PRACTICAL LIST QUESTIONS

**(PROGRAMMING FUNDAMENTALS USING C++)**

C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0285750.wmf

**SUBMITTED BY:** **NEHA KUMARI**

**SUBMITTED TO:** **MRS.SURUCHI CHAWLA**

**ROLL NO:** **2020331**

**DEPARTMENT: BSC COMPUTER SCIENCE**

**COURSE:** **C++**

**SEMERTER:** **1ST**

**DATE OF SUBMISSION:** **28th FEB 2021**

**PRACTICAL LIST QUESTIONS**

**(C++)**

**Q1.**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 prompt the user to enter the value of n. #include<iostream>

#include<cmath>

#include<stdlib.h>

using namespace std;

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

{

double s=0;

int n;

cout<<"argc"<<argc<<endl;

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

cout<<"argv="<<argv[i];

n=atoi(argv[1]);

int i=1;

while(i<=n)

s=s+pow(-1,(i+1))\*(1.0/(i\*i));

i++;

cout<<"sum of the series is "<<s<<endl;

return 0;

}

**OUTPUT:**

**enter the value of n12**

**the sum of series is0.819282**

**Q2.**Write a program to remove the duplicates from an array.

#include<iostream>

using namespace std;

int main()

{

int a[10],n,i,j,k;

cout<<"how many elements?";

cin>>n;

cout<<"enter element of array"<<endl;

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

{

cin>>a[i];

}

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

{

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

{

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

{

for(k=j;k<n-1;k++)

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

n--;

}

else

j++;

}

}

cout<<endl;

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

{

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

}

return 0;

}

**OUTPUT:**

**how many elements?7**

**enter element of array**

**1**

**2**

**2**

**4**

**5**

**7**

**8**

**124578**

**Q3.**Write a program that’s prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

#include <iostream>

#include <cstring>

using namespace std;

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

{

char s[20];

cout<<"argc= "<<argc<<endl;

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

cout<<"argv :"<<argv[i]<<endl;

strcpy(s,argv[1]);

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

{

int count=0;

for (int j=0;j<strlen(s);j++){

if (s[j]==i)

count++;

}

if (count!=0)

cout<<static\_cast<char>(i)<<" appears "<<count<<" times"<<endl;

}

return 0;

}

**OUTPUT:**

**argc= 2**

**argv :C:\Users\neha\OneDrive\Documents\C++\coolpanda\str\_ch\_count\bin\Debug\str\_ch\_count.exe**

**argv :neha**

**a appears 1 times**

**h appears 1 times**

**e appears 1 times**

**n appears 1 times**

**Q4.**Write a menu driven program to perform following operations on strings (without using inbuilt string function):

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

**b)** Concatenate two strings.

**c)** Compare two strings.

**d)** Calculate length of the strings (use pointer).

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

**f)** Reverse the string.

#include<iostream>

#include<cstring>

using namespace std;

int main()

{

int i,j,t;

int ch;

int count=0;

char a[20],b[20],c[40],d[20];

cout<<"enter first string :";

cin.getline(a,20);

cout<<"enter second string :";

cin.getline(b,20);

int l1=strlen(a);

int l2=strlen(b);

char ch1='y';

while(ch1=='y')

{

cout<<"enter choice :";

cin>>ch;

switch(ch)

{

case 1: for(i=0; i < l1; i++)

{

cout<<"address of elements "<< i <<"is :"<<&a+i<< endl;

}

break;

case 2: for(i=0,t=0; i < l1; i++,t++)

{

c[t]=a[i];

}

for(j=0,t=l1; j < l2; j++,t++)

{

c[t]=b[j];

}

cout<<"after concatenation :";

puts(c);

cout<< endl;

char c[40];

strcat(a,b);

cout<<"concatenated string :";

puts(a);

break;

case 3:if(strcmp(a,b)==0)

cout<<"strings are equal "<< endl;

else

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

break;

case 4: char\*p;

p=a;

while(p[i]!='\0')

{

i++;

}

cout<<"length of first string :"<< i<< endl;

char\*q;

i=0;

q=b;

while(\*(q++)!='\0'

{

i++;

}

cout<<"length of second string :"<< i<< endl;

break;

case 5:for(i=0; i < l1;i++)

