EXTRA QUESTIONS OF 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++**

**SEMESTER: 1ST**

**DATE OF SUBMISSION: 4th MARCH, 2021**

**EXTRA PROGRAMS OF FUNDEMENTAL PROGRAMMING OF C++**

**Q1. WAP to print a message on the monitor.**

#include<iostream>

using namespace std;

int main()

{

cout<<"Hello World!";

return 0;

}

**OUTPUT:**

**Hello World!**

**Q2.WAP to perform various arithmetic operations on two numbers..**

#include<iostream>

using namespace std;

int main()

{

int x,y,z;

int x1,y1,z1;

x=6;

y=8;

z=x+y;

x1=y/2;

y1=y-x;

z1=x\*y;

cout<<"The sum is" <<z<<"\n";

cout<<"The division is"<<x1<<"\n";

cout<<"The subtraction is"<<y1<<"\n";

cout<<"The product is"<<z1<<"\n";

return 0;

}

**OUTPUT:**

**The sum is14**

**The division is4**

**The subtraction is2**

**The product is48**

**Q3.**

#include <iostream>

using namespace std;

int main()

{

int a=1,b=9,c=3,d=20;

int x;

x=a\*b/c-d% 10;

cout<<"X="<<x<<endl;

return 0;

}

**OUTPUT:**  **X=3**

**Q4.WAP of Fibonacci series.**

#include<iostream>

using namespace std;

int main()

{

int a,b,n,i,sum;

a=0;

b=1;

cout<<"enter the number of terms to be printed in Fibonacci series";

cin>>n;

cout<<"Fibonacci series\n";

cout<<a<<","<<b<<",";

i=4;

while(i<=n)

{

sum=a+b;

cout<<sum<<",";

a=b;

b=sum;

i=i+1;

}

return 0;

}

**OUTPUT:**

**enter the number of terms to be printed in fibonacci series7**

**Fibonacci series**

**0,1,1,2,3,5,**

**Q5.WAP of to solve the following quadratic equation:**

**5x2+2x+1.**

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int a,b,c,x1,x2;

float discriminent,real,imaginary;

a=5;

b=2;

c=1;

discriminent=b\*b-4\*a\*c;

if(discriminent>0)

{

x1=(-b+sqrt(discriminent))/(2\*a);

x2=(-b+sqrt(discriminent))/(2\*a);

cout<<"x1="<<x1<<endl;

cout<<"x2="<<x2<<endl;

}

else if(discriminent==0)

{

cout<<"roots are real , same" <<endl;

x1=-b/(2\*a);

cout<<"x1=x2="<<x1<<endl;

}

else

{

real=-b/(2\*a);

imaginary=sqrt(-discriminent)/(2\*a);

cout<<"Roots are complex, different"<<endl;

cout<<"x1="<<real<<"+"<<imaginary<<"i" <<endl;

cout<<"x2="<<real<<"-"<<imaginary<<"i"<<endl;

}

return 0;

}

**OUTPUT:**

**Roots are complex, different**

**x1=0+0.4i**

**x2=0-0.4i**

**Q6.WAP to print odd number up to n terms.**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int i,n;

cout<<"Enter the value of n";

cin>>n;

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

{

cout<<"Odd numbers is"<<i<<endl;

i=i+1;

}

cout<<" The series of odd number up to n"<<n;

return 0;

}

**OUTPUT:**

**Enter the value of n7**

**Odd numbers is1**

**Odd numbers is3**

**Odd numbers is5**

**Odd numbers is7**

**The series of odd number up to n7**

**Q7.WAP to print factorial of n terms.**

#include<iostream>

using namespace std;

int main()

{

int p,n;

int fact=1;

cout<<"Enter the number whose factorial to be computed"<<endl;

cin>>n;

for(p=n;p>=1;p=p-1)

{

fact=fact\*p;

}

cout<<"The factorial of"<<n<<"is"<<fact;

return 0;

}

**OUTPUT:**

**Enter the number whose factorial to be computed**

**7**

**The factorial of7is5040**

**Q8.WAP to print the factorial of any number.**

#include<iostream>

using namespace std;

int main()

{

int n,x;

cout<<"Enter the number";

cin>>n;

x=1;

while(x<=n)

