

Assignment

NED University of Engineering & Tech.  
Spring Semester-2021

Computers & Programming (EE-163)

Department of Electrical Engineering  
FE-EE

# Assignment Solution

Name	<b>Irsa Sarfaraz</b>
Roll Number	<b>EE-20165</b>
Section	<b>D</b>
Assignment No.	<b>Assignment # 01</b>

Don't write below this line

---

Marks Obtained	
----------------	--

Status	
--------	--

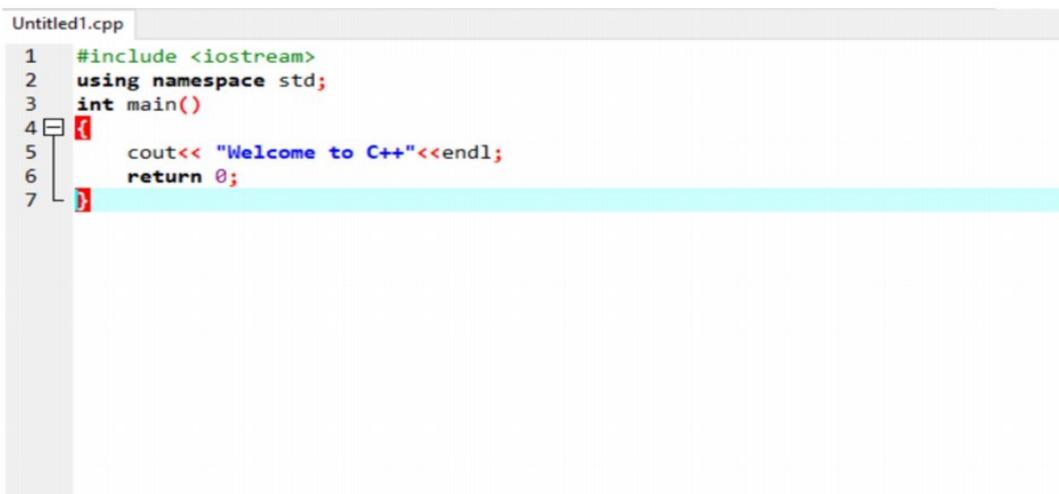
Signature: \_\_\_\_\_

# ASSIGNMENT

Q1: Write a program to print the following text on the console screen.

Welcome to C++

## CODE:



```
Untitled1.cpp
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout<< "Welcome to C++" << endl;
6     return 0;
7 }
```

## OUTPUT:



```
Welcome to C++
-----
Process exited after 0.05088 seconds with return value 0
Press any key to continue . . .
```

**Q2:** Using escape sequences to print the text in following fashion.

Welcome  
to  
C++

## CODE:

```
Untitled1.cpp
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout<< "Welcome" << "\n" << "to" << "\nC++" << endl;
6     return 0;
7 }
```

## OUTPUT:

```
Welcome
to
C++
-----
Process exited after 0.05207 seconds with return value 0
Press any key to continue . . .
```

**Q3:** Write a program that takes two integer numbers from the user and print their sum on console screen.

## CODE:

```
1 #include<iostream> //header file of input output stream
2 using namespace std;
3 int main() //pre-defined function of c++
4 {
5     //variable declaration
6     int num1, num2;
7
8
9     //input phase
10    cout << "Enter first integer ";
11    cin >> num1;
12    cout << "Enter second integer ";
13    cin >> num2;
14
15
16    //basic arithmetic operation
17    cout << "Sum of above two integer inputs are " << num1 + num2 << endl;
18
19    return 0;
20 }
```

## OUTPUT:

```
Enter first integer 5
Enter second integer 10
Sum of above two integer inputs are 15
-----
Process exited after 7.858 seconds with return value 0
Press any key to continue . . .
```

**Q4:** What are Arithmetic operators? Write the rule of precedence for the Arithmetic operators.

## ARITHMETIC OPERATORS:

Arithmetic operators are used to perform arithmetic operations on variables and data .Arithmetic operators compute the result of specific arithmetic operation and returns its result. The arguments are not modified. There are various arithmetic operators in C++.

