Date: 6/01/2022

1. Write a program to find the Harmonic Mean.

CODE:

#include<stdio.h>

int main()

{

int n,i;

float sum=0,hm;

printf("\nEnter the size of the array:\n\n");

scanf("%d",&n);

float arr[n];

printf("\nEnter %d numbers to find their harmonic mean..\n\n",n);

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

scanf("%f",&arr[i]);

sum+=(1/arr[i]);

}

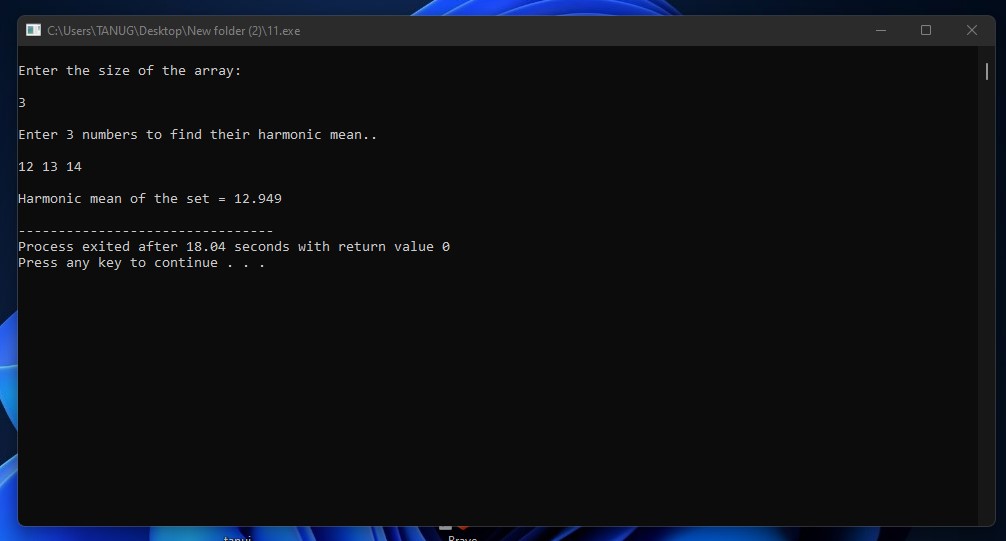
hm=n/sum;

printf("\nHarmonic mean of the set = %0.3f\n",hm);

return 0;

}

**OUTPUT:**

****

**2. WRITE A PROGRAM COUNT THE NUMBER OF DIGITS IN AN INTEGER.**

**CODE:**

#include<stdio.h>

int main()

{

long int num,count=0,rem;

printf("\nEnter an Integer value:\n\n");

scanf("%d",&num);

while(num!=0)

{

rem=num%10;

count++;

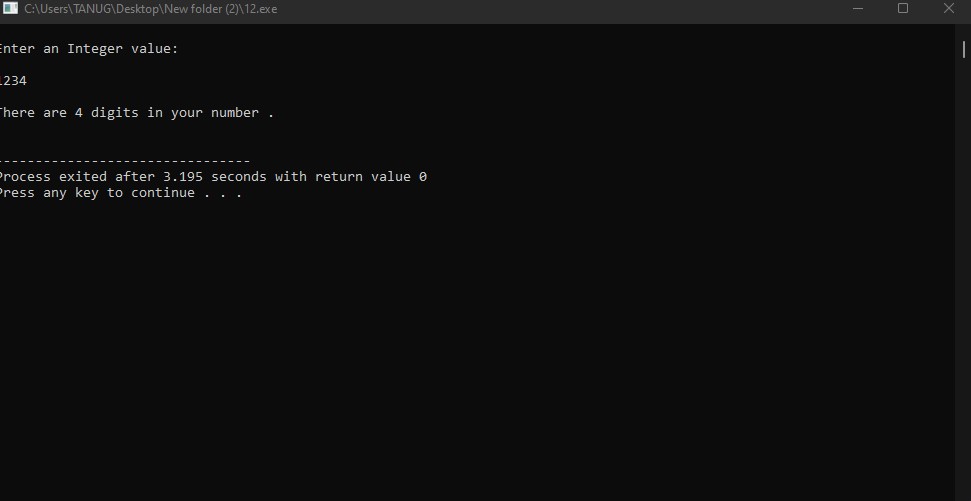
num=num/10;

}

printf("\nThere are %d digits in your number .\n\n",count);

return 0;

}

**OUTPUT:**

**3. Write a program menu based to find the LCM and GCD of a given two numbers.**

**CODE:**

#include <stdio.h>

int getlcm(int a, int b){

int m;

m = (a > b) ? a : b;

do{

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

return m;

m++;

}while(1);

}

int arraylcm(int arr[], int n){

int i,lcm = getlcm(arr[0],arr[1]);

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

lcm = getlcm(lcm, arr[i]);

}

return lcm;

}

int getgcd(int a, int b)

{

if (a == 0)

return b;

return getgcd(b % a, a);

}

int arraygcd(int arr[], int n){

int gcd = arr[0],i;

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

gcd = getgcd(arr[i], gcd);

return gcd;

