DEPARTMENT OF COMPUTER APPLICATIONS

**MAHARAJA SURAJMAL INSTITUTE**

C-4, Janakpuri, New Delhi, 110058



**C++ PRACTICAL FILE**

SUBJECT CODE – BCA

**PRACTICAL- C++ LAB**

**Submitted to:** **Submitted by:**

Ms. Vinita Tomar Name: Deepanshu Arya

Assistant Professor Course: BCA 3-A

Department of Computer Enrolment No: 04014902021

Applications

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment Number | Topic | Page Number | Teacher’s Signature |
|  |  |  |  |
|  | WAP to perform insertion /Deletion of element in an array |  |  |
|  | WAP to perform following string related function on an array  a. Finding string length  b. Concatenation of two strings  c. Comparing two string  d. Copy one string to another |  |  |
|  | WAP to perform sorting in Array:-  a. Bubble sort  b. Write an algorithm for sorting an array of integer using Selection Sort Technique and Sort the following numbers in ascending order using the selection sort algorithm (Show all the passes) : {15,4,23,32,87,43,8,72}  c. Insertion sort  d. Merging two sorted array |  |  |
|  | WAP to perform searching in array:-  a. Linear search  b. Binary Search |  |  |
|  | WAP to Display the following on sparse matrix: -  a. Upper triangular matrix  b. Lower triangular matrix  c. Diagonal matrix  d. Tri-diagonal matrix |  |  |
|  | WAP to Store 2D Sparse Matrix into One D Arrays |  |  |
|  | WAP to perform the following on sparse matrix: -  a. Sparse matrix to tuple form  b. Transpose of sparse matrix  c. Add two sparse matrices  d. Subtract two sparse matrices |  |  |

**Q1.** WAP to calculate factorial of a number.

**Code :-**

#include<iostream>

using namespace std;

int main(){

int num,fact=1;

cout<<"Enter number : ";

cin>>num;

for(int i=num;i>=1;i--){

fact=fact\*i;

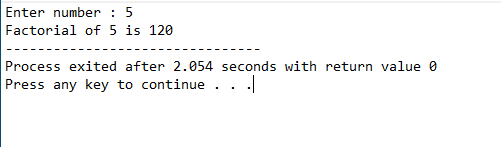
}

cout<<"Factorial of "<<num<<" is "<<fact;

return 0;

}

**Output :-**



**Q2.** WAP to print fibbonacci numbers using recursion.

**Code :-**

#include<iostream>

using namespace std;

int main(){

int num,fibbo=1,num1=1,temp;

cout<<"Enter till how many number you have to print series : ";

cin>>num;

cout<<"Series is :"<<endl;

cout<<" 1 , ";

for(int i=2;i<=num;i++){

cout<<fibbo<<" , ";

temp=fibbo;

fibbo=fibbo+num1;

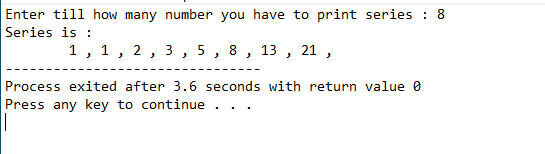
num1=temp;

}

return 0;

}

**Output :-**



**Q3.** WAP to calculate area and perimeter of a circle by making a function that returns more than one value.

**Code :-**

#include <iostream>

#define PI 3.14

using namespace std;

int find\_area\_peri(float radi, float& peri, float& area);

int main() {

float radi, peri, area;

cout << "Enter radius of circle : ";

cin >> radi;

find\_area\_peri(radi, peri, area);

cout << "Perimeter : " << peri << endl;

cout << "Area : " << area << endl;

return 0;

}

int find\_area\_peri(float radi, float& peri, float& area) {

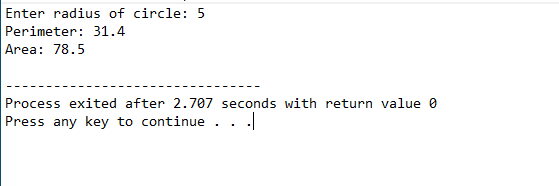
peri = 2 \* PI \* radi;

area = PI \* radi \* radi;

return 0;

}

**Output :-**



**Q4.** WAP to show the concept of function overloading to find the volume of cone, sphere, and cylinder.

**Code :-**

#include#include <iostream>

#define PI 3.14

using namespace std;

float volume\_of\_sphere(float radius) {

return 4.0/3.0\*PI\*radius\*radius\*radius;

}

float volume\_of\_cylinder(float radius, float height) {

return PI\*radius\*radius\*height;

}

float volume\_of\_cone(float radius, int height) {

return 1.0/3.0\*PI\*radius\*radius\*height;