{

if(n% x==0)

{

cout<<"Factor is"<<x<<"\n";

}

x=x+1;

}

return 0;

}

**OUTPUT:**

**Enter the number8**

**Factor is1**

**Factor is2**

**Factor is4**

**Factor is8**

**Q9.WAP to print the sum of series:1/1+1/2+……..+1/n.**

#include<iostream>

using namespace std;

int main()

{

int n,pro=1;

double s=0;

cout<<"Enter the value of n";

cin>>n;

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

{

pro=pro\*i;

s=s+ (1.0/pro);

}

cout<<"Sum of series is"<<s;

return 0;

}

**OUTPUT:**

**Enter the value of n6**

**Sum of series is1.71806**

**Q10.WAP to the sum of odd number up to 100.**

#include<iostream>

using namespace std;

int main()

{

int i,n,sum;

n=100;

sum=0;

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

{

if(i% 2==0)

{

sum=sum+i;

}

}

cout<<"Sum of odd numbers from 1 to 100 is "<< sum;

return 0;

}

**OUTPUT:**

**Sum of odd numbers from 1 to 100 is 2550**

**Q11.WAP to print odd numbers up to 50**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int i,n;

cout<<"Enter the value of n";

cin>>n;

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

{

cout<<"Odd numbers is"<<i<<endl;

i=i+1;

}

cout<<" The series of odd number upto n"<<n;

return 0;

}

**OUTPUT:**

**Enter the value of n50**

**Odd numbers is1**

**Odd numbers is3**

**Odd numbers is5**

**Odd numbers is7**

**Odd numbers is9**

**Odd numbers is11**

**Odd numbers is13**

**Odd numbers is15**

**Odd numbers is17**

**Odd numbers is19**

**Odd numbers is21**

**Odd numbers is23**

**Odd numbers is25**

**Odd numbers is27**

**Odd numbers is29**

**Odd numbers is31**

**Odd numbers is33**

**Odd numbers is35**

**Odd numbers is37**

**Odd numbers is39**

**Odd numbers is41**

**Odd numbers is43**

**Odd numbers is45**

**Odd numbers is47**

**Odd numbers is49**

**The series of odd number up to n50**

**Q12.WAP to check whether the number is prime or not.**

#include<iostream>

using namespace std;

int main()

{

int n,i;

cout<<"Enter the number";

cin>>n;

for( i=2;i<n;i++)

{

if(n%i==0)

{

cout<<"Number is not prime";

break;

}

}

if(i==n)

cout<<"Number is prime";

return 0;

}

**OUTPUT:**

**Enter the number56**

**Number is not prime**

**Q13.WAP to print greatest common division of any number.**

#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 numbers28 36**

**The gcd of28and36is4**

**Q14.WAP to print Maximum number.**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int a,b,c;

cout<<"Enter the value of a, b and c\n";

cin>>a>>b>>c;

cout<<"Maximum number is";

if(a>b && a>c)

{

cout<<a;

}

else if(b>a && b>c)

{

cout<<b;

}

else

cout<<c;

return 0;

}

**OUTPUT:**

**Enter the value of a, b and c**

**5 7 8**

**Maximum number is8**

**Q15.WAP to print the sum of even numbers up to 100.**

#include<iostream>

using namespace std;

int main()

{

int n=100,i,Sum=0,x=2;

cout<<"Sum of 100 even no is";

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

{

Sum=Sum+x;

x=x+2;

}

cout<<Sum;

return 0;

}

**OUTPUT:**

**Sum of 100 even no is10100**

**Q16.** **WAP TO generate series1^2+2^2+\_\_\_n^2**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int n,sum=0,i;

cout<<"Enter the value of n";

cin>>n;

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

{

sum=sum+pow(i,i);

}

cout<<"Sum is"<<sum;

return 0;

}

**OUTPUT:**

**Enter the value of n5**

**Sum is3413**

**Q17.WAP to generate the series S=1!+2!+3!\_\_\_n!**

#include<iostream>

using namespace std;

int main()

{

int i,j,sum=0,n;

cout<<"Enter n";

cin>>n;

for(i=1;i<=n;i+=2)

{

int pro=1;