{

if(islower(a[i]))

a[i]=toupper(a[i]);

}

for(i=0; i < l2;i++)

{

if(islower(b[i]))

b[i]=toupper(b[i]);

}

cout<<"converted string first :";

puts(a);

cout<<"converted string 2:";

puts(b);

cout<< endl;

break;

case 6: for(i=0; i < l1; i++)

{

if(isupper(a[i]))

a[i]=tolower(a[i]);

}

for(i=0; i < l2; i++)

{

if(isupper(b[i]))

b[i]=tolower(b[i]);

}

cout<<"converted string first :";

puts(a);

cout<<"converted string 2:";

puts(b);

cout<< endl;

break;

case 7: for(i=0,j=l2-1; i < l2/2; i++,j--)

{

ch=b[i];

b[i]=b[j];

b[j]=ch;

}

cout<<"Reversed string :";

puts(b);

break;

}

cout<<"do u want to continue";

cin>>ch1;

}

return 0;

}

**OUTPUT**

**enter first string :naina**

**enter second string :lakshita**

**enter choice :1**

**address of elements 0is :0x22fdf0**

**address of elements 1is :0x22fe04**

**address of elements 2is :0x22fe18**

**address of elements 3is :0x22fe2c**

**address of elements 4is :0x22fe40**

**do u want to continue y**

**enter choice :2**

**after concatenation :nainalakshita**

**concatenated string :nainalakshita**

**do u want to continue y**

**enter choice :3**

**strings are not equal**

**do u want to continue y**

**enter choice :4**

**length of first string :13**

**length of second string :8**

**do u want to continue y**

**enter choice :5**

**converted string first :NAINAlakshita**

**converted string 2:LAKSHITA**

**do u want to continue y**

**enter choice :6**

**converted string first :nainalakshita**

**converted string 2:lakshita**

**do u want to continue y**

**enter choice :7**

**Reversed string :atihskal**

**do u want to continue n**

**Q5.**Write a program to merge two ordered arrays to get a single ordered array.

#include<iostream>

using namespace std;

int main()

{

int a[5],b[5],c[10],i;

cout<<"enter element in 1st array";

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

{

cin>>a[i];

}

cout<<"enter element in 2nd array";

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

{

cin>>b[i];

}

cout<<"elements of array after merge";

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

{

c[i]=a[i];

c[i+5]=b[i];

}

for(i=0;i<10;i++)

{

cout<<c[i];

}

return 0;

}

**OUTPUT:**

**enter element in 1st array 1 2 3 4 5 6**

**enter element in 2nd array 7 8 9 10 11 12**

**elements of array after merge12345678910**

**Q6.** Write a program to search a given element in a set of N numbers using Binary search

i) with recursion ii) without recursion

**\*With recursion**

#include<iostream>

using namespace std;

void input(int x[],int n)

{

cout<<"enter the elements\n";

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

cin>>x[i];

}

void binsearch(int x[],int n, int x1,int low,int high)

{

int mid=0;

if(low>high)

{

cout<<"element not found";

return;

}

else

{

mid=(low+high)/2;

if(x[mid]==x1)

{

cout<<"element found at position"<<(mid +1);

return;

}

else if(x[mid]>x1)

binsearch(x,n,x1,low,mid-1);

else

binsearch(x,n,x1,mid+1,high);

}

}

int main()

{

int a[10];

int n;

cout<<"how many elements you want to enter\n";

cin>>n;

input(a,n);

char ch='y';

int x1;

while(ch=='y')

{

cout<<"enter the element you want to search";

cin>>x1;

binsearch(a,n,x1,0,n-1);

cout<<"do you want to continue";

cin>>ch;

}

return 0;

}

**OUTPUT**

**how many element you want to enter**

**6**

**enter the element**

**1**

**3**

**8**

**9**

**70**

**78**

**enter the element you want to search78**

**element found at position6do you want to continue**

**y**

**enter the element you want to search6**

**element is not founddo you want to continue**

**n**

**\*without recursion**

#include<iostream>

using namespace std;

int main()

{

int a[10];

int x,n;

cout<<"how many element you want to enter\n";

cin>>n;

cout<<"enter the element\n";

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

cin>>a[i];

char ch1='y';

while(ch1=='y')