}

int main() {

int n,i;

printf("\nEnter the size of the array:\n\n");

scanf("%d",&n);

int arr[n];

printf("\nEnter %d integers to find their LCM & GCD\n\n",n);

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

scanf("%d",&arr[i]);

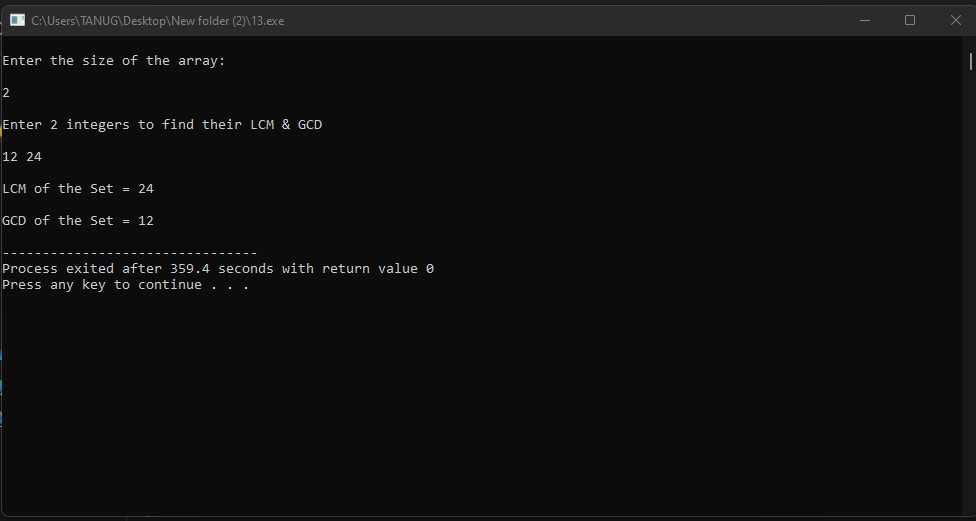
printf("\nLCM of the Set = %d\n",arraylcm(arr,n));

printf("\nGCD of the Set = %d\n",arraygcd(arr,n));

return 0;

}

**OUTPUT:**

****

**Date:13/01/2022**

**4. Write a program to enter the data for 10 emploeeys.**

**CODE:**

#include<stdio.h>

struct employee{

char empName[50];

int empID;

char empDes[50];

int empSalary;

};

int main()

{

int i;

struct employee data[10];

printf("\nEnter data for 10 employees of the organisation..\n\n ");

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

printf("\nSerial No - %d\n\n",i+1);

printf("\tEmployee's Name: ");

scanf("%s",&data[i].empName);

printf("\tEmployee's I'D: ");

scanf("%d",&data[i].empID);

printf("\tEmployee's Designation: ");

scanf("%s",&data[i].empDes);

printf("\tEmployee's Salary: ");

scanf("%d",&data[i].empSalary);

}

printf("\nDetail information for 10 Employee's:\n");

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

printf("\nEmployee I'd - %d :\n\n",data[i].empID);

printf("\tName: %s\n",data[i].empName);

printf("\tDesignation: %s\n",data[i].empDes);

printf("\tSalary: %d\n",data[i].empSalary);

}

}

**OUTPUT:**

5. Write a program to pass the entire structure to a function were to calculate the salary of an employee and display the Result.