- Addition(+)
- Sustration(-)
- Multiplication(\*)
- Division(/)
- Modulo operator (%)

## RULE OF PRECEDENCE IN ARITHMETIC OPERATORS:

The order of precedence of arithmetic operators in C++ is:

- Multiplication (\*), division (/), modulus (%)
- Addition(+), subtraction(-)

**Q5:** How C++ program is compiled into an executable?

## C++ COMPILED INTO AN EXECUTABLE:

The compilation of a C++ program involves three steps: Preprocessing: the preprocessor takes a C++ source code file and deals with the #includes, #define s and other preprocessor directives. Linking: the linker takes the object files produced by the compiler and produces either a library or an executable file.

**Q6:** Write a program that take two numbers and operation to perform on those numbers as input from the user and print the result of operation on console screen.  
(Hint: Use if-statements to identify the required operation)

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int num1,num2;
6     char ope;
7     cout<< "Enter fisrt number" << endl;
8     cin>>num1;
9     cout<< "Enter the second number" << endl;
10    cin>>num2;
11    cout<<"Enter the operation to be performed on that numbers" << endl;
12    cin>>ope;
13    if(ope== '+')
14    {
15        cout<< "num1 + num2 =" << " " << num1+num2 << endl;
16    }
17    if(ope== '-')
18    {
19        cout<<"num1 - num2 =" << " " << num1-num2 << endl;
20    }
21    if(ope== '*')
22    {
23        cout<<"num1 * num2 =" << " " << num1*num2 << endl;
24    }
25    if(ope== '/')
26    {
27        cout<<"num1 / num2 =" << " " << num1/num2 << endl;
```

## OUTPUT:

```
Enter fisrt number
64
Enter the second number
8
Enter the operation to be performed on that numbers
/
num1 / num2 = 8

-----
Process exited after 4.919 seconds with return value 0
Press any key to continue . . .
```

**Q7:** Write a program that asks user to enter two integers, obtains the numbers from the user, and then prints the larger number followed by the words “is larger.” If the numbers are equal, print the message “These numbers are equal.”

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int num1,num2;
6     cout<< "Enter any two numbers" << endl;
7     cin>>num1>>num2;
8     if(num1 > num2)
9     {
10         cout<<num1<<" <<"is larger" << endl;
11     }
12     if(num2 > num1)
13     {
14         cout<<num2<<" <<"is larger" << endl;
15     }
16     if(num1 == num2 )
17     {
18         cout<<"These numbers are equal";
19     }
20     return 0;
21 }
```

### Output case 1:

```
Enter any two numbers
84
56
84 is larger
```

### Output case 2:

```
Enter any two numbers
20
20
These numbers are equal
```

**Q8:** Write a program that inputs three integers from the keyboard and prints the sum, average, product, smallest and largest of these numbers. The screen dialog should appear as follows:

```
Input three different integers: 13 27 14
Sum is 54
Average is 18
Product is 4914
Smallest is 13
Largest is 27
```

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int a,b,c;
6     cout<<"Input three different integers"<<endl;
7     cin>>a>>b>>c;
8     cout<<"Sum is"<< " "<<a+b+c<<endl;
9     cout<<"Average is"<< " "<<(a+b+c)/3<<endl;
10    cout<<"Product is"<< " "<<a*b*c<<endl;
11    if(a<b && a<c)
12    {
13        cout<<"Smallest is"<< " "<<a<<endl;
14    }
15    if(b<a && b<c)
16    {
17        cout<<"Smallest is"<< " "<<b<<endl;
18    }
19    if(c<a && c<b)
20    {
21        cout<<"Smallest is"<< " "<<c<<endl;
22    }
23    if(a>b && a>c)
24    {
25        cout<<"Largest is"<< " "<<a<<endl;
26    }
27    if(b>a && b>c)
```

```
28 {
29     cout<<"Largest is"<< " "b<<endl;
30 }
31 if(c>a && c>b)
32 {
33     cout<<"Largest is"<< " "c<<endl;
34 }
35 return 0;
36 }
```

## OUTPUT:

```
Input three different integers
1
2
3
Sum is 6
Average is 2
Product is 6
Smallest is 1
Largest is 3
```

**Q9:** Write a program that reads an integer and determines and prints whether its odd or even.

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int num;
6     cout<<"Input the number"<<endl;
7     cin>>num;
8     if(num%2==0)
9     {
10         cout<<"The number is even"<<endl;
11     }
12     else
13     {
14
15         cout<<"The number is odd"<<endl;
16     }
17     return 0;
18 }
```