{

cout<<"enter the element you want to search";

cin>>x;

int low=0;

int high=n-1;

int mid=0;

while(low<=high)

{

mid=(low+high)/2;

if(a[mid]==x)

{

cout<<"element found at position"<<(mid+1);

break;

}

else if(a[mid]>x)

high =mid-1;

else

low=mid +1;

}

if(low>high)

{

cout<<"element is not found";

}

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

cin>>ch1;

}

return 0;

}

**OUTPUT:**

**how many element you want to enter**

**6**

**enter the element**

**1**

**3**

**8**

**9**

**70**

**78**

**enter the element you want to search78**

**element found at position6do you want to continue**

**y**

**enter the element you want to search6**

**element is not found do you want to continue**

**n**

Q7.Write a program to calculate GCD of two numbers i) with recursion ii) without recursion

**i) With recursion:**

#include<iostream>

using namespace std;

int gcd(int n, int m)

{

int rem;

rem=n% m;

if(rem==0)

return m;

else

return gcd(m,rem);

}

int main()

{

int n,m;

cout<<"enter the two numbers";

cin>>n>>m;

int z=gcd(n,m);

cout<<"GCD of"<<n<<"and"<<m<<"is"<<z;

return 0;

}

**OUTPUT**

**enter the two numbers 5 25**

**GCD of5and25is5**

**ii) Without recursion:**

#include<iostream>

using namespace std;

int main()

{

int a,b,gcd;

cout<<"enter two numbers";

cin>>a>>b;

for(int i=1;i<=a&& i<=b;i++)

{

if((a%i==0)&&(b%i==0))

{

gcd=i;

}

}

cout<<"the gcd of"<<a<<"and"<<b<<"is"<<gcd;

return 0;

}

**OUTPUT:**

**enter two number 8 24**

**The gcd of 8 and 24 is 8**

**Q8.** Create Matrix class. Write a menu-driven program to perform following matrix operations:

a) Sum.

b) Product.

c) Transpose.

#include<iostream>

using namespace std;

class matrix1

{

int a[5][5];

int row,col;

public:

matrix1()

{

row=col=0;

}

matrix1(int i,int j)

{

row=i;

col=j;

}

void getinput(int,int);

void display();

matrix1 add(matrix1);

matrix1 multiply(matrix1);

matrix1 tranpose();

};

void matrix1::getinput(int i1, int j1)

{

int i,j;

row=i1;

col=j1;

cout<<"enter the elements matrix\n";

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

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

cin>>a[i][j];

}

matrix1 matrix1::add(matrix1 o2)

{

int i,j;

matrix1 o3;

if(row==o2.row && col==o2.col)

{

o3.row=row;

o3.col=col;

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

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

o3.a[i][j]=a[i][j]+o2.a[i][j];

}

else

cout<<"matrices order are not and cannot be added";

return o3;

}

void matrix1::display()

{

int i,j;

cout<<"matrix is \n";

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

{

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

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

cout<<endl;

}

}

matrix1 matrix1::multiply(matrix1 o2)

{

int i,j;

matrix1 o3;

if(col==o2.row)

{

o3.row=row;

o3.col=o2.col;

for(i=0;i<o3.row;i++)

{

for(j=0;j<o3.col;j++)

{

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

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

o3.a[i][i]=o3.a[i][j]+a[i][k]\*o2.a[k][j];

}

}

}

else

cout<<"matrices can not be multiplied\n ";

return o3;

}

matrix1 matrix1::tranpose()

{

int i,j;

matrix1 o3;

o3.row=col;

o3.col=row;

for(i=0;i<o3.row;i++)

for(j=0;j<o3.row;j++)

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

return o3;

}

int main()

{

matrix1 o1(2,3),o2(2,3),o3;

int row1,col1,row2,col2;

int ch;

char ch1='y';

while(ch1=='y')

{

cout<<"0.input matrices\n";

cout<<"1.addition\n";

cout<<"2 multiplication\n";

cout<<"3 tranpose\n";

cout<<"4 display\n";

cout<<"enter the choice\n";

cin>>ch;

switch(ch)

{

case 0: cout<<"enter the number of rows and cols of first matrix";

cin>>row1>>col1;

o1.getinput(row1,col1);

cout<<"enter the number of rows and cols of second matrix";

cin>>row2>>col2;

o2.getinput(row2,col2);

break;