**CODE:**

**#include<stdio.h>**

**#include<conio.h>**

**struct employee{**

**int ID;**

**char Name[50];**

**char Des[50];**

**int BasicSalary;**

**float HRA;**

**float DA;**

**float Salary;**

**};**

**int cal(struct employee bhoi);**

**int main()**

**{**

**int i;**

**struct employee data;**

**printf("\n...... Passing an entire Structure to a function .....\n\n");**

**printf("\nEnter the following to calculate the Basic Salary of an Employee...\n\n ");**

**printf("\tEmployee I'D: ");**

**scanf("%d",&data.ID);**

**printf("\tEmployee Name: ");**

**scanf("%s",&data.Name);**

**printf("\tEmployee Designation: ");**

**scanf("%s",&data.Des);**

**printf("\tEmployee Basic Salary: ");**

**scanf("%d",&data.BasicSalary);**

**printf("\tHRA in Percentage: ");**

**scanf("%f",&data.HRA);**

**printf("\tDA in Percentage: ");**

**scanf("%f",&data.DA);**

**data.Salary=cal(data);**

**getch();**

**}**

**int cal(struct employee bhoi)**

**{**

**printf("\nSo,Total Salary for the Employee...\n\n");**

**printf("\tEmployee I'd: %d\n\n",bhoi.ID);**

**printf("\tName: %s\n\n",bhoi.Name);**

**printf("\tDesignation: %s\n\n",bhoi.Des);**

**printf("\tBasic Salary: %d\n\n",bhoi.BasicSalary);**

**bhoi.HRA=(bhoi.BasicSalary/100)\*bhoi.HRA;**

**bhoi.DA=(bhoi.BasicSalary/100)\*bhoi.DA;**

**printf("\tHRA: %0.2f\n\n",bhoi.HRA);**

**printf("\tDA: %0.2f\n\n",bhoi.DA);**

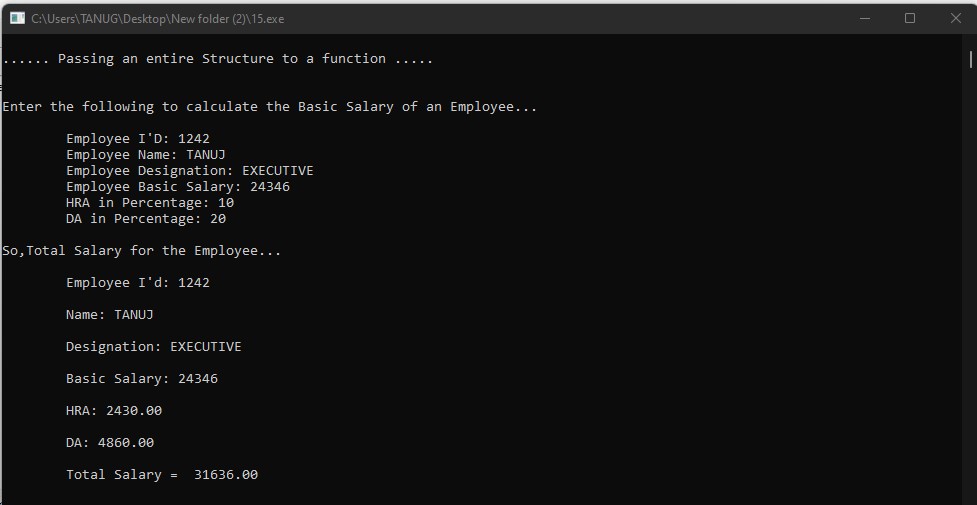
**bhoi.Salary=bhoi.BasicSalary+bhoi.HRA+bhoi.DA;**

**printf("\tTotal Salary = %0.2f\n\n",bhoi.Salary);**

**return bhoi.Salary;**

**}**

**OUTPUT:**

****

**6. Write a program to pass each and every member variable to a function where to calculate the “Salary” of an employee and display the result.**

**CODE:**

#include<stdio.h>

#include<conio.h>

struct employee{

int ID;

char Name[50];

char Des[50];

int BasicSalary;

float HRA;

float DA;

float Salary;

};

float cal(int ID, char Name[], char Des[], int BasicSalary, float HRA, float DA, float Salary);

int main()

{

int i;

struct employee data;

printf("\n...... Passing all the member variables of a Structure to a function .....\n\n");

printf("\nEnter the following to calculate the Basic Salary of an Employee...\n\n");

printf("\tEmployee I'D: ");

scanf("%d",&data.ID);

printf("\tEmployee Name: ");

scanf("%s",&data.Name);

printf("\tEmployee Designation: ");

scanf("%s",&data.Des);

printf("\tEmployee Basic Salary: ");

scanf("%d",&data.BasicSalary);

printf("\tEmployee HRA: ");

scanf("%f",&data.HRA);

printf("\tEmployee DA: ");

scanf("%f",&data.DA);

data.Salary=cal(data.ID, data.Name,data.Des,data.BasicSalary,data.HRA,data.DA,data.Salary);

getch();

}

float cal(int ID, char Name[50], char Des[50], int BasicSalary, float HRA, float DA, float Salary)

{

printf("\nSo,Total Salary for the Employee...\n\n");

printf("\tEmployee I'd: %d\n\n",ID);

printf("\tName: %s\n\n",Name);

printf("\tDesignation: %s\n\n",Des);

printf("\tBasic Salary: %d\n\n",BasicSalary);

HRA=(BasicSalary/100)\*HRA;

DA=(BasicSalary/100)\*DA;

printf("\tHRA: %0.2f\n\n",HRA);

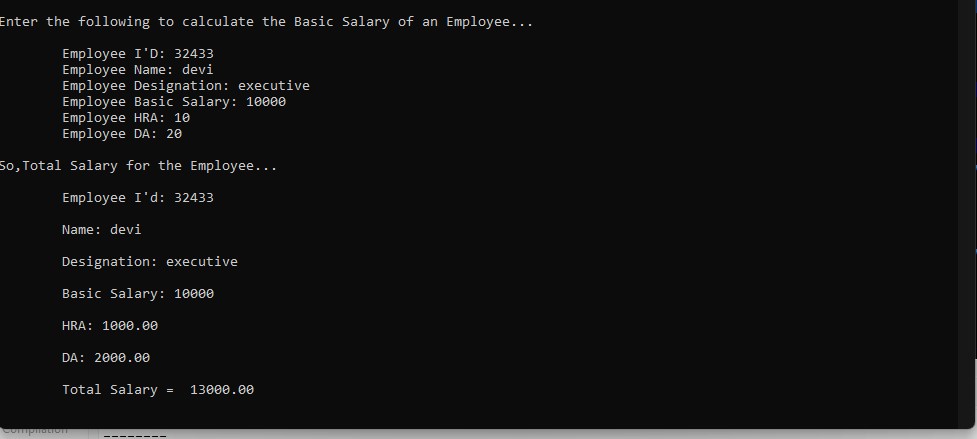
printf("\tDA: %0.2f\n\n",DA);