## Output case 1:

```
Input the number  
64  
The number is even
```

## Output case 2:

```
Input the number  
87  
The number is odd
```

**Q10:** Write a program that inputs a five digit integer, separates the integer into its digit and prints them separated by three spaces each. For example, if the user types in 42339, the program should print:

4        2        3        3        9

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int num;
6     cout<<"Enter five digit integer:";
7     cin>>num;
8
9     cout<<num/10000<<" ";
10    num=num%10000;
11    cout<<num/1000<<" ";
12    num=num%1000;
13    cout<<num/100<<" ";
14    num=num%100;
15    cout<<num/10<<" ";
16    num=num%10;
17    cout<<num;
18
19    return 0;
20 }
```

## OUTPUT:

```
Enter five digit integer:45678
4 5 6 7 8
-----
Process exited after 8.592 seconds with return value 0
Press any key to continue . . .
```

**Q11:** Develop a C++ program that uses a `while` statement to determine the gross pay for each of several employees. The company pays “straight time” for the first 40 hours worked by each employee and pays “time-and-a-half” for all hours worked in excess of 40 hours. You are given a list of the employees of the company, the number of hours each employee worked last week and the hourly rate of each employee. Your program should input this information for each employee and should determine and display the employee's gross pay.

**Sample Output:**

```
Enter hours worked (-1 to end): 39
Enter hourly rate of the employee: 10.00
Salary is 390.00 Rs.
```

```
Enter hours worked (-1 to end): 40
Enter hourly rate of the employee: 10.00
Salary is 400.00 Rs.
```

```
Enter hours worked (-1 to end): 41
Enter hourly rate of the employee: 10.00
Salary is 415.00 Rs.
```

```
Enter hours worked (-1 to end): -1
```

## CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     double hours,rate,pay;
6     cout<< "Enter hours worked (-1 to end):"<<endl;
7     cin>>hours;
8     while(hours>0)
9     {
10        cout<<"Enter hourly rate of the employee:"<<endl;
11        cin>>rate;
12        if(hours<=40)
13        {
14            pay=(hours*rate);
15        }
16        if(hours>40)
17        {
18            pay=(40*rate)+((hours-40)*(rate*1.5));
19        }
20        cout<< "Salary is"<<pay<<"Rs"<<endl;
21        cout<< "\nEnter hours worked (-1 to end):"<<endl;
22        cin>>hours;
23    }
24    return 0;
25 }
```

## OUTPUT CASE 1:

```
Enter hours worked (-1 to end):  
40  
Enter hourly rate of the employee:  
10  
Salary is400Rs  
  
Enter hours worked (-1 to end):
```

## OUTPUT CASE 2:

```
Enter hours worked (-1 to end):  
41  
Enter hourly rate of the employee:  
10  
Salary is415Rs  
  
Enter hours worked (-1 to end):  
-
```

**Q12:** Write a program that ask user to enter an integer number and evaluates its factorial. Your program should print the output as below,

```
Enter an integer : 5  
5 x 4 x 3 x 2 x 1 = 120
```

## CODE:

```
1 #include <iostream>  
2 using namespace std;  
3 int main()  
4 {  
5     int a,b,product;  
6     product= 1;  
7     cout<<"Input any integer:"<<endl;  
8     cin>>a;  
9     for(int i=5; i>=1; i--)  
10    {  
11        product= product*i;  
12        cout<<i<<"*";  
13    }  
14    cout<<"\b="<<product;  
15    return 0;  
16}  
17
```

## OUTPUT:

```
Input any integer:
```

```
5
```

```
5*4*3*2*1=120
```

**Q13:** Write a program that ask user to input the number of elements in a Fibonacci sequence and then generates a Fibonacci sequence up-to the given number of elements.