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

break;

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

break;

case 3: o3=o1.tranpose();

break;

case 4: o1.display() ;

o2.display() ;

o3.display();

}

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

cin>>ch1;

}

return 0;

}

**OUTPUT**

**0.input matrices**

**1.addition**

**2 multiplication**

**3 transpose**

**4 display**

**enter the choice**

**0**

**enter the number of rows and cols of first matrix 1 3**

**enter the elements matrix**

**1 2 3**

**enter the number of rows and cols of second matrix 1 3**

**enter the elements matrix**

**4 5 6**

**do you want to continue**

**y**

**0.input matrices**

**1.addition**

**2 multiplication**

**3 tranpose**

**4 display**

**enter the choice**

**4**

**matrix is**

**1 2 3**

**matrix is**

**4 5 6**

**matrix is**

**do you want to continue**

**y**

**0.input matrices**

**1.addition**

**2 multiplication**

**3 transpose**

**4 display**

**enter the choice**

**1**

**do you want to continue**

**y**

**0.input matrices**

**1.addition**

**2 multiplication**

**3 transpose**

**4 display**

**enter the choice**

**4**

**matrix is**

**1 2 3**

**matrix is**

**4 5 6**

**matrix is**

**5 7 9**

**do you want to continue**

**y**

**0.input matrices**

**1.addition**

**2 multiplication**

**3 transpose**

**4 display**

**enter the choice**

**2**

**matrices can not be multiplied**

**do you want to continue**

**n**

**Q9.** 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>**

**#include<cstring>**

**using namespace std;**

**class person**

**{**

**public:**

**char name [20];**

**person()**

**{**

**strcpy(name," ");**

**}**

**void input()**

**{**

**cout<<"enter the name";**

**cin>>name;**

**}**

**virtual void display()**

**{**

**cout<<name;**

**}**

**};**

**class student: public person**

**{**

**public:**

**char course[20];**

**double marks;**

**int year;**

**student()**

**{**

**strcpy(course," ");**

**marks=year=0;**

**}**

**void input()**

**{**

**person::input();**

**cout<<"enter the marks";**

**cin>>marks;**

**cout<<"enter the year";**

**cin>>year;**

**}**

**virtual void diaplay()**

**{**

**person::display();**

**cout<<marks<<endl;**

**cout<<year<<endl;**

**}**

**};**

**class employee:public person**

**{**

**public:**

**char department[20];**

**double salary;**

**employee()**

**{**

**strcpy(department," ");**

**salary=0;**

**}**

**void input()**

**{**

**person::input();**

**cout<<"enter the department";**

**cin>>department;**

**cout<<"enter the salary";**

**cin>>salary;**

**}**

**virtual void display()**

**{**

**person::display();**

**cout<<department<<endl;**

**cout<<salary<<endl;**

**}**

**};**

**int main()**

**{**

**person \*bp;**

**person o1;**

**student o2;**

**employee o3;**

**bp=&o1;**

**o1.input();**

**bp->display();**

**bp=&o2;**

**o2.input();**

**bp->display();**

**bp=&o3;**

**o3.input();**

**bp->display();**

**}**

**OUTPUT:**

**enter the name neha**

**neha enter the name lakshita**

**enter the marks 89**

**enter the year358**

**lakshitaenter the name naina**

**enter the department computerscience**

**enter the salary 567.98**

**nainacomputerscience**

**567.98**

**Q10.**Create a class triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

**#include<iostream>**

**#include<math.h>**

**#include<cstring>**

**using namespace std;**

**class triangle**

**{**

**double area;**

**char type[20];**

**public:**

**triangle(char\*p)**

**{**

**area=0;**

**strcpy(type,p);**

**}**

**void computearea(int a,int b,int c);**

**void computearea(int a,int b);**

**void computearea(int a);**

**triangle operator=(triangle o1);**

**int operator==(triangle o2);**

**void display()**

**{**

**cout<<"Type of the triangle is"<<type<<endl;**

**cout<<"Area is"<<area<<"\n";**

**}**

**};**

**void triangle::computearea(int a, int b, int c)**

**{**

**double s;**

**s=(a+b+c)/2.0;**

**area=sqrt(s\*(s-a)\*(s-b)\*(s-c));**

**}**

**void triangle::computearea(int a, int b)**

**{**

**area=1.0/2\*a\*b;**

**}**

**void triangle::computearea(int a)**

**{**

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