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

pro=pro\*j;

sum=sum+pro;

cout<<endl;

}

cout<<sum;

return 0;

}

**OUTPUT:**

**Enter n5**

**127**

**Q18.WAP to print the following pattern:**

#include<iostream>

using namespace std;

int main()

{

int i,j,row;

cout<<"Enter the number of rows";

cin>>row;

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

{

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

{

cout<<"1";

}

cout<<endl;

}

return 0;

}

**OUTPUT:**

**Enter the number of rows7**

**1**

**11**

**111**

**1111**

**11111**

**111111**

**1111111**

**Q19.WAP to print the following pattern:**

#include<iostream>

using namespace std;

int main()

{

int i,j,row;

cout<<"Enter the number of rows";

cin>>row;

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

{

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

{

cout<<"\*";

}

cout<<endl;

}

return 0;

}

**OUTPUT:**

**Enter the number of rows7**

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

**\*\*\*\*\*\***

**\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\***

**Q20.WAP to print the following pattern:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j,row;**

**cout<<"Enter the number of rows";**

**cin>>row;**

**for(i=1;i<=row;i++)**

**{**

**cout<<i<<endl;**

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

**{**

**cout<<j;**

**}**

**}**

**return 0;**

**}**

**OUTPUT:**

**Enter the number of rows6**

**1**

**12**

**123**

**1234**

**12345**

**123456**

**123456**

**Q21.WAP to print the following pattern:**

#include<iostream>

using namespace std;

int main()

{

int rows;

int n=10;

cout<<"Enter the number of rows";

cin>>rows;

for(int i=rows;i>=1;i--)

{

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

cout<<j;

for(int k=i-1;k>=1;k--)

cout<<k;

n++;

cout<<endl;

}

return 0;

}

**OUTPUT:**

**Enter the number of rows6**

**12345654321**

**123454321**

**1234321**

**12321**

**121**

**1**

**Q22.WAP to print the greatest element of the array:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,max,x;**

**max=-1;**

**cout<< "Enter number of term ";**

**cin>>n;**

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

**{**

**cout<<"Enter number";**

**cin>>x;**

**if(x>max)**

**max=x;**

**}**

**cout<< "Maximum is"<<max;**

**return 0;**

**}**

**OUTPUT:**

**Enter number of term 5**

**Enter number90**

**Enter number98**

**Enter number99**

**Enter number6780**

**Enter number0**

**Maximum is6780**

**Q23.WAP to print the following Diamond pattern:**

#include <iostream>

using namespace std;

int main()

{

int rows;

int n=10;

cout<<"No. of rows: ";

cin>>rows;

for(int i = 1; i<=rows; i++)

{

for(int p = 1; p<=n; p++)

cout<<" ";

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

cout<<j;

for(int k = i-1; k>=1; k--)

cout<<k;

n--;

cout<<endl;

}

n++;

for(int i = rows; i>=1; i--)

{

for(int p = 1; p<=n; p++)

cout<<" ";

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

cout<<j;

for(int k = i-1; k>=1; k--)

cout<<k;

n++;

cout<<endl;

}

return 0;

}

**OUTPUT:**

**No. of rows: 7**

**1**

**121**

**12321**

**1234321**

**123454321**

**12345654321**

**1234567654321**

**1234567654321**

**12345654321**

**123454321**

**1234321**

**12321**

**121**

**1**

**Q24.WAP to print the following pattern:**

#include <iostream>

using namespace std;

int main()

{

for(int i=1 ;i<=7; i++)

{

for(int j=1; j<=5; j++)

{

if(i==1 || i==7)

cout<<"=";

else

{

if(j==1 || j==5)

cout<<"\*";

else

cout<<" ";

}

}

cout<<endl;

}

return 0;

}

**OUTPUT:**

=====

\* \*

\* \*

\* \*

\* \*

\* \*

=====

**Q25.Array list series:**

#include <iostream>

using namespace std;

int main()

{

int list[10] = {0};

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

list[2\*i + 1] = i+2;

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

cout<<list[i]<<endl;

return 0;