(Hint: In Fibonacci sequence, the next element is the sum of two previous values)

Sample Output:

```
Enter number of elements: 10
0 1 1 2 3 5 8 13 21 34
```

## CODE:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int first=1;
6     int prev=0;
7     int x;
8     int sum;
9     cout<<"Enter number of elements:";
10    cin>>x;
11    cout<<prev<<" ";
12
13    for(int i=2; i<=x; i++)
14    {
15
16        cout<<first<<" ";
17        sum=first+prev;
18        prev=first;
19        first=sum;
20
21    }
22
23    return 0;
24
25
26
27 }
```

## OUTPUT:

```
Enter number of elements:10
0 1 1 2 3 5 8 13 21 34
-----
Process exited after 4.416 seconds with return value 0
Press any key to continue . . .
```

**Q14:** Write a program that reads three non-zero double values and determines and prints whether they could represents sides of triangle.

[Hint:  $a, b$  and  $c$  represents sides of triangle if following criteria is met,

$$\begin{aligned}a + b &> c \\a + c &> b \\b + c &> a\end{aligned}$$

Sample Output:

```
Enter length of three sides: 3 4 5
They are sides of triangle.

Enter length of three sides: 2 2 5
They are not sides of triangle.

Enter length of three sides: 2.4 3.8 5.5
They are sides of triangle.
```

## CODE:

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4
5 int main()
6 {
7     double a,b,c;
8     for(int i=1; i<=3; i++)
9     {
10         cout<<"enter three values ";
11         cin>>a>>b>>c;
12
13
14     if(a + b > c&& a + c > b&& | b + c > a)
15     {
16         cout<<"They are sides of triangle.\n"<<endl;
17     }
18
19     else
20     {
21         cout<<"they are not sides of triangles.\n"<<endl;
22     }
23
24
25     }
26     return 0;
27 }
```

## OUTPUT:

```
enter three values 3 4 5
They are sides of triangle.

enter three values 2 2 5
they are not sides of triangles.

enter three values 2.4 3.8 5.5
They are sides of triangle.

-----
Process exited after 27.65 seconds with return value 0
Press any key to continue . . .
```

**Q15:** Write a program that reads three non-zero double values and determines and prints whether they are sides of right triangle. The program should verify the results up to 4 decimal places.

[Hint: Use Pythagoras theorem to determine whether the three sides form right triangle.]

Sample Output:

```
Enter length of three sides: 3 4 5
The sides represents right triangle.

Enter length of three sides: 4 5 6.403
The sides don't represents right triangle.

Enter length of three sides: 4 5 6.4031
The sides represents right triangle.
```

## CODE:

```
1 #include <iostream>
2 #include<cmath>
3 using namespace std;
4 int main()
5 {
6     double a,b,c;
7     for( int i=1; i<=3; i++)
8     {
9
10    cout<<"Enter length of three sides:"<<endl;
11    cin>>a>>b>>c;
12    if(pow(a,2)==pow(b,2)+pow(c,2) || pow(b,2)==pow(a,2)+pow(c,2) || pow(c,2)==pow(b,2)+pow(a,2))
13    {
14        cout<<"The sides represent right triangle."<<endl;
15    }
16    else
17    {
18        cout<< "The sides don't represents right triangle."<<endl;
19    }
20 }
21 return 0;
22 }
```

## OUTPUT:

```
Enter length of three sides:  
3  
4  
5  
The sides represent right triangle.  
Enter length of three sides:  
4  
5  
6.403  
The sides don't represents right triangle.  
Enter length of three sides:  
4  
5  
6.4031  
The sides don't represents right triangle.  
-----  
Process exited after 22.85 seconds with return value 0  
Press any key to continue . . .
```

**Q16:** Write a program that ask user to input a floating point number and computes exponential of that number using Taylor series as below,

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Also, prompt the user for desired accuracy of  $e$  (i.e., the number of terms in summation).

Sample Output:

```
Enter a value whose exponential needs to be evaluated: 1  
Enter number of terms for evaluation: 20  
Result is: 2.71828
```

## CODE:

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4 int main()
5 {
6
7     double x;
8     int n;
9     int fact=1;
10    double exp=1.0;
11
12    cout<<"Enter a value whose exponential needs to be evaluated: ";
13    cin>>x;
14    cout<<"Enter number of terms for evaluation: ";
15    cin>>n;
16
17    for(int i=1;i<=n;i++)
18    {
19        fact=fact*i;
20        exp=exp+(pow(x,i))/fact;
21    }
22
23    cout<<"Result is: "<<exp;
24
25    return 0;
26
27 }
```

## OUTPUT:

```
Enter a value whose exponential needs to be evaluated: 1
Enter number of terms for evaluation: 20
Result is: 2.71828
-----
Process exited after 46.92 seconds with return value 0
Press any key to continue . . .
```

**Q17:** Write a program that ask user to input angle in radians and computes its sine using Taylor series as below,

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

Also, prompt the user for desired accuracy of *sine*. (i.e., the number of terms in summation).

Sample Output:

```
Enter a value for sin evaluation: 2
Enter number of terms in the summation: 10
Result is: 0.909297
```

## CODE:

```
1 #include<iostream>
2 #include<cmath>
3 u C:\Program Files (x86)\Dev-Cpp\MinGW64\lib\gcc\x86_64-w64-mingw32\4.9.2\include
4 int main()
5 {
6     double x,a,b;
7     int n;
8     int fact=1;
9     double sinx=0.0;
10
11
12    cout<<"Enter a value for sine evaluation: ";
13    cin>>x;
14    cout<<"Enter number of terms in the summation: ";
15    cin>>n;
16
17    for(int i=1;i<=n;i++)
18    {
19        fact*=(2*i)+1;
20        a=pow(-1,i);
21        b=pow(x,(2*i)+1);
22        sinx=sinx+(a*b)/fact;
23    }
24    cout<<"Result is: "<<sinx;
25
26    return 0; |
27 }
```

## OUTPUT:

```
Enter a value for sine evaluation: 2
Enter number of terms in the summation: 10
Result is: -1.35772
-----
Process exited after 4.786 seconds with return value 0
Press any key to continue . . .
```

**Q18:** Write a program that prints the following pattern as shown below,

You can make separate programs for all these four patterns. *Extra Credit:* If all the four patterns are printed in a single program.

[Hint: Use nested for loop].

## CODE 01:

```
1 #include<iostream> //header file for input output stream
2 using namespace std;
3 int main() //pre-defined function of C++
4 {
5     //variable declaration
6     char star;
7     //variable initialization
8     star='*';
9     //nested for loop
10    for(int i=1; i<=10; ++i)
11    {
12        for(int j=1;j<=i; ++j)
13        {
14            cout<<star;
15        }
16        cout<<endl;
17    }
18    return 0;
19 }
```

## OUTPUT:

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.04541 seconds with return value 0  
Press any key to continue . . .
```

## CODE 02:

```
1 #include<iostream> //header file for input output stream
2 using namespace std;
3 int main() //pre-defined function of C++
4 {
5     //variable declaration
6     char star;
7     //variable initialization
8     star='*';
9     //nested for loop
10    for(int i=10; i>=1; --i)
11    {
12        for(int j=1; j<=i; ++j)
13        {
14            cout<<star;
15        }
16        cout<<endl;
17    }
18    return 0;
19 }
```

## OUTPUT:

## CODE 03:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5
6     for(int i=10; i>=1; --i)
7     {
8         for(int j=1; j<=10-i; ++j)
9         {
10            cout<<" ";
11        }
12        for(int k=1; k<=i; ++k)
13        {
14            cout<<"*";
15        }
16        cout<<endl;
17    }
18    return 0;
19 }
```

## OUTPUT:

## CODE 04:

```
1 #include<iostream> //header file for input output stream
2 using namespace std;
3 int main() //pre-defined function of C++
4 {
5     //variable declaration
6     char star;
7     //variable initialization
8     star='*';
9     //nested for loop
10    for(int i=1; i<=10; ++i) //for first five rows
11    {
12        for(int j=1; j<=10-i; ++j) //spaces for first five rows
13        {
14            cout<<" ";
15        }
16        for(int k=1; k<=i; ++k) //for columns(five rows)
17        {
18            cout<<star;
19        }
20        cout<<endl;
21    }
22
23    return 0;
24 }
25
```

## OUTPUT:

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
-----  
Process exited after 0.06712 seconds with return value 0  
Press any key to continue . . .
```

**Q19:** Write a program that prints the following diamond shape.

[Hint: Use nested for loop].

```
*  
***  
*****  
*****  
*****  
****  
***  
*
```

## CODE:

```
1 #include<iostream> //header file for input output stream  
2 using namespace std;  
3 int main() //pre-defined function of C++  
4 {  
5     //variable declaration  
6     char star;  
7  
8     //variable initialization  
9     star='*';  
10  
11    //nested for loop  