**}**

**triangle triangle::operator=(triangle o1)**

**{**

**area=o1.area;**

**strcpy(type,o1.type);**

**return(\*this);**

**}**

**int triangle::operator==(triangle o2)**

**{**

**if(area==o2.area)**

**return 1;**

**else**

**return 0;**

**}**

**int main()**

**{**

**triangle o1("scalene"),o2("equilateral"),o3("isoscels");**

**o1.computearea(1,2,3);**

**o2.computearea(3);**

**o3.computearea(4,5);**

**o1.display();**

**o2.display();**

**o3.display();**

**cout<<"operator overloading"<<"\n";**

**o2=o3;**

**o2.display();**

**o3.display();**

**if(o2==o3)**

**cout<<"Area is equal";**

**else**

**cout<<"Area is not equal";**

**return 0;**

**}**

**OUTPUT:**

**Type of the triangle is scalene**

**Area is0**

**Type of the triangle is equilateral**

**Area is3.89711**

**Type of the triangle is isosceles**

**Area is10**

**Operator overloading**

**Type of the triangle is isosceles**

**Area is10**

**Type of the triangle is isosceles**

**Area is10**

**Area is equal**

**Q11.**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>

using namespace std;

int main()

{

int p,q;

double r;

char ch='y';

while(ch=='y')

{

cout<<"enter numerator";

cin>>p;

cout<<"enter denominator";

cin>>q;

try

{

if(q==0)

throw q;

else

r=p/q;

}

catch(int eq)

{

cout<<"exception :if denominator is zero divide not allowed";

}

cout<<r<<endl;

cout<<"do u want to continue"<<endl;

cin>>ch;

}

return 0;

}

**OUTPUT**

**enter numerator6**

**enter denominator0**

**exception :if denominator is zero divide not allowed1.97626e-322**

**do u want to continue**

**y**

**enter numerator6**

**enter denominator3**

**2**

**do u want to continue**

**n**

**Q12.** 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>

using namespace std;

class matrix

{

int a[5][5];

int row,col;

public:

matrix()

{

row=col=0;

}

matrix(int i, int j)

{

row=i;

col=j;

}

void getinput(int,int);

void display();

matrix add(matrix);

matrix multiply(matrix);

matrix transpose();

int getrow()

{

return row;

}

int getcol()

{

return col;

}

};

void matrix::getinput(int i1, int j1)

{

int i,j;

row=i1;

col=j1;

cout<<"enter the element matrix";

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

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

cin>>this->a[i][j];

}

matrix matrix::add(matrix o2)

{

int i,j;

matrix o3;

if(row==o2.row&& col==o2.col)

{

o3.row=row;

o3.col=col;

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

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

o3.a[i][j]=a[i][j]+o2.a[i][j];

}

else

{

throw "matrix order are not equal and cannot be added";

}

return o3;

}

void matrix::display()

{

int i,j;

cout<<"matrix is\n";

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

{

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

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

cout<<endl;

}

}

matrix matrix::multiply(matrix o21)

{

int i,j;

matrix o3;

if(col==o21.row)

{

o3.row=row;

o3.col=o21.col;

for(i=0;i<o3.row;i++)

{

for(j=0;j<o3.col;j++)

{

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

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

o3.a[i][i]=o3.a[i][j]+a[i][k]\*o21.a[k][j];

}

}

}

else

throw"matrix cannot be multiplied\n";

return o3;

}

matrix matrix::transpose()

{

int i,j;

matrix o3;

o3.row=col;

o3.col=row;

for(i=0;i<o3.row;i++)

for(j=0;j<o3.col;j++)

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

return o3;

}

int main()

{

matrix o1(2,3),o2(2,3),o3;

int row1,col1,row2,col2;

int ch;

char ch1='y';

while(ch1=='y')

{

cout<<"\n1 input\n";

cout<<"\n2 addition\n";

cout<<"\n3 multiplication\n";

cout<<"\n4 transpose\n";

cout<<"\n5 display\n";

cout<<"\n enter your choice\n";

cin>>ch;

switch(ch)

