**UCS540**

**Data Structures and Algorithms**

Submitted by: Ritik Singh (102104089)

Class: 3EE3

**Lab Assignment 1**

**Objective: Students should be able to write programs in ‘C’ language. Simple programming exercise on various programming constructs in ‘C’ follows.**

1. **Write a program in C to convert miles into kilometers (Km). Hint: 1 Mile=1.609 Km. [Use macros, relevant name and types for variables].**

**Code:**

// Online C compiler to run C program online

#include <stdio.h>

#define miles\_to\_km\_factor 1.609

int main() {

// Write C code here

double miles,kilometers;

printf("enter distance in miles");

scanf("%lf", &miles);

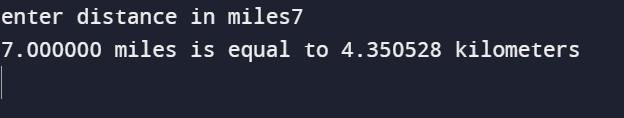
kilometers=miles/1.609;

printf("%lf miles is equal to %lf kilometers \n",miles,kilometers);

return 0;

}

**OUTPUT:**

****

1. **Write a program to find the number of positive, negative and zeros in a sequence of inputs (numbers) entered as data.**

**Code:**

#include <iostream>

using namespace std;

int main() {

int pos=0,neg=0,zeros=0;int size=0;

cout<<"Enter size"<<endl;

cin>>size;

int arr[size];

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

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

cin>>arr[i];

}

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

if(arr[i]<0)

neg++;

else if(arr[i]>0)

pos++;

else if(arr[i]==0)

zeros++;

}

cout<<"Number of positive number: "<<pos<<endl;

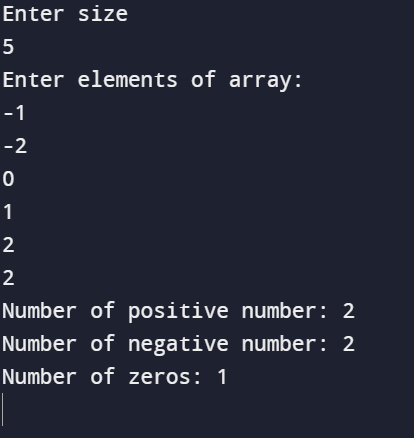
cout<<"Number of negative number: "<<neg<<endl;

cout<<"Number of zeros: "<<zeros<<endl;

return 0;

}

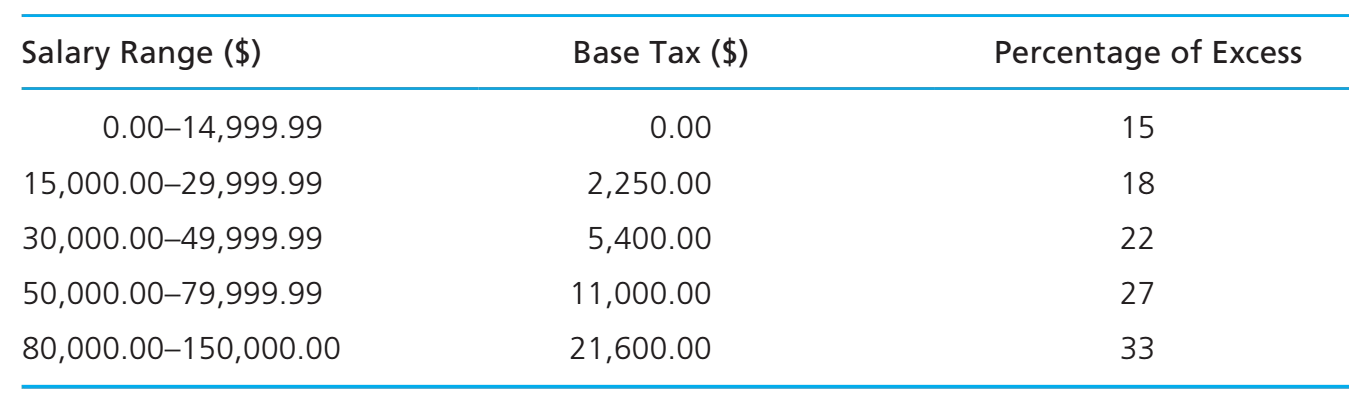
**OUTPUT:**

****

1. **Compute the tax due based on a tax table given below:**

**Program Input: Salary amount.**

**Program Output: Returns the tax due for 0.0 <= salary <= 150,000.00; returns -1.0 if salary is outside the table range.**



**Code:**

#include<iostream>

using namespace std;

int main(){

double calculateTax(double salary) {

if (salary >= 0.0 && salary <= 14999.99) {

return 0.15 \* salary;

}

else if (salary >= 15000.00 && salary <= 29999.99) {

return 2250.00 + 0.18 \* (salary - 15000.00);

}

else if (salary >= 30000.00 && salary <= 49999.99) {

return 5400.00 + 0.22 \* (salary - 30000.00);

7

}

else if (salary >= 50000.00 && salary <= 79999.99) {

return 11000.00 + 0.27 \* (salary - 50000.00);