Salary=BasicSalary+HRA+DA;

printf("\tTotal Salary = %0.2f\n\n",Salary);

return Salary;

}

**OUTPUT: **

**7. Write a Program to convert binary number to octal and binary number to decimal.**

**CODE:**

#include<stdio.h>

#include<math.h>

int main()

{

long int bin,oct=0,dec=0,pro=0,d;

printf("\nEnter a binary number\n\n");

scanf("%ld",&bin);

while(bin>0)

{

dec+=(bin%10)\*round(pow(2,pro));

pro++;

bin/=10;

}

d=dec;

pro=1;

while(d>0)

{

oct+=(d%8)\*pro;

pro\*=10;

d/=8;

}

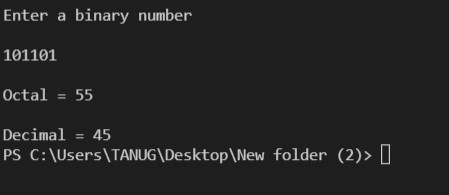
printf("\nOctal = %ld\n",oct);

printf("\nDecimal = %ld\n",dec);

return 0;

}

**OUTPUT:**



8. Write a program to find the maximum and minimum and how many times they occur in the array.

**CODE:**

#include<stdio.h>

#include<math.h>

int main()

{

int n,i,max,min,cmax=0,cmin=0;

printf("\nEnetr the size of the array:\n");

scanf("%d",&n);

int arr[n];

printf("\nEnter %d integers to find the Max and Min number\n",n);

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

scanf("%d",&arr[i]);

max=arr[0];

min=arr[0];

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

{

if(arr[i]>max)

max=arr[i];

else if(arr[i]<min)

min=arr[i];

}

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

{

if(arr[i]==max)

cmax++;

else if(arr[i]==min)

cmin++;

}

printf("\nMaximum = %d\n\nMinimum = %d\n",max,min);

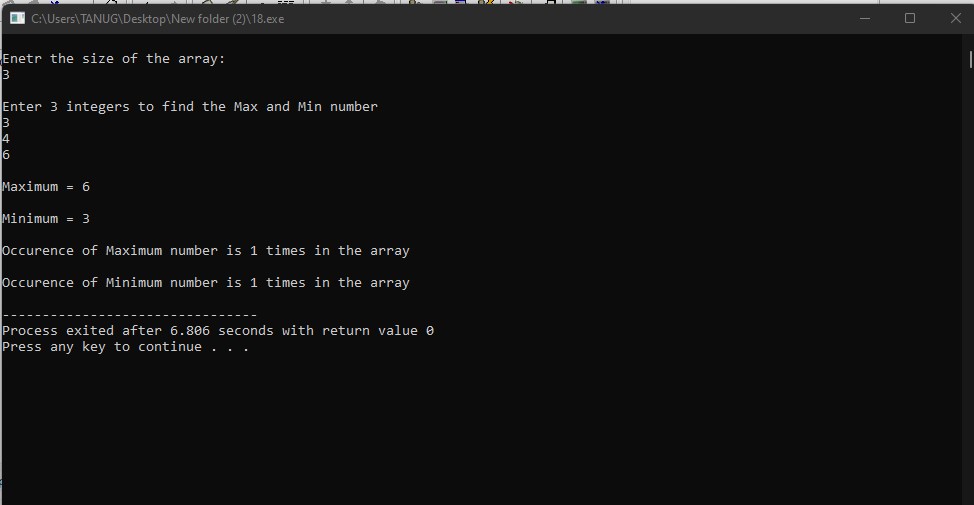
printf("\nOccurence of Maximum number is %d times in the array\n",cmax);

printf("\nOccurence of Minimum number is %d times in the array\n",cmin);

return 0;

}

OUTPUT:



9. Write a program to find the kth smallest element in an array of elements.

CODE:

#include<stdio.h>

#include<conio.h>

int main(){

int i,j,n,k,temp;

printf("\nEnter the size of the array:\n\n");

scanf("%d",&n);

int arr[n];

printf("\nEnter %d integer values:\n\n",n);

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

scanf("%d",&arr[i]);

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

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

if(arr[i]>arr[j]){

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

repeat:

printf("\nWhich position's smallest number you wanna see??\n\n ");

scanf("%d",&k);

if(k<1 || k>n){

printf("\nError !! Please try again - \n");

goto repeat;

}

else if(k==1)

printf("\nThe 1st smallest element = %d\n",arr[k-1]);

else if(k==2)

printf("\nThe 2nd smallest element = %d\n",arr[k-1]);

else if(k==3)

printf("\nThe 3rd smallest element = %d\n",arr[k-1]);

