***NAME: NIKITA ARORA***

***ROLLNO: 2019329***

**C++ PRACTICAL**

**1. Write a program to compute the sum of the first n terms of the following series:**

**S = 1 - 1 / (2 ^ 2) + 1 / (3 ^ 3) - ... 1 / (n ^ n) where ^ is exponentiation.**

**The number of terms n is to be taken from user through command line. If command line argument is not found then prompt the user to enter the value of n.**

#include<iostream.h>

#include<math.h>

double series(int n)

{

int i;

double sum=0.0, ser;

for(i=1;i<=n;i++)

{

ser=(pow(-1,i+1)\*1/pow(i,i));

sum+=ser;

}

return sum;

}

int main(int argc,char \*argv[])

{

int n;

n=atoi(argv[1]);

cout<<"The no upto which we have to calculate series is "<<n<<"\n";

double res=series(n);

cout<<res;

return 0;

}

**2. Write a program to remove the duplicates from an array.**

#include<iostream.h>

void getdata(int a[],int num);

void duplicate(int a[],int &num);

int display(int a[],int num);

int main()

{

int num;

int a[10];

cout<<"Enter size of array ";

cin>>num;

getdata(a,num);

duplicate(a,num);

display(a,num);

return 0;

}

void getdata(int a[],int num)

{

int i;

cout<<"enter elements of array";

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

cin>>a[i];

}

void duplicate(int a[],int &num)

{

int i,j,k;

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

{

for(j=i+1;j<num;)

{

if(a[i]==a[j])

{

for(k=j;k<num;k++)

{

a[k]=a[k+1];

}

num--;

}

else

j++;

}

}

}

int display(int a[],int num)

{

cout<<"array after removing duplicates ";

int i;

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

cout<<a[i];

}

**3. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.**

#include<iostream.h>

#include<string.h>

int main(int argc,char \*argv[])

{

int count;

cout<<"argc is"<<argc;

for(int i=0;i<argc;i++)

cout<<argv[i]<<endl;

for(int i='A';i<='Z';i++)

{

count=0;

for(int j=1;j<argc;j++)

{

for(int k=0;k<strlen(argv[j]);k++)

{

if(argv[j][k]==i)

count++;

}

}

if(count!=0)

cout<<(char)i<<"appear"<<count<<"times"<<endl;

}

return 0;

}

**4. Write a menu driven program to perform following operations on strings (without using**

**inbuilt string functions):**

**a) Show address of each character in string**

**b) Concatenate two strings.**

**c) Compare two strings**

**d) Calculate length of the string (use pointers)**

**e) Convert all lowercase characters to uppercase**

**f) Reverse the string**

#include<iostream.h>

#include<stdio.h>

void length();

void reverse();

void compare();

void concatenate();

void lowerToUpper();

void address();

int main()

{

int x;

char ch='y';

do

{ cout<<"MENU:"<<endl;

cout<<"1.Length of string"<<endl;

cout<<"2.Reverse the string"<<endl;

cout<<"3.Compare two strings"<<endl;

cout<<"4.Concatenate two strings"<<endl;

cout<<"5.lowercase characters to uppercase"<<endl;

cout<<"6.address of each character of string"<<endl;

cout<<"Enter choice ";

cin>>x;

switch(x)

{

case 1: length();

case 2: reverse();

case 3: compare();

case 4: concatenate();

case 5: lowerToUpper();

case 6: address();

}

cout<<"Do you want to continue:";

cin>>ch;

}while(ch=='y');

}

void length()

{

char str[80];

cout<<"Enter string:"<<endl;

cin.getline(str,80);

int l;

for(l=0;str[l]!='\0';l++);

cout<<"Length of string is: "<<l<<endl;

return ;

}

void reverse()

{

char str[80];

cout<<"Enter string: ";

cin.getline(str,80);

int l;

for(l=0;str[l]!='\0';l++);

int temp;

for(int i=0,j=l-1;i<l/2;i++,j--)

{

temp=str[i];

str[i]=str[j];

str[j]=temp;