}

else if (salary >= 80000.00 && salary <= 150000.00) {

return 21600.00 + 0.33 \* (salary - 80000.00);

}

else {

return -1.0;

}

}

int main() {

double salary;

cout<<"Enter the salary \n";

cin>>salary;

double tax = calculateTax(salary);

if (tax == -1.0) {

printf("Error: Salary is outside the table range.\n");

} else {

printf("Tax due: $%.2f\n", tax);

}

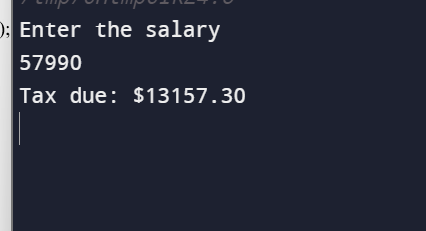
return 0;

}

return 0;

}

**OUTPUT:**

****

1. **Write an interactive program (menu driven) in ‘C’ (using functions) to compute the area of a selected geometrical figure from a list of such figures (square, rectangle, and circle).**

**Code:**

#include<iostream>

using namespace std;

double calculateSquareArea(double side){

return side \*side;

}

double calculateRectangleArea(double length,double width){

return length\*width;

}

double calculateCircleArea(double radius){

return 3.14\*radius\*radius;

}

int main(){

int choice;

double side,length,width,radius,area;

do{

cout<<"choose a geometrical figure:\n";

cout<<"1. square"<<endl;

cout<<"2. rectangle"<<endl;

cout<<"3. circle"<<endl;

cout<<"4. exit"<<endl;

cout<<"enter your choice(1-4)"<<endl;

cin>>choice;

switch(choice){

case 1:

cout<<"enter side length of square:\n";

cin>>side;

area=calculateSquareArea(side);

cout<<"area of square:\n"<<area;

break;

case 2:

cout<<"enter length of rectangle:\n";

cin>>length;

cout<<"enter width of rectangle:\n";

cin>>width;

area=calculateRectangleArea(length,width);

cout<<"area of rectangle:\n"<<area;

break;

case 3:

cout<<"enter radius of circle:\n";

cin>>radius;

area=calculateCircleArea(radius);

cout<<"area of circle:\n"<<area;

break;

case 4:

cout<<"exiting the progam\n";

break;

default:

cout<<"invalid choice, enter again:\n";

break;

}

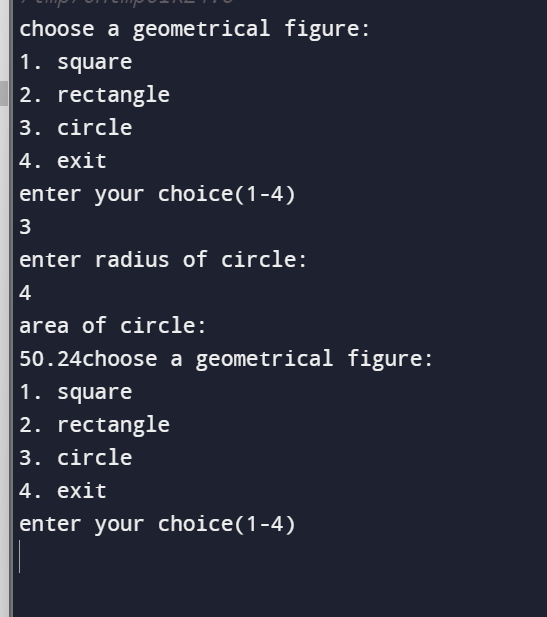
}

while(choice!=4);

return 0;

}

**OUTPUT:**

****

1. **Write a program to display first n elements of Fibonacci series.**

**Code:**

// Online C++ compiler to run C++ program online

#include <iostream>

using namespace std;

int main() {

// Write C++ code here

int n;

cin>>n;

int t1=0;

int t2=1;

int nextterm;

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

cout<<t1<<endl;

nextterm=t1+t2;

t1=t2;

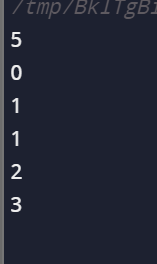
t2=nextterm;

}

return 0;

}

**OUTPUT:**

****

1. **Write a program to print a table book from Table X to Table Y. X and Y are user inputs.**

**Code:**

// Online C++ compiler to run C++ program online

#include <iostream>

using namespace std;

int main() {

int x,y;

cout<<"enter the starting table value:\n";

cin>>x;

cout<<"enter the ending table value:\n";

cin>>y;

cout<<"multiplication table from table "<<x<<" to table "<<y<<endl;

for(int i=1;i<=10;i++){

for(int j=x;j<=y;j++){

cout<<j<<" x"<<i<<" = "<< j\*i<<" ";