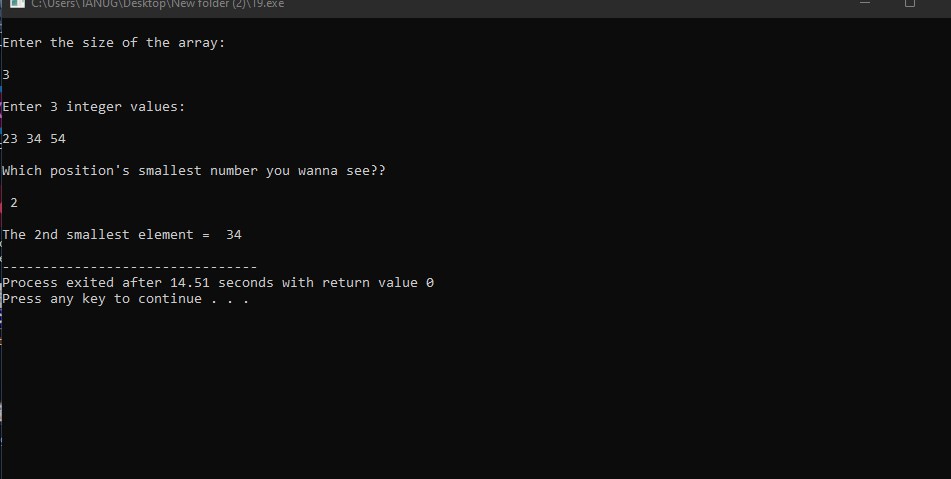
else

printf("\nThe %dth smallest element = %d\n",arr[k-1]);

return 0;

}

OUTPUT:



DATE:27/01/2022

10. Write a program to demonstrate use of scope resolution operator.

CODE:

#include <iostream>

using namespace std;

int b=67;

int main()

{

int b = 30;

{

cout << "Inside the inner block:\n";

cout << "b = " << b << endl;

cout << "::b = " << ::b << endl;

}

b = b+10;

cout << "Outside the block:\n";

cout << "b = b+10\n";

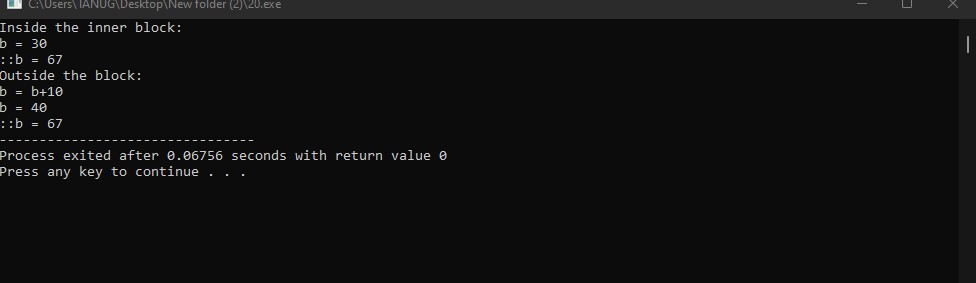
cout << "b = " << b << endl;

cout << "::b = " << ::b;

return 0;

}

OUTPUT:



11. Write a program to demonstrate use of reference variable.

CODE:

#include <iostream>

using namespace std;

int main()

{

int S;

int &L = S;

cout << "Enter the age of S: \n";

cin >> S;

cout << "Applying --int &L = Saloman-- \n";

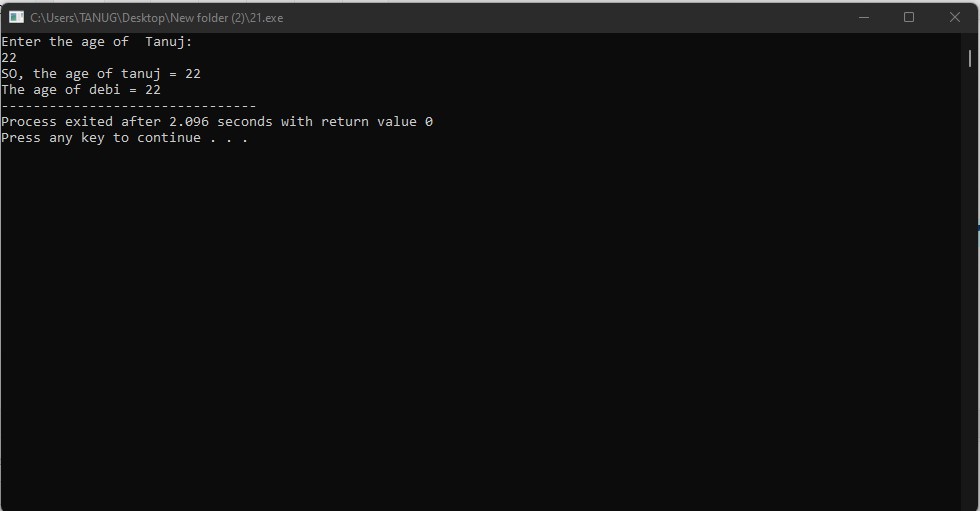
cout << "SO, the age of S = " << S << endl;

cout << "The age of L = " << L;

return 0;

}

OUTPUT:



12. Write a program to show the use of the pointers.

CODE:

#include <iostream>

using namespace std;

int main()

{

int a, \*b;

a = 50;

b = &a;

cout << "Integer a = " << a << endl;

cout << "\nApplying -- b = &a -- \n";

cout << "\n\*b = " << \*b << endl;

cout << "\nb = " << b << endl;