}

**OUTPUT:**

**0**

**2**

**0**

**3**

**0**

**4**

**0**

**5**

**0**

**6**

**Q26.** **WAP to print max.no of an array.**

#include <iostream>

#include<iomanip>;

using namespace std;

int main() // Array is a homogeneous ordered collection of elements

{

int a[7];

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

for(int i=0; i<7; i++)

{

cin>>a[i];

}

int max=-1;

cout<<"Elements of an array are\n ";

for(int i=0; i<7; i++)

{

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

if(a[i]>max)

max=a[i];

}

cout<<"The max element of an array is "<<max;

return 0;

}

**OUTPUT:**

**Enter the elements of an array 4 5 2 8 2 5 9**

**Elements of an array are**

**4 5 2 8 2 5 9The max element of an array is 9**

**Q27.WAP to print the reverse the array.**

#include <iostream>

#include <iomanip>

using namespace std;

int main()

{

int a[10];

int n;

cout<<"How many elements u want to enter : ";

cin>>n;

cout<<"Enter the elements ";

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

cin>>a[i];

cout<<"reverse of an array is : "<<endl;

for(int j=n-1; j>=0 ; j--)

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

return 0;

}

**OUTPUT:**

**How many elements u want 2 enter : 4**

**Enter the elements 2 4 6 8**

**Reverse of an array is :**

**8 6 4 2**

**Q28.WAP to perform addition ,subtraction ,transpose of the matrix.**

#include<iostream>

using namespace std;

int main()

{

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

int row1,col1,row2,col2,row3,col3;

cout<<"enter the number of rows and columns of first matrix";

cin>>row1>>col1;

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

cin>>row2>>col2;

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

for(i=0;i<row1;i++)

for(j=0;j<col1;j++)

cin>>a[i][j];

cout<<"enter the element of second matrix\n";

for(i=0;i<row2;i++)

for(j=0;j<col2;j++)

cin>>b[i][j];

int ch;

char ch1='y';

while(ch1=='y')

{

cout<<"1 addition\n";

cout<<"2 transpose\n";

cout<<"3 subtraction\n";

cout<<"4 display\n";

cout<<"enter the choice\n";

cin>>ch;

switch(ch)

{

case 1:if(row1==row2&& col1==col2)

{

row3=row1;

col3=col1;

for(i=0;i<row1;i++)

for(j=0;j<col1;j++)

c[i][j]=a[i][j]+b[i][j];

}

else

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

break;

case 2: row1=col1;

col3=row1;

for(i=0;i<row3;i++)

for(j=0;j<col3;j++)

c[i][j]=a[j][i];

break;

case 3:if(row1==row2&&col1==col2)

{

row3=row1;

col3=col1;

for(i=0;i<row1;i++)

for(j=0;j<col1;j++)

c[i][j]=a[i][j]-b[i][j];

break;

}

case 4:cout<<"A matrix is\n";

for(i=0;i<row1;i++)

{

for(j=0;j<col1;j++)

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

cout<<endl;

}

cout<<"B matrix is\n";

for(i=0;i<row2;i++)

{

for(j=0;j<col2;j++)

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

cout<<endl;

}

cout<<" resulting matrix is\n";

for(i=0;i<row3;i++)

{

for(j=0;j<row3;j++)

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

cout<<endl;

}

}

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

cin>>ch1;

}

return 0;

}

**OUTPUT:**

**enter the number of rows and columns of first matrix2 2**

**enter the number of rows and columns of second matrix2 2**

**enter the elements of first matrix**

**1 2 3 4**

**enter the element of second matrix**

**5 6 7 8**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**1**

**do you want to continue**

**y**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**4**

**A matrix is**

**1 2**

**3 4**

**B matrix is**

**5 6**

**7 8**

**resulting matrix is**

**6 8**

**10 12**

**do you want to continue**

**y**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**2**

**do you want to continue**

**y**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**4**

**A matrix is**

**1 2**

**3 4**

**B matrix is**

**5 6**

**7 8**

**resulting matrix is**

**1 3**

**2 4**

**do you want to continue**

**y**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**3**

**do you want to continue**

**y**

**1 addition**

**2 transpose**

**3 subtraction**

**4 display**

**enter the choice**

**4**

**A matrix is**

**1 2**

**3 4**

**B matrix is**

**5 6**

**7 8**

**resulting matrix is**

**-4 -4**

**-4 -4**

**do you want to continue**

**n**

**Q29.WAP to print the min. no of an array .**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

const int size=10;

int a[size],n;

cout<<"enter how many elements of an array";

cin>>n;

cout<<"enter the element of an array";