{

case 1:cout<<"enter the number of rows and cols of first matrix"; cin>>row1>>col1;

o1.getinput(row1,col1);

cout<<"enter the number of rows and cols of second matrix";

cin>>row2>>col2 ;

o2.getinput(row2,col2);

break;

case 2: try

{

o3=o1.add(o2);

}

catch(const char\*p)

{

cout<<"inside catch"<<p;

}

break;

case 3: try

{

o3=o1.multiply(o2);

}

catch(const char\*p)

{

cout<<"inside catch"<<p;

}

break;

case 4: o3=o1.transpose();

break;

case 5: o1.display();

o2.display();

o3.display();

break;

}

cout<<"do you to continue"<<endl;

cin>>ch1;

}

return 0;

}

**OUTPUT:**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**1**

**enter the number of rows and cols of first matrix 2 3**

**enter the element matrix 1 2 3 4 5 6**

**enter the number of rows and cols of second matrix 2 3**

**enter the element matrix 1 2 3 4 5 6**

**do you to continue**

**y**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**4**

**do you to continue**

**y**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**5**

**matrix is**

**1 2 3**

**4 5 6**

**matrix is**

**1 2 3**

**4 5 6**

**matrix is**

**1 4**

**2 5**

**3 6**

**do you to continue**

**y**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**3**

**inside catchmatrix cannot be multiplied**

**do you to continue**

**y**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**2**

**do you to continue**

**y**

**1 input**

**2 addition**

**3 multiplication**

**4 transpose**

**5 display**

**enter your choice**

**5**

**matrix is**

**1 2 3**

**4 5 6**

**matrix is**

**1 2 3**

**4 5 6**

**matrix is**

**2 4 6**

**8 10 12**

**do you to continue**

**n**

**Q13.** 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>

#include<fstream>

using namespace std;

class student

{

public:

char name[20];

int age;

double mark;

void getinput()

{

cout<<"enter the name";

cin>>name;

cout<<"enter the age";

cin>>age;

cout<<"enter the marks";

cin>>mark;

}

};

int main()

{

student o1,o2,o3,o4,o5;

o1.getinput();

o2.getinput();

o3.getinput();

o4.getinput();

o5.getinput();

ofstream f1;

f1.open("student.txt",ios::out);

if(!f1)

{

cout<<"unable to open file";

return 0;

}

f1<<o1.name;

f1.width(10);

f1<<o1.age;

f1.width(10);

f1<<o1.mark<<endl;

f1<<o2.name;

f1.width(10);

f1<<o2.age;

f1.width(10);

f1<<o2.mark<<endl;

f1<<o3.name;

f1.width(10);

f1<<o3.age;

f1.width(10);

f1<<o3.mark<<endl;

f1<<o4.name;

f1.width(10);

f1<<o4.age;

f1.width(10);

f1<<o4.mark<<endl;

f1<<o5.name;

f1.width(10);

f1<<o5.age;

f1.width(10);

f1<<o5.mark<<endl;

f1.close();

ifstream f2;

f2.open("student.txt",ios::in);

if(!f2)

{

cout<<"unable to open file";

return 0;

}

cout<<"file output"<<endl;

char ch;

while(f2.get(ch))

cout<<ch;