\*b = 100;

cout << "\nNow, Applying -- \*b = 100 -- \n";

cout << "\nInteger a = " << a << endl;

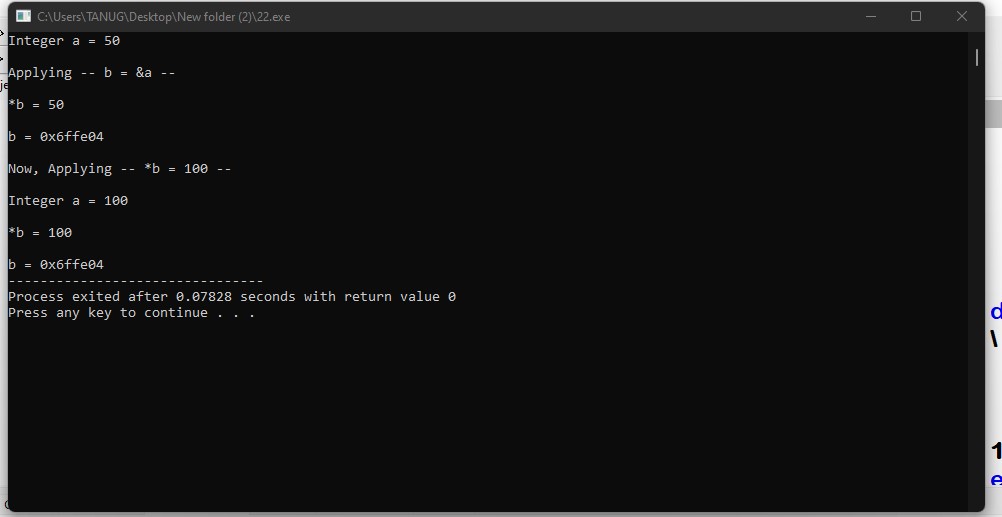
cout << "\n\*b = " << \*b<<endl;

cout << "\nb = " << b;

return 0;

}

OUTPUT:



13. Write a program to show the use of generic pointers

CODE:

#include <iostream>

using namespace std;

int main()

{

int x=8, \*y;

void \*z;

y=&x;

z=y;

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

cout << "&x = "<<&x<<endl;

cout << "\nApplying ( y=&x and z=y )...\n ";

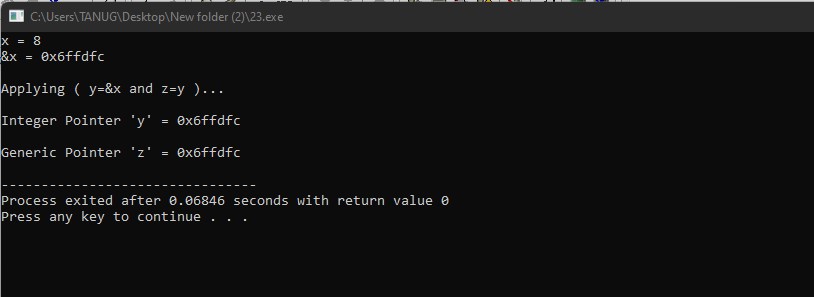
cout << "\nInteger Pointer 'y' = " << y << endl;

cout << "\nGeneric Pointer 'z' = " << z << endl;

return 0;

}

OUTPUT



14. . Write a program to show the use of enum datatype

CODE:

#include <iostream>

using namespace std;

int main()

{

enum color{violet,indigo,blue,green,yellow,orange,red};

int num;

Repeat:

cout<<"enum color{violet,indigo,blue,green,yellow,orange,red}"<<endl;

cout << "Enter a number between 1 to 7: "<<endl;

cin >> num;

switch(num-1)

{

case violet:

cout << "You got Violet";

break;

case indigo:

cout << "You got Indigo";

break;

case blue:

cout << "You got Blue";

break;

case green:

cout << "You got Green";

break;

case yellow:

cout << "You got Yellow";

break;

case orange:

cout << "You got Orange";

break;

case red:

cout << "You got Red";

break;

default:

cout << "Invalid Number !!\n";

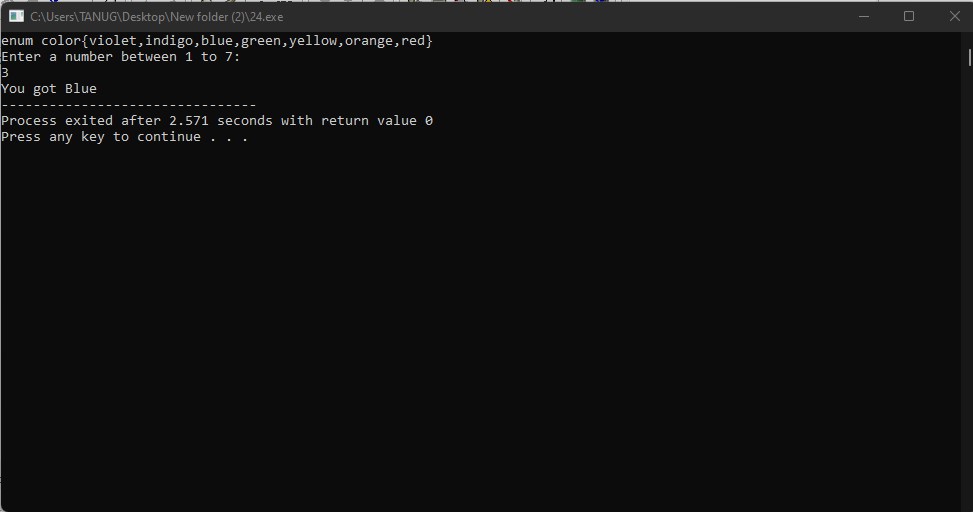
goto Repeat;