}

cout<<"Reverse string: "<<str<<endl;

return ;

}

void compare()

{

char str1[80],str2[80];

cout<<"Enter First String: "<<endl;

gets(str1);

cout<<"Enter Second String: "<<endl;

gets(str2);

int i;

for(i=0;str1[i]==str2[i] && str1[i]!='\0'&& str2[i]!='\0';i++);

{

if(str1[i]-str2[i]==0)

cout<<"Strings are equal "<<endl;

else

cout<<"Strings are not equal "<<endl;

}

return ;

}

void concatenate()

{

char str1[80],str2[80];

cout<<"Enter First string: ";

cin.getline(str1,80);

cout<<"Enter Second String: ";

cin.getline(str2,80);

int l=0;

for(l=0;str1[l]!='\0';l++);

for(int i=0;str2[i]!='\0';i++)

{

str1[l++]=str2[i];

}

str1[l]='\0';

cout<<"\nThe concatenate string is: "<<str1<<endl;

return ;

}

void lowerToUpper()

{

char str[80];

cout<<"Enter string: ";

cin.getline(str,80);

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

{

str[i]=(str[i]>='a' && str[i]<='z')?(str[i]-32):str[i];

}

cout<<"Uppercase string "<<str<<endl;

return ;

}

void address()

{

char str[80];

cout<<"Enter string: ";

cin.getline(str,80);

cout<<"the address of each char of string: " ;

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

{

cout<<static\_cast<void\*>(&str[i])<<endl;

}

}

**5. Write a program to merge two ordered arrays to get a single ordered array.**

#include<iostream.h>

void get\_data(int x[],int y);

void merge(int a[],int b[],int c[],int x,int y);

void display(int x[], int y);

int main()

{

int a[10],b[10],c[20],x,y;

cout<<"enter the size for array 'A' and'B'";

cin>>x>>y;

get\_data(a,x);

get\_data(b,y);

merge(a,b,c,x,y);

cout<<"the Array 'A': ";

display(a,x);

cout<<endl;

cout<<"the Array 'B': ";

display(b,y);

cout<<endl;

cout<<"the Array 'C': ";

display(c,x+y);

cout<<endl;

return (0);

}

void get\_data(int x[],int y)

{

cout<<"enter values";

for(int i=0;i<y;cin>>x[i++]);

}

void display(int x[], int y)

{

for(int i=0;i<y;i++)

cout<<x[i]<<" ";

}

void merge(int a[],int b[],int c[],int x,int y)

{

int k=0,i=0,j=0;

while(i<x&&j<y)

{

if(a[i]<b[j])

{

c[k]=a[i];

k++;

i++;

}

else

{

c[k]=b[j];

k++;

j++;

}

}

while(i<x)

{

c[k]=a[i];

k++;

i++;

}

while(j<y)

{

c[k]=b[j];

j++;

k++;

}

}

**6. Write a program to search a given element in a set of N numbers using Binary search  (i) with recursion (ii) without recursion.**

**1.With recursion**

#include<iostream.h>

void getdata(int[],int);

void binary(int[],int,int);

int main()

{

int a[30],n,x;

cout<<"enter size of array";

cin>>n;

getdata(a,n);

cout<<"enter the element to search";

cin>>x;

binary(a,n,x);

return 0;

}

void getdata(int a[],int n)

{

cout<<"enter array element";

int i;

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

cin>>a[i];

}

void binary(int a[],int n,int x)

{

int first,last,mid,flag=0;

first=0;

last=n-1;

while(first<=last)

{

mid=(first+last)/2;

if(x==a[mid])

{

flag=1;

break;

}

else

if(x>a[mid])

first=mid+1;

else

last=mid-1;

}

if(flag==1)

cout<<"No is found at"<<mid+1;

else

cout<<"Not found";

}

**BINARY SEARCH WITH RECURSION**

#include<iostream.h>

void getdata(int[],int);

int binary(int[],int,int,int);

int main()