}

int main(){

float radius,height;

int height\_int;

cout<<"Program for calculating volumes : "<<endl<<endl;

cout<<"Enter radius for sphere,cylinder,cone: ";

cin>>radius;

cout<<"Enter height for cylinder,cone: ";

cin>>height;

height\_int = (int)height;

cout<<">->->->- VOLUMES -<-<-<-<"<<endl;

cout<<"Sphere: \t"<<volume\_of\_sphere(radius)<<" cube units"<<endl;

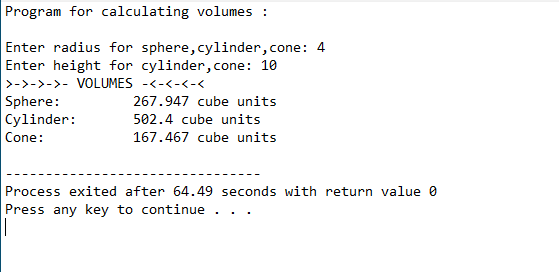
cout<<"Cylinder: \t"<<volume\_of\_cylinder(radius, height)<<" cube units"<<endl;

cout<<"Cone: \t\t"<<volume\_of\_cone(radius, height\_int)<<" cube units"<<endl;

return 0;

}

**Output :-**



**Q5.** WAP to calculate the area of a circle using inline function : -

**Code :-**

#include <iostream>

#define PI 3.14

using namespace std;

inline float area\_circle(float radius);

int main() {

float radius;

cout<<"Enter radius of circle: ";

cin>>radius;

cout<<"Area : "<<area\_circle(radius)<<" sq. units"<<endl;

return 0;

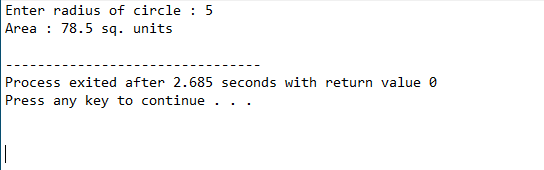
}

inline float area\_circle(float radius){

return PI\*radius\*radius;

}

**Output :-**



**Q6.** WAP to demonstrate the use of default arguments to calculate the volume of a box.

**Code :-**

#include <iostream>

using namespace std;

float volume(float l=10, float b=10, float h=10);

int main() {

float l, b, h;

cout << "Enter length (default 10) of Box : ";

cin >> length;

cout << "Enter breadth (default 10) of Box : ";

cin >> breadth;

cout << "Enter height (default 10) of Box : ";

cin >> height;

cout << "Volume with all default arguments : " << volume() << endl;

cout << "Volume with only length specified : " << volume(l) << endl;

cout << "Volume with both length and breadth : " << volume(l, b)<< endl;

cout << "Volume with all arguments : " << volume(l,b, h)<< endl;

return 0;

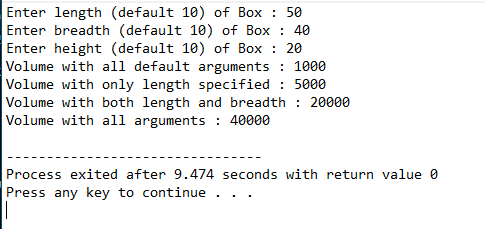
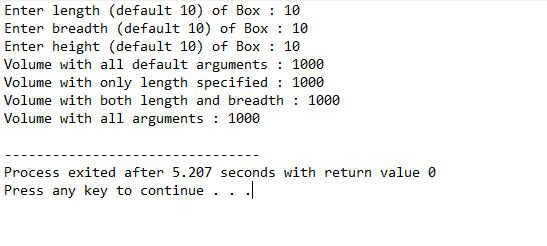
}

float volume(float l, float b, float h){

return l\*b\*h;

}

**Output :-**



**Q7.** WAP to calculate mean of n elements of an array.

**Code :-**

#include <iostream>

using namespace std;

int main(){

int size,sum=0;

cout<<"Enter size of array : ";

cin>>size;

int \*arr=new int[size];

cout<<"Enter value of array elements : "<<endl;

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

cin>>arr[i];

sum+=arr[i];

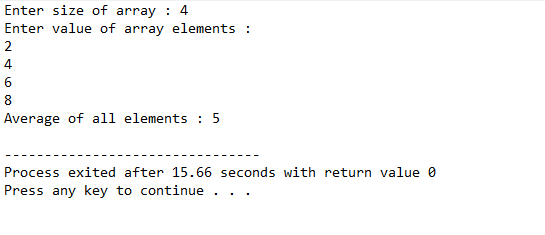
}

cout<<"Average of all elements : "<<(float) sum/size<<endl;

return 0;

}

**Output :-**



**Q8.** WAP to find the transpose of a matrix.

**Code :-**

#include <iostream>

using namespace std;

int main() {

int a[10][10], transpose[10][10], row, column, i, j;

cout << "Enter rows and columns of matrix: ";

cin >> row >> column;

cout << "\nEnter elements of matrix: " << endl;