}

return 0;

}

**OUTPUT:**

****

**DATE:3/2/2022**

**15. Write a program to demonstrate “const pointer”**

**CODE:**

**#include <iostream>**

**using namespace std;**

**int main ()**

**{**

**int x=111;**

**int \*const p=&x;**

**cout <<"( int x=111 )\n";**

**cout <<"( int \*const p=&x )\n";**

**cout <<"x = "<<x<<endl;**

**cout <<"\*p = "<<\*p<<endl;**

**cout <<"p = "<<p<<endl;**

**\*p=222;**

**cout <<"Now, Applying \*p=222 ...\n";**

**cout <<"\*p = "<<\*p<<endl;**

**cout <<"p = "<<p<<endl;**

**cout <<"In Constant Pointer the Variable's Value can be changed \nbut the Pointer's value (memory location)is Constant\n\n";**

**int y=123;**

**int z=893;**

**const int\* p2=&y;**

**cout <<"( int y=123 )\n";**

**cout <<"( int z=893 )\n";**

**cout <<"( const int\* p2=&y )\n";**

**cout <<"\*p2 = "<<\*p2<<endl;**

**cout <<"p2 = "<<p2<<endl;**

**p2=&z;**

**cout <<"(\*p2 = 893 will show error) So, Applying p2=&z ...\n";**

**cout <<"\*p2 = "<<\*p2<<endl;**

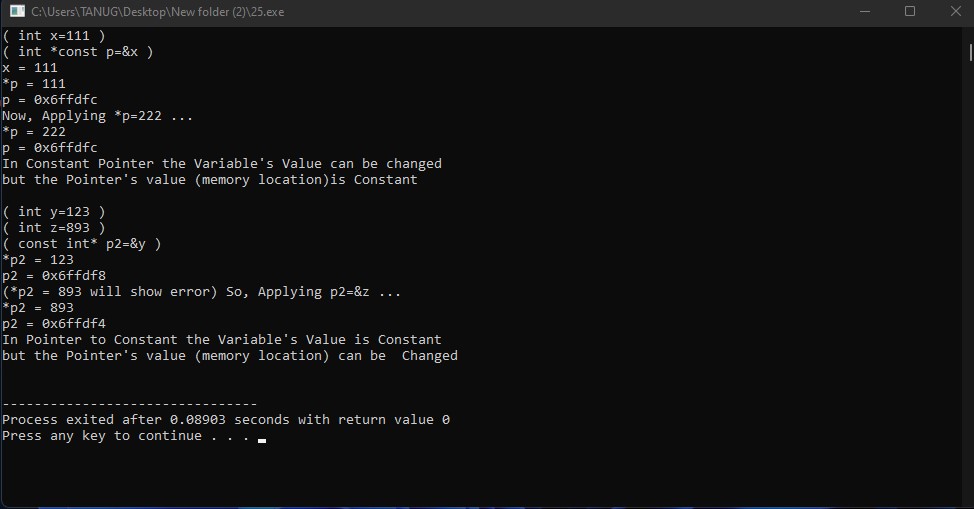
**cout <<"p2 = "<<p2<<endl;**

**cout <<"In Pointer to Constant the Variable's Value is Constant\nbut the Pointer's value (memory location) can be Changed\n\n";**

