2:

l.add(); break;

case 3:

l.del(); break;

case 4:

l.display(); break;

default:

cout<<"\n\tInvalid option Selected, Please try Again\n"; break;

}

cout<<"\nTo continue with this Queue operations, Press Y/y: "; cin>>ans;

cout<<endl;

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

return 0;

}

NAME🡪YASH BHAVATHANKAR

ROLL\_NO🡪205A027

DIVISION🡪SE-1

#include<iostream> #define SIZE 10

using namespace std;

class dqueue

{

int queue1[SIZE]; int front,rear;

public:

dqueue()

{

front=-1; rear=-1;

}

int deQueueFront(); int deQueueRear(); void enQueueRear(); void enQueueFront(); void display\_front();

void display\_rear();

};

void dqueue::enQueueRear()

{

int value;

if(rear==SIZE-1)

{

cout<<"\nQueue is full, Insertion is not possible!!! "; return;

}

else

{

if(front==-1)

{

//inserting first element front=0;

cout<<"\nEnter the value to be inserted:"; cin>>value;

rear=rear+1; queue1[rear] = value;

}

else

{

cout<<"\nEnter the value to be inserted:"; cin>>value;

rear=rear+1; queue1[rear] = value;

}

}

}

void dqueue::enQueueFront()

{

int value; if(front==0)

{

cout<<"\n Insertion is not possible, element exists at index 0!!!"; return;

}

else

{

cout<<"\nEnter the value to be inserted:"; cin>>value;

if(front==-1)//first element

{

front=rear=0;

}

else

{

front--;

}

queue1[front] = value;

}

}

int dqueue::deQueueRear()

{

int deleted\_element;

// deleted = queue1[rear]; if(front == -1)

{

cout<<"\nQueue is Empty!!! Deletion is not possible!!!"; return 0;

}

else if(front==rear) //only one element in Queue

{

front=rear=-1;

}

else if(rear==0)

{

deleted\_element = queue1[rear] ; rear=rear-1;

}

else

{

deleted\_element = queue1[rear]; rear=rear-1;

}

return deleted\_element;

}

int dqueue::deQueueFront()

{

int deleted\_element;

// deleted\_element = queue1[front]; if(front == -1)

{

cout<<"\nQueue is Empty!!! Deletion is not possible!!!"; return 0;

}

else if(front==rear)//only one element in Q

{

deleted\_element = queue1[front]; front=rear=-1;

}

else

{

deleted\_element = queue1[front]; front=front+1;

}

return deleted\_element ;

}

void dqueue::display\_front()

{

int i;

if(front == -1)

cout<<"\nQueue is Empty!!! Display is not possible!!!"; else

{

cout<<"\nThe Queue\_front element is:"; cout<<queue1[front]<<"\t";

}

}

void dqueue::display\_rear()

{

int i;

if(front == -1)

cout<<"\nQueue is Empty!!! Display is not possible!!!"; else{

cout<<"\nThe Queue\_Rear element is:"; cout<<queue1[rear]<<"\t";

}

}

int main()

{ dqueue DQ; char ch;

int choice1, value;

cout<<"\n\*\*\*\*\*\*\* Double Ended Queue operations\*\*\*\*\*\*\*\n"; do

{

cout<<"\n1.Insert at rear end \n"; cout<<"2.Delete from rear end \n"; cout<<"3.Delete from front end \n"; cout<<"4.Insert at front end \n"; cout<<"5.Display\_front \n"; cout<<"6.Display\_rear \n";

cout<<"\nEnter your choice : "; cin>>choice1;

switch(choice1)

{

case 1: DQ.enQueueRear(); break;

case 2: value = DQ.deQueueRear(); cout<<"\nThe value deleted is "<<value; break;

case 3: value=DQ.deQueueFront(); cout<<"\nThe value deleted is "<<value; break;

case 4: DQ.enQueueFront() ; break;

case 5: DQ.display\_front(); break;

case 6: DQ.display\_rear(); break;

default: cout<<"Wrong choice";

}

cout<<"\nDo you want to perform another operation (Y/y/n/N): "; cin>>ch;

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

return 0;

}

NAME🡪YASH BHAVATHANAKR

ROLL-NO🡪205A027

DIVISION🡪SE-1

#include<iostream> using namespace std; class queue

{

int front,rear,a; int arrq[20]; public:

queue()

{

a=0;

front=-1; rear=0;

}

void add();

void deliver\_order(); int qempty();

int qfull();

void display\_first(); void display\_last();

};

int queue :: qfull()

{

if(rear==a-2)

return 1;

else

return 0;

}

int queue :: qempty()

{

if(front==rear)

return 1;

else

return 0;

}

void queue :: add()

{

if(a==0)

{

cout<<"Enter maximum number orders to be placed in a day : "; cin>>a;

}

int order,ans,order\_no,count; count=1;

do

{

if(qfull()==1)

{

cout<<"Orders are full can't place your your order !!\n"; break;

}

else

{

cout<<"Pizza Types :\n1.A\n2.B\n3.C\n4.D\n"; cout<<"Enter your order for : ";

cin>>order;

cout<<"Your Order No. is "<<count<<"\n"; if(front==-1)

{

front=0; count=1; arrq[rear]=order;

}

else

{

arrq[rear]=order; rear=(rear+1)%a;

}

count++;

cout<<"Do you wish to add another order(1/0) ";

cin>>ans;

}

}

while(ans==1);

}

void queue :: deliver\_order()

{

if(qempty()==1)

cout<<"All orders are delivered !!\n";

else

{

int x=arrq[front];

cout<<"Order for "<<x<<" no. pizza delivered\n"; if(front==rear)

front=rear=-1;

else

front=(front+1)%a;

}

}

void queue :: display\_first()

{

if(qempty()==1)

cout<<"All orders are delivered !!\n";

else

{

cout<<"First Order is for "<<arrq[front]<<" no. pizza delivered\n";

}

}

void queue :: display\_last()

{

if(qempty()==1)

cout<<"All orders are delivered !!\n";

else

{

cout<<"Last Order is for "<<arrq[rear]<<" no. pizza\n";

}

}

int main()

{

queue q; int ch; char ans; do

{

cout<<"WELCOME TO PIZZA PARLOR \n1.Place Your Order.\n2.Display First Order\n3.Display Last Order.\n4.Deliver Order.\n";

cout<<"Enter your choice : "; cin>>ch;

switch(ch)

{

case 1 :q.add();

break;

case 2 :q.display\_first();

break;

case 3 :q.display\_last();

break;

case 4 :q.deliver\_order(); break;

default : cout<<"Invalid choice!!\n";

}

cout<<"Do you wish to continue..(y/n) : "; cin>>ans;

}

while(ans=='y'||ans=='Y'); return 0;

}