// Storing matrix elements

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

for (int j = 0; j < column; ++j) {

cout << "Enter element a" << i + 1 << j + 1 << ": ";

cin >> a[i][j];

}

}

// Printing the a matrix

cout << "\nEntered Matrix: " << endl;

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

for (int j = 0; j < column; ++j) {

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

if (j == column - 1)

cout<<endl<<endl;

}

}

// Computing transpose of the matrix

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

for (int j = 0; j < column; ++j) {

transpose[j][i] = a[i][j];

}

// Printing the transpose

cout << "\nTranspose of Matrix: " << endl;

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

for (int j = 0; j < row; ++j) {

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

if (j == row - 1)

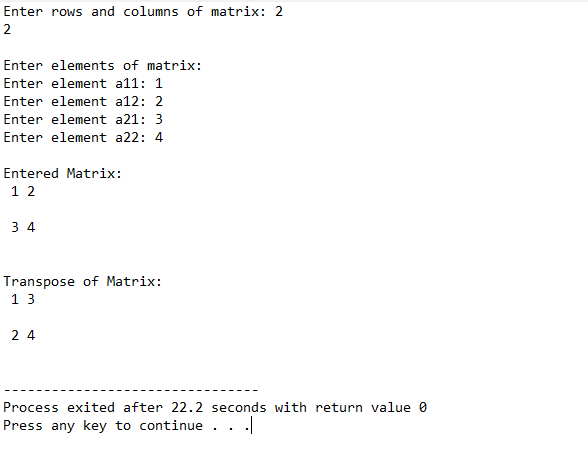
cout << endl << endl;

}

return 0;

}

**Output :-**



**Q9.** WAP to add two matrices using functions.

**Code :-**

#include<iostream>

using namespace std;

int main(){

int m,n,p,q,i,j,A[5][5],B[5][5],C[5][5];

cout << "Enter rows and column of matrix A : ";

cin>>m>>n;

cout<<"Enter rows and column of matrix B : ";

cin>>p>>q;

if((m!=p) && (n!=q)){

cout<<"Matrices cannot be added!";

exit(0);

}

cout<<"Enter elements of matrix A : ";

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

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

cin>>A[i][j];

cout<<"Enter elements of matrix B : ";

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

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

cin>>B[i][j];

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

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

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

cout<<"Sum of matrices\n";

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

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

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

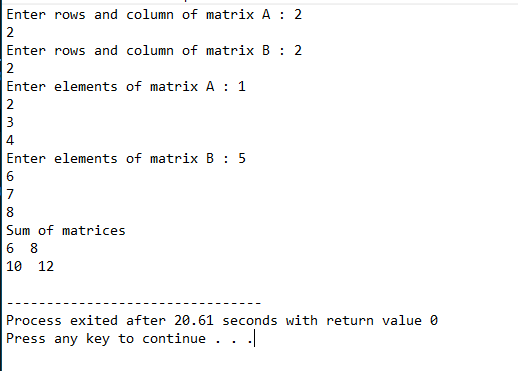
cout<<"\n";

}

return 0;

}

**Output :-**



**Q10.** WAP to check whether a given string is a palindrome or not.

**Code :-**

#include <iostream>

using namespace std;

int check\_palindrome(string str){

int n = str.length();

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

if (str[i] != str[n-i-1])

return 0;

}

return 1;

}

int main() {

string str;

cout<<"Enter a string: ";

cin>>str;

if(check\_palindrome(str))

cout<<"String is palindrome."<<endl;

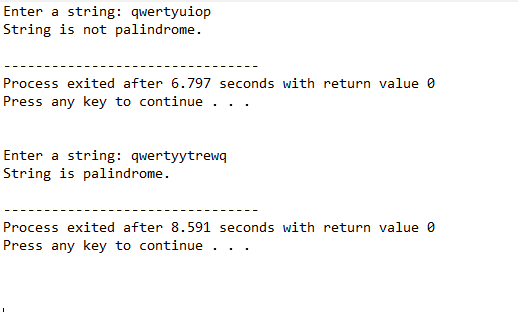
else

cout<<"String is not palindrome."<<endl;

return 0;

}

**Output :-**



**Q11.** WAP to calculate average of n numbers of an array by allocation and de- allocation of memory using new and delete keywords..

**Code :-**

#include <iostream>

using namespace std;

int main(){

int size,sum=0;

cout<<"Enter size of array : ";

cin>>size;

int \*arr=new int[size];

cout<<"Enter value of array elements: "<<endl;

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

cin>>arr[i];

sum+=arr[i];

}

cout<<"Average of all elements : "<<(float) sum/size<<endl;

delete [] arr;

return 0;

}

**Output :-**