{

int a[30];

int n,x;

cout<<"enter size of array ";

cin>>n;

getdata(a,n);

cout<<"enter the element to search ";

cin>>x;

int z=binary(a,0,n-1,x);

if(z!=-1)

cout<<"element found ";

else

cout<<"element not found ";

return 0;

}

void getdata(int a[],int n)

{

cout<<"enter array element ";

int i;

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

cin>>a[i];

}

int binary(int a[],int first,int last,int x)

{

int mid;

if(first<=last)

{

mid=(first+last)/2;

if(a[mid]==x)

return 1;

else if(a[mid]>x)

return binary(a,first,last-1,x);

else

return binary(a,first+1,last,x);

}

else

return -1;

}

**7. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.**

**1.With Recursion**

#include<iostream.h>

int gcd(int,int);

int main()

{

int a,b,x;

cout<<"Enter two numbers "<<"\n";

cin>>a>>b;

x=gcd(a,b);

cout<<"GCD of "<<a<<" "<<b<<"is"<<x;

return 0;

}

int gcd(int n,int m)

{

if(m==0)

return n;

return gcd(m,n%m);

}

**2.Without Recursion**

#include<iostream.h>

int gcd(int,int);

int main()

{

int a,b,x;

cout<<"Enter two numbers "<<"\n";

cin>>a>>b;

x=gcd(a,b);

cout<<"GCD of"<<a<<" "<<b<<"is"<<x;

}

int gcd(int n,int m)

{

int r;

r=n%m;

while(r!=0)

{

n=m;

m=r;

r=n%m;

}

return m;

}

**8. Create Matrix class. Write a menu-driven program to perform following Matrix**

**operations:**

**a) Sum**

**b) Product**

**c) Transpose**

#include<iostream.h>

class matrix

{

int a[10][10];

int row,col;

public:

matrix()

{

row=2;

col=2;

}

matrix(int r,int c)

{

row=r;

col=c;

}

void getinput();

matrix add (matrix o1);

matrix multiply(matrix o1);

matrix transpose();

void display();

};

void matrix::getinput()

{

cout<<"Enter the size of matrix";

cin>>row>>col;

cout<<"Enter the elements of matrix";

for(int i=0;i<row;i++)

{

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

{

cin>>a[i][j];

}

cout<<"\n";

}

}

matrix matrix::add(matrix o1)

{

matrix o3;

o3.row=row;

o3.col=col;

for(int i=0;i<row;i++)

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

o3.a[i][j]=a[i][j]+o1.a[i][j];

return o3;

}

matrix matrix::multiply(matrix o1)

{

matrix o3;

o3.row=row;

o3.col=o1.col;

for(int i=0;i<row;i++)

for(int j=0;j<o1.col;j++)

{

o3.a[i][j]=0;

for(int k=0;k<col;k++)

{

o3.a[i][j]=o3.a[i][j]+a[i][k]\*o1.a[k][j];

}

}

return o3;

}

matrix matrix::transpose()

{

matrix o3;

o3.row=row;

o3.col=col;

for(int i=0;i<row;i++)

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

{

o3.a[i][j]=a[j][i];

}

return o3;

}

void matrix::display()

{

for(int i=0;i<row;i++)

{

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

cout<<a[i][j]<<" ";

cout<<"\n";

}

}

int main()

{

int x;

char ch='y';

matrix o1,o2,o3;

o1.getinput();

o2.getinput();

do

{

cout<<"1.ADD"<<endl;

cout<<"2.MULTIPLY"<<endl;

cout<<"3.TRANSPOSE"<<endl;

cout<<"Enter choice"<<endl

cin>>x;

switch(x)

{

case 1: o3=o1.add(o2);

cout<<"Sum="<<"\n";

o3.display();

break;

case 2: o3=o1.multiply(o2);

cout<<"Multiply="<<"\n";

o3.display();

break;

case 3: o3=o1.transpose();

cout<<"Transpose="<<"\n";

o3.display();

break;

}

cout<<"Do you want to continue";

cin>>ch;

}while(ch=='y');