}

**OUTPUT:**

**enter the name neha**

**enter the age 18**

**enter the marks 449**

**enter the name lakshita**

**enter the age19**

**enter the marks678**

**enter the name sakshi**

**enter the age18**

**enter the marks 3456**

**enter the name bani**

**enter the age19**

**enter the marks5693**

**enter the name rainy**

**enter the age 18**

**enter the marks 6840**

**file output**

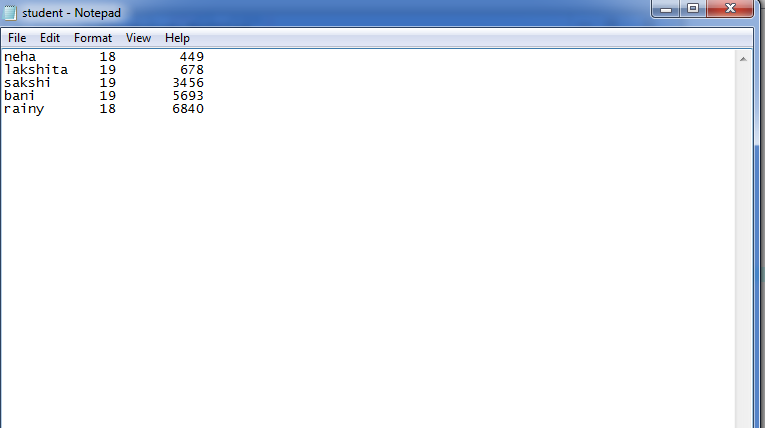
**neha 18 449**

**lakshita 19 678**

**sakshi 18 3456**

**bani 19 5693**

**rainy 18 6840**

**FILE:**

**Q14.**Copy the contents of one text file to another file, after removing all whitespaces.

#include<iostream>

#include<fstream>

using namespace std;

class student

{

public:

char name[20];

int age;

double mark;

void getinput()

{

cout<<"enter the name";

cin>>name;

cout<<"enter the age";

cin>>age;

cout<<"enter the marks";

cin>>mark;

}

};

int main()

{

student o1,o2,o3,o4,o5;

o1.getinput();

o2.getinput();

o3.getinput();

o4.getinput();

o5.getinput();

ofstream f1;

f1.open("student.txt",ios::out);

if(!f1)

{

cout<<"unable to open file";

return 0;

}

f1<<o1.name;

f1.width(10);

f1<<o1.age;

f1.width(10);

f1<<o1.mark<<endl;

f1<<o2.name;

f1.width(10);

f1<<o2.age;

f1.width(10);

f1<<o2.mark<<endl;

f1<<o3.name;

f1.width(10);

f1<<o3.age;

f1.width(10);

f1<<o3.mark<<endl;

f1<<o4.name;

f1.width(10);

f1<<o4.age;

f1.width(10);

f1<<o4.mark<<endl;

f1<<o5.name;

f1.width(10);

f1<<o5.age;

f1.width(10);

f1<<o5.mark<<endl;

f1.close();

ifstream f2;

f2.open("student.txt",ios::in);

if(!f2)

{

cout<<"unable to open file";

return 0;

}

cout<<"file output"<<endl;

char ch;

while(f2.get(ch))

cout<<ch;

cout<<"\\ copy one file to another file after removing whitespaces";

ifstream f3;

f3.open("student.txt",ios::in);

ofstream f4;

f4.open("student2.txt",ios::out);

while(f3.get(ch))

{

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

f4.put(ch);

}

f3.close();

f4.close();

return 0;

}

**OUTPUT:**

**enter the name neha**

**enter the age 18**

**enter the marks 578**

**enter the namelakshita**

**enter the age19**

**enter the marks578**

**enter the namerainy**

**enter the age18**

**enter the marks907**

**enter the namebani**

**enter the age19**

**enter the marks59789**

**enter the namesakshi**

**enter the age18**

**enter the marks857**

**file output**

**neha 18 578**

**lakshita 19 578**

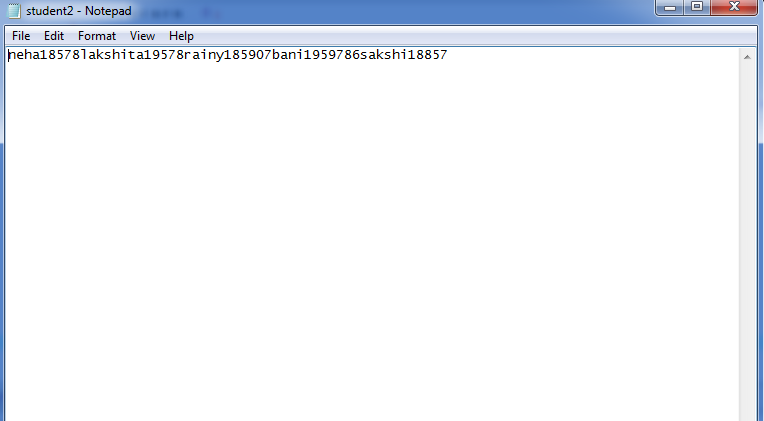
**rainy 18 907**

**bani 19 59789**

**sakshi 18 857**

**\ copy one file to another file after removing whitespaces**

**FILE:**

****