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

{

cin>>a[i];

}

int min=99;

cout<<"element of an array are\n";

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

{

cout<<setw(4)<<a[i];

if(a[i]<min)

min=a[i];

}

cout<<"minimum is"<<min;

return 0;

}

**OUTPUT:**

**enter how many elements of an array5**

**enter the element of an array67**

**90**

**78**

**56**

**78**

**element of an array are**

**67 90 78 56 78minimum is56**

**Q30.WAP to print the sum of elements of an array.**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

const int size=10;

int a[size],n,sum=0;

cout<<"enter how many elements of an array";

cin>>n;

cout<<"enter the element of an array";

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

{

cin>>a[i];

sum=sum+a[i];

}

cout<<"the sum of array is"<<sum;

return 0;

}

**OUTPUT:**

**enter how many elements of an array4**

**enter the element of an array78**

**990**

**89**

**0**

**the sum of array is1157**

**Q31.WAP to search the position of an element in an array(linear search).**

#include<iostream>

using namespace std;

int main()

{

int a[10],n,x,i;

cout<<"how many element?";

cin>>n;

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

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

{

cin>>a[i];

}

cout<<"enter element to search";

cin>>x;

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

{

if(a[i]==x)

{

cout<<"element is found at position"<<i+1;

break;

}

}

if(i==n)

cout<<" element is not found";

return 0;

}

**OUTPUT:**

**how many element?4**

**enter element of array**

**7**

**8**

**0**

**9**

**enter element to search9**

**element is found at position4**

**Q32.WAP to print the odd number of elements of an array.**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

const int size=10;

int a[size],n,sum=0;

cout<<"enter how many elements of an array";

cin>>n;

cout<<"enter the element of an array";

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

{

cin>>a[i];

}

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

{

if(a[i]%2!=0)

cout<<"odd elements are"<<a[i];

cout<<endl;

}

return 0;

}

**OUTPUT:**

**enter how many elements of an array4**

**enter the element of an array7**

**3**

**6**

**1**

**odd elements are7**

**odd elements are3**

**odd elements are1**

**Q33.WAP to print even elements of an array.**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

const int size=10;

int a[size],n,sum=0;

cout<<"enter how many elements of an array";

cin>>n;

cout<<"enter the element of an array";

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

{

cin>>a[i];

}

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

{

if(a[i]%2==0)

cout<<"even elements are"<<a[i];

cout<<endl;

}

return 0;

}

**OUTPUT:**

**enter how many elements of an array4**

**enter the element of an array9**

**8**

**6**

**45**

**even elements are8**

**even elements are6**

**Q34.WAP to perform multiplication of matrix .**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

int A[4][4],B[4][4],C[4][4];

int ch1,row1,col1,row2,col2,row3,col3;

cout<<"enter the number of rows and columns of first matrix";

cin>>row1>>col1;

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

cin>>row2>>col2;

char ch='y';

while(ch=='y')

{

cout<<"1 input\n";

cout<<"2 multiplication\n";

cout<<"3 display\n";

cout<<"enter your choice\n";

cin>>ch1;

switch(ch1)

{

case 1:{

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

for(int i=0;i<row1;i++)

for(int j=0;j<col1;j++)

cin>>A[i][j];

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

for(int i=0;i<row2;i++)

for(int j=0;j<col2;j++)

cin>>B[i][j];

break;

}

case 2: if(col1!=row2)

{

cout<<"matrix cannot be multiplied";

}

else

{

row3=row1;

col3=col2;

for(int i=0;i<row1;i++)

{

for(int j=0;j<col2;j++)

{

C[i][j]=0;

for(int k=0;k<col1;k++)

C[i][j]=C[i][j]+A[i][k]+B[k][j];

}

}

}

break;

case 3: cout<<"first matrix\n";

for(int i=0;i<row1;i++)

{

for(int j=0;j<col1;j++)

cout<<setw(3)<<A[i][j];

cout<<endl;