}

**9. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and  Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism.**

#include<iostream.h>

class person

{

private:

char name[20];

public:

void input()

{

cout<<"enter name ";

cin>>name;

}

virtual void display()

{

cout<<"name is "<<name<<"\n";

}

};

class student:public person

{

private:

char course[20];

float marks;

int year;

public:

void input()

{

person::input();

cout<<"enter course ";

cin>>course;

cout<<"enter marks ";

cin>>marks;

cout<<"enter year ";

cin>>year;

}

virtual void display()

{

person::display();

cout<<"course is "<<course<<"\n"<<"marks is "<<marks<<"\n "<<"year is "<<year<<"\n ";

}

};

class employee :public person

{

char department[20];

double salary;

public:

void input()

{

person::input();

cout<<"enter department ";

cin>>department;

cout<<"enter salary ";

cin>>salary;

}

virtual void display()

{

person::display();

cout<<"department is "<<department<<"\n"<<"salary is "<<salary<<"\n";

}

};

int main()

{

person o1;

student o2;

employee o3;

person\* p1;

o1.input();

p1=&o1;

p1->display();

o2.input();

p1=&o2;

p1->display();

o3.input();

p1=&o3;

p1->display();

}

**10. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.**

#include<iostream.h>

#include <math.h>

class triangle

{

int side1,side2,side3;

float area;

public:

triangle()

{

area=-1;

side1=side2=side3=-1;

}

void area1(int x)

{

side1=x;

area=(sqrt(3)/4)\*(side1\*side1);

}

void area1(int x,int y)

{

side1=x;

side2=y;

area=0.5\*(side1\*side2);

}

void area1(int x,int y,int z)

{

float s;

s=(x+y+z)/2.0;

side1=x;

side2=y;

side3=z;

area=sqrt(s\*(s-x)\*(s-y)\*(s-z));

}

void display()

{

if(side1!=-1)

cout<<side1;

if(side2!=-1)

cout<<side2;

if(side3!=-1)

cout<<side3;

cout<<"\n the area: "<<area<<endl;

}

triangle operator=(triangle t)

{

side1=t.side1;

side2=t.side2;

side3=t.side3;

area=t.area;

return \*this;

}

int operator==(triangle t)

{

if((side1==t.side1)&&(side2==t.side2)&&(side3==t.side3))

return 1;

else

return 0;

}

};

int main()

{

triangle t1;

triangle t2,t3,t4;

t2.area1(4);

t3.area1(2,4);

t4.area1(1,2,3);

t2.display();

t3.display();

t4.display();

t2=t3;

if(t2==t3)

cout<<"Triangles are same";

else

cout<<"Triangles are not same";

return 0;

}

**11. Write a program to read two numbers p and q. If q is 0 then throw an exception else  display the result of p/q.**

#include<iostream.h>

class divide

{

char a[30];

public:

divide(char\* x)

{

strcpy(a,x);

}

void display()

{

cout<<a;

}

};

int main()

{

int a,b,z;

cout<<"Enter the numbers: ";

cin>>a>>b;

try

{

if(b==0)

throw divide("Attempt to divide by zero");

z=a/b;

cout<<z;

}

catch(divide o1)

{

o1.display();

}

}

**12. Rewrite Matrix class of Q8 with exception handling. Exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by main() function.**

#include<iostream.h>

#include<iomanip.h>

class matrix

{

private:

int a[10][10];

int row,col;

public :

matrix(int r,int c)

{

row=r;

col=c;

}

matrix()

{

row=3;

col=3;

}

void getinput();

matrix operator +(matrix o1);

matrix operator -(matrix o1);

matrix operator \*(matrix o1);

matrix transpose();

void display();

};

void matrix::getinput()

{

cout<<"Enter the elements of the array : ";

for(int i=0;i<row;i++)

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

cin>>a[i][j];

}

matrix matrix::operator +(matrix o1)

{

if (row!=o1.row||col!=o1.col)

{

throw"Incompatible matrices ";