}

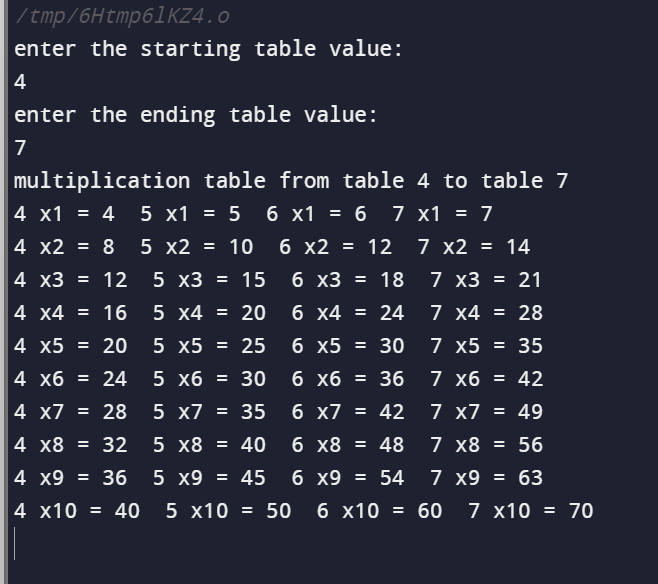
cout<<endl;

}

return 0;

}

**OUTPUT:**

****

1. **Write a program to compute factorial of a number using iterative approach.**

**Code:**

#include<iostream>

using namespace std;

int main(){

int n;

int fact=1;

cout<<"enter the no. whose factorial needs to be calculated\n";

cin>>n;

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

fact=fact\*i;

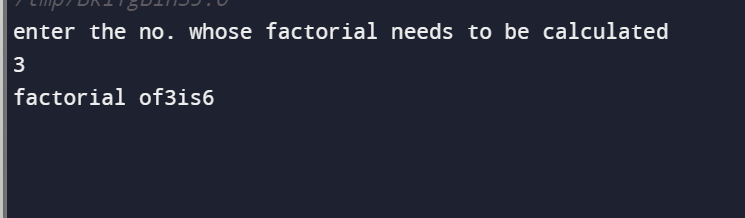
}

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

return 0;

}

**OUTPUT:**

****

1. **Write a program to swap two numbers using functions.**

**Code:**

#include<iostream>

using namespace std;

void swap(int ,int);

int main(){

int a,b;

cout<<"enter 2 no. to be swapped\n";

cin>>a>>b;

cout<<"no. after swapping are:";

swap(a,b);

return 0;

}

void swap(int x,int y){

int z;

z=x;

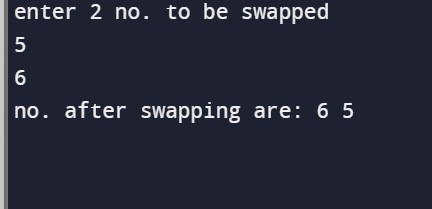
x=y;

y=z;

cout<<" "<<x<<" "<<y;

}

**OUTPUT:**

****

1. **Write a function that returns the first integer between n\_min and n\_max entered as data to the calling function (main).**

**Code:**

#include <iostream>

using namespace std;

int findFirstInteger(int n\_min, int n\_max) {

for (int i = n\_min; i <= n\_max; ++i) {

if (i % 1 == 0) {

return i;

}

}

return -1;

}

int main() {

int n\_min = 5;

int n\_max = 15;

int result = findFirstInteger(n\_min, n\_max);

if (result != -1) {

cout << "The first integer between " << n\_min << " and " << n\_max << " is: " << result << endl;

} else {

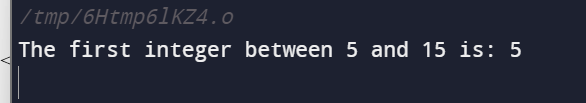
cout << "No integer found between " << n\_min << " and " << n\_max << endl;

}

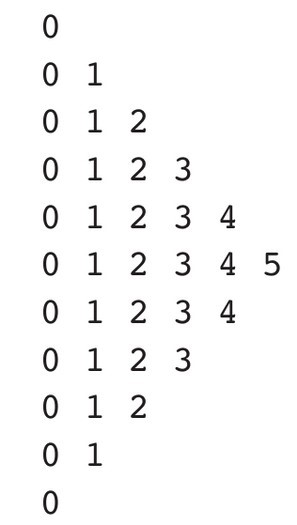
return 0;

}

**OUTPUT:**



**10) Write nests of loops that cause the following output to be displayed.**

****

**Code:**

#include<iostream>

using namespace std;

int main(){

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

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

cout<<j<<" ";

}

cout<<endl;

}

for(int i =4;i>=0;i--){

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

cout<<j<<" ";

}

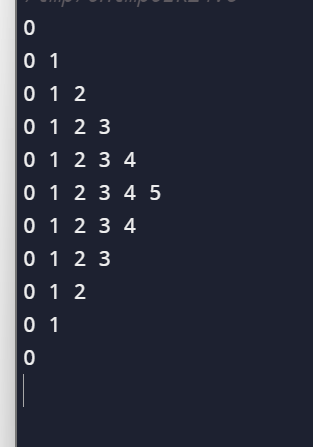
cout<<endl;

}

return 0;

}

**OUTPUT:**

****