}

cout<<"second matrix\n";

for(int i=0;i<row2;i++)

{

for(int j=0;j<col2;j++)

cout<<setw(3)<<B[i][j];

cout<<endl;

}

cout<<"product of matrices is as follows C=\n";

for(int i=0;i<row3;i++)

{

for(int j=0;j<col3;j++)

cout<<setw(3)<<C[i][j];

cout<<endl;

}

}

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

cin>>ch;

}

return 0;

}

**OUTPUT:**

**enter the number of rows and columns of first matrix2 3**

**enter the number of rows and columns of second matrix3 2**

**1 input**

**2 multiplication**

**3 display**

**enter your choice**

**1**

**enter the elements of first matrix**

**1 2 3 4 5 6**

**enter the elements of second matrix**

**1 2 3 4 5 6**

**do you want to continue**

**y**

**1 input**

**2 multiplication**

**3 display**

**enter your choice**

**2**

**first matrix**

**1 2 3**

**4 5 6**

**second matrix**

**1 2**

**3 4**

**5 6**

**product of matrices is as follows C=**

**15 18**

**24 27**

**do you want to continue**

**n**

**Q35.WAP to print the following pattern:**

#include <iostream>

using namespace std;

int main()

{

int rows;

char ch,ch1;

ch='E';

cout<<"Enter no. of rows : ";

cin>>rows;

for(int i=1; i<=rows; i++ )

{

ch1=ch;

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

{

cout<<ch1;

ch1++;

}

cout<<endl;

ch1++;

}

return 0;

}

**OUTPUT:**

**Enter no. of rows : 5**

**E**

**EF**

**EFG**

**EFGH**

**EFGHI**

**Q36.WAP <EOF> TO STOP AND TOTAL:**

#include <iostream>

using namespace std;

int main()

{

int x;

int sum=0;

cout<<"Enter ur nums: <EOF> to stop.\n";

while(cin>>x)

sum+=x;

cout<<"\nThe total is: "<<sum<<endl;

return 0;

}

**OUTPUT:**

**Enter ur nums: <EOF> to stop.**

**1 3 5 24 65 0 ^D**

**The total is: 98**

**Q37.** **WAP of SUM IN FUNCTION:**

#include <iostream>

using namespace std;

void sum(int a, int b)

{

cout<<(a+b);

}

int main()

{

sum(40,65);

return 0;

}

**OUTPUT:**

**105**

**Q38.** **MANIPULATORS IN OCTAL, HEXADECIMAL, DECIMAL:**

#include <iostream>

#include<iomanip>

using namespace std;

int main()

{

int x=56;

double y=2189.457892;

char ch='G';

cout<<"numbers without manipulators are\n";

cout<<x<<y<<ch;

cout<<"\nWith Manipulators";

cout<<setw(5)<<x<<endl;

cout<<setw(6)<<setfill('\*')<<x<<endl;

cout<<'$'<<fixed<<setw(10)<<y<<endl;

cout<<setfill(' ')<<setw(10)<<ch<<endl;

cout<<"Octal\n";

cout<<oct<<x;

cout<<"hexadecimal\n";

cout<<hex<<x;

cout<<"decimal\n";

cout<<dec<<x;

return 0;

}

**OUTPUT:**

**numbers without manipulators are**

**562189.46G**

**With Manipulators 56**

**\*\*56**

**$2189.457892**

**G**

**Octal**

**70hexadecimal**

**38decimal**

**56**

**Q39.** **WAP to find max no and also find how many times it occurs.**

#include<iostream>

using namespace std;

int main()

{

int i,n,max=0,count,x;

cout<<"enter no of terms";

cin>>n;

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

{

cout<<"enter no";

cin>>x;

if(x>max)

{

max=x;

count=1;

}

else

if(x==max)

count++;

cout<<endl;

}

cout<<"max no is"<<max;

cout<<"ocurs"<<count;

return 0;

}

**OUTPUT:**

**enter no of terms5**

**enter no1**

**enter no2**

**enter no5**

**enter no3**

**enter no2**

**max no is5ocurs1**

**Q40.CREATE a class fraction with constructor overloading.**

#include<iostream>

using namespace std;

class fraction

{

int n,d;

public:

fraction()