}

matrix o3(row,col);

for(int i=0;i<row;i++)

{

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

o3.a[i][j]=a[i][j]+o1.a[i][j];

}

return o3;

}

matrix matrix::operator -(matrix o1)

{

if (row!=o1.row||col!=o1.col)

throw"Incompatible matrices ";

matrix o3(row,col);

for(int i=0;i<row;i++)

{

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

o3.a[i][j]=a[i][j]-o1.a[i][j];

}

return o3;

}

matrix matrix::operator \*(matrix o1)

{

if(col!=o1.row)

throw"Incompatible matrices ";

matrix o3(row,o1.col);

for(int i=0;i<row;i++)

for (int j=0;j<o1.col;j++)

{

o3.a[i][j]=0;

for(int k=0;k<col;k++)

{

o3.a[i][j]=o3.a[i][j]+a[i][k]\*o1.a[k][j];

}

}

return o3;

}

matrix matrix::transpose()

{

matrix o3(row,col);

for(int i=0;i<row;i++)

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

o3.a[i][j]=a[j][i];

return o3;

}

void matrix::display()

{

cout<<"Matrix :"<<"\n";

for(int i=0;i<row;i++)

{

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

cout<<setw(5)<<a[i][j];

cout<<"\n";

}

}

int main()

{

int x;

matrix o1(2,2),o2( 2,3);

matrix o3(2,2);

o1.getinput();

o2.getinput();

char ch='y';

while(ch=='y')

{

cout<<"\n"<<"1.Addition "<<endl;

cout<<"2.Subtraction "<<endl;

cout<<"3.Multiply "<<endl;

cout<<"4.Transpose "<<endl;

cout<<"Enter the choice : "<<endl;

cin>>x;

switch(x)

{

case 1: try

{

o3=o1+o2;

cout<<"Add";

o3.display();

cout<<endl;

}

catch(char\*p)

{

cout<<p;

}

break;

case 2: try

{

o3=o1-o2;

cout<<"Subtract ";

o3.display();

cout<<endl;

}

catch(char\*p)

{

cout<<p;

}

break;

case 3: try

{

o3=o1\*o2;

cout<<"Multiply ";

o3.display();

cout<<endl;

}

catch(char\*p)

{

cout<<p;

}

break;

case 4: {

o3=o1.transpose();

o3=o2.transpose();

cout<<"Transpose ";

o3.display();

cout<<endl;

break;

}

cout<<" do you want to continue \n";

cin>>ch;

}

}

}

**13. Create a class Student containing fields for Roll No,Name, Class, Year and Total Marks.Write a program to store 5 objects of Student class in a file. Retrieve these records from file and display them.**

#include<iostream.h>

#include<fstream.h>

#include<iomanip.h>

class student

{

public:

int rollno;

char name[20];

int marks,year;

void input()

{

cout<<"Enter rollno,name,marks,year ";

cin>>rollno>>name>>marks>>year;

}

};

int main()

{

char ch;

student o1[5];

for(int i=0;i<5;i++)

o1[i].input();

ofstream f1;

f1.open("student.dat");

for(int i=0;i<5;i++)

{

f1<<o1[i].rollno<<setw(10)<<o1[i].name<<setw(10)<<o1[i].marks<<setw(10)<<o1[i].year<<endl;

}

f1.close();

ifstream f2;

f2.open("student.dat");

if(!f2)

{

cout<<"unable to open the file";

exit(100);

}

while(f2.get(ch))

{

cout<<ch;

}

f2.close();

}

**14. Copy the contents of one text file to another file, after removing all whitespaces.**

#include<iostream.h>

#include<fstream.h>

int main()

{

char ch;

ifstream f1;

ofstream f2;

f1.open("A1.dat");

if(!f1)

{

cout<<"unable to open file";

exit(100);

}

f2.open("A2.dat");

if(!f2)

{

cout<<"unable to open the file";

}

while(f1.get(ch))

if(ch!=' '&& ch!='\n'&& ch!='\t')

f2.put(ch);

f1.close();

f2.close();

}