**return 0;**

**}**

**OUTPUT:**

****

**16.Write a program to demonstrate New and Delete**

**CODE:**

**#include <iostream>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int \*p;**

**float \*q;**

**char \*r;**

**char c;**

**p=new int;**

**q=new float;**

**r=new char;**

**cout<<"We have applied the 'new' to the pointer variables p, q and r"<<endl;**

**\*p=78;**

**\*q=86.04;**

**\*r='A';**

**cout <<"\*p = "<< \*p <<endl;**

**cout <<"\*q = "<< \*q <<endl;**

**cout <<"\*r = "<< \*r <<endl;**

**cout <<"Press F to free up the memory using 'delete'"<<endl;**

**cin>>c;**

**if(c == 'f' ||c == 'F')**

**{**

**delete p;**

**delete q;**

**delete r;**

**cout<<"Memory locations are deallocated Successfully ";**

**}**

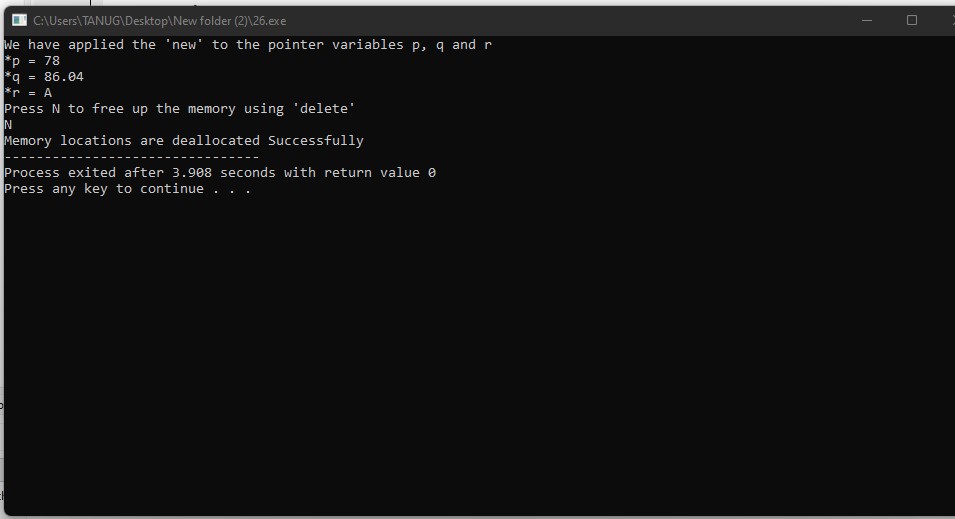
**else**

**cout <<"Program ended without freeing up the memory !!";**

**return 0;**

**}**

**OUTPUT:**

****

**17. Write a program to demonstrate new and delete operator in array.**

**CODE:**

**#include <iostream>**

**using namespace std;**

**int main ()**

**{**

**int \*p,size;**

**char c;**

**cout << "Enter the array size\n";**

**cin >> size;**

**cout<<"An integer array of size "<<size<<" through pointer p is created using 'new'";**

**p = new int[size];**

**cout << "So, Enter "<<size<<" Integers\n";**

**for (int i=0;i<size;i++)**

**cin >> \*(p+i);**

**cout << "Your given Integer Values are:\n";**

**for (int i=0;i<size;i++)**

**cout << \*(p+i) << endl;**

**cout <<"Press N to free up the memory using 'delete' operator"<<endl;**

**cin>>c;**

**if(c == 'n' ||c == 'N')**

**{**

**delete [] p;**

**cout<<"Memory locations are deallocated Successfully ";**

**}**

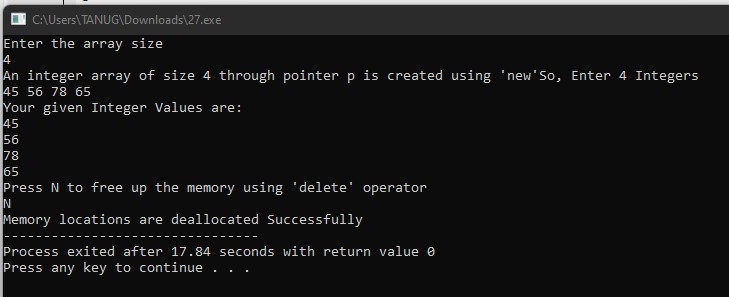
**else**

**cout <<"Program ended without freeing up the memory !!";**

**return 0;**

**}**

**OUTPUT:**

****

**18. Write a program to perform Multiplication and transpose operations**

**CODE:**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int a[3][3],b[3][3],c[3][3];**

**cout<<"Enter integers for 3x3 matrix (A):\n";**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**cin>>a[i][j];**

**}**

**cout<<"Enter integers for 3x3 matrix (B):\n";**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**cin>>b[i][j];**

**}**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**{**

**c[i][j]=0;**

**for(int k=0;k<3;k++)**

**c[i][j] +=a[i][k]\*b[k][j];**

**}**

**}**

**cout<<"Matrix Multiplication is :\n";**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

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

**cout<<endl;**

**}**

**cout<<"Now Transposing the above matrix we got:\n";**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**{**

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

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

**}**

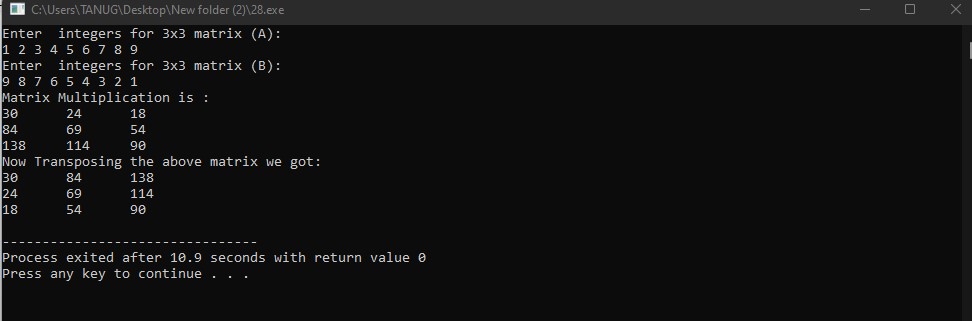
**cout<<endl;**

**}**

**return 0;**

**}**

**OUTPUT:**

****

**Submitted By :-**

**Name :**

**Roll No : 21241000**