{

cout<<"parameterless constructor"<<endl;

n=0;

d=0;

}

fraction(int x, int y)

{

cout<<"parametric constructor"<<endl;

n=x;

d=y;

}

fraction(const fraction&o1)

{

cout<<"copy constructor o3"<<endl;

n=o1.n;

d=o1.d;

}

void getinput(int x, int y);

void display();

};

void fraction::getinput(int x ,int y)

{

n=x;

d=y;

}

void fraction::display()

{

cout<<n<<"/"<<d;

}

int main()

{

int x,y;

fraction o1(8,4),o2(50,100),o3(o2);

o1.display();

cout<<endl;

o2.display();

cout<<endl;

o3.display();

o1.getinput(78,90);

cout<<endl;

o1.display();

cout<<endl;

o3.display();

cout<<endl;

return 0;

}

**OUTPUT:**

**parametric constructor**

**parametric constructor**

**copy constructor o3**

**8/4**

**50/100**

**50/100**

**78/90**

**50/100**

**Q41.WRITE a function to swap the numbers.**

#include<iostream>

using namespace std;

void swap(int &a1,int &b1);

int main()

{

int x,y;

cout<<"enter the numbers";

cin>>x>>y;

swap(x,y);

cout<<"values after swap";

cout<<x<<y;

return 0;

}

void swap(int &a1,int &b1)

{

int temp;

temp=a1;

a1=b1;

b1=temp;

}

**OUTPUT:**

**enter the numbers5 6**

**values after swap6 5**

**Q42.WAP to print the address of string by using pointers.**

**#**include<iostream>

Using namespace std;

int main()

{

char \*p="Hello";

cout<<"string"<<p;

cout<<"string at based address"<<\*p<<endl;

cout<<"character next to h"<<(p+1)<<endl;

cout<<"H address"<<static\_cast<void\*>(p)<<endl;

cout<<"e address"<<static\_cast<void\*>(p+1)<<endl;

return 0;

}

**OUTPUT:**

**stringHellostring at based addressH**

**character next to hello**

**H address0x488000**

**e address0x488001**

**Q43.WAP to throw multiple errors.**

#include<iostream>

#include<cstdlib>

using namespace std;

int main()

{

double dividend,divisor;

cout<<"enter the dividend and divisor";

cin>>dividend>>divisor;

try

{

if(divisor==0.00)

throw divisor;

double quotient=dividend/divisor;

cout<<"Quotient is:"<<quotient <<endl;

}

catch(double &error)

{

cout<<"error 100:divisor 0\n";

}

catch(double &neg)

{

cout<<"error 101:negative divisor\n";

}

catch(...)

{

}

return 0;

}

**OUTPUT:**

**enter the dividend and divisor5**

**6**

**Quotient is:0.833333**

**Q44.WAP to sum the n numbers enter by the user and throw exception ofnegative number without effecting the sum.**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int num,sum=0;

cout<<" enter the number";

while( cin>>num)

try

{

if(num<0)

throw num;

sum+= num;

}

catch( int& negaerror)

{

cout<<"error 101: negative number/n";

}

cout<<" the sum is"<<sum;

return 0;

}

**OUTPUT:**

**enter the number 5**

**8**

**0**

**-9**

**error 101: negative number/n8**

**7**

**0**

**^D**

**the sum is28**

**Q45.WAP to check whether the number is a palindrome or noy.**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

int n,x,y,rev=0;

cout<<"enter the number";

cin>>n;

y=n;

while(n!=0)

{

x=n%10;

rev=(rev\*10)+x;

n=n/10;

}

cout<<"the reverse of the number is "<<rev<<endl;

if(y==rev)

cout<<"the number is palindrome";

else

cout<<"the number is not palindrome";

return 0;

}

**OUTPUT:**

**enter the number1221**

**the reverse of the number is 1221**

**the number is palindrome**

**Q46.WAP to print the following pattern:**

#include<iostream>

#include<iomanip>

using namespace std;

int main()

{

int a[6][6],i,j;

for(i=0;i<6;i++)

for(j=0;j<6;j++)

if(i+j<5)

a[i][j]=1;

else if(i+j>5)

a[i][j]=-1;

else

a[i][j]=0;

for(i=0;i<6;i++)

{

for(j=0;j<6;j++)

cout<< setw(4)<<a[i][j];

cout<<endl;

}

return 0;

}

**OUTPUT:**

**1 1 1 1 1 0**

**1 1 1 1 0 -1**

**1 1 1 0 -1 -1**

**1 1 0 -1 -1 -1**

**1 0 -1 -1 -1 -1**

**0 -1 -1 -1 -1 -1**

**Q47.WAP to print the following pattern:**

#include<iostream>

using namespace std;

int main()

{

int rows;

char ch,ch1;

ch='E';

cout<<"enter the number of rows";

cin>>rows;

for(int i=1;i<=rows;i++)

{

ch1=ch;

for(int p=1;p<=i;p++)

{

cout<<ch1;

ch1++;

}

cout<<endl;

ch--;

}

return 0 ;

}

**OUTPUT:**

**enter the number of rows5**

**E**

**DE**

**CDE**

**BCDE**

**ABCDE**

**Q48.WAP to extract a substring.**

#include<iostream>

#include<cstring>

using namespace std;

int main()

{

string str1="naina";

string str2;

cout<<"str1="<<str1<<endl;;

cout<<"str2="<<str2<<endl;;

str2=str1.substr();

cout<<"str2="<<str2<<endl;

str2=str1.substr(2,3);

cout<<"str2="<<str2<<endl;

return 0;

}

**OUTPUT:**

**str1=naina**

**str2=**

**str2=naina**

**str2=ina**

**Q49.WAP to swap the strings.**

#include<iostream>

#include<cstring>

using namespace std;

int main()

{

string str1="naina";

string str2="thakur";

cout<<"str1="<<str1<<endl;

cout<<"str2="<<str2<<endl;

swap(str1,str2);

cout<<"str1 after swap="<<str1<<endl;

cout<<"str2 after swap="<<str2<<endl;

return 0;

}

**OUTPUT:**

**str1=naina**

**str2=thakur**

**str1 after swap=thakur**

**str2 after swap=naina**

**Q50.CREATE a class fraction and show operator overloading.**

#include<iostream>

using namespace std;

class fraction

{

private:

int n,d;

public:

fraction()

{

n=d=0;

}

fraction(int i,int j)

{

n=i;

d=j;

}

fraction(const fraction &o1)

{

n=o1.n;

d=o1.d;

}

void store(int i, int j)

{

n=i;

d=j;

}

void print()const

{

cout<<n<<"/"<<d<<endl;

}

inline fraction operator+(fraction o1);

fraction operator++();

fraction operator++(int);

void operator+=(fraction);

fraction operator=(fraction o1)

{

n=o1.n;

d=o1.d;

return (\*this);

}

};

fraction fraction::operator+(fraction o1)

{

fraction temp;

temp.n=n\*o1.d+o1.n\*d;

temp.d=d\*o1.d;

return temp;

}

fraction fraction::operator++()

{

n+=d;

return(\*this);

}

fraction fraction::operator++(int)

{

fraction temp(n,d);

n+=d;

return temp;

}

void fraction::operator+=(fraction o1)

{

n=n\*o1.d+o1.n\*d;

d=d\*o1.d;

}

int main()

{

fraction o1,o2(4,5),o3(o2);

o1.print();

o2.print();

o3.print();

o1.store(5,6);

cout<<"preincrement\n";

o3=++o1;

o3.print();

o1.print();

cout<<"compound assignment\n";

o3+=o1;

o3.print();

o1.print();

cout<<"binary operator\n";

o3=o1+o2;

o1.print();

o2.print();

o3.print();

cout<<"post increment\n";

o3=o2++;

o3.print();

o2.print();

cout<<"assignment operator\n";

o3=o2=o1;

o3.print();

o2.print();

o1.print();

return 0;

}

**OUTPUT:**

**0/0**

**4/5**

**4/5**

**preincrement**

**11/6**

**11/6**

**compound assignment**

**132/36**

**11/6**

**binary operator**

**11/6**

**4/5**

**79/30**

**post increment**

**4/5**

**9/5**

**assignment operator**

**11/6**

**11/6**

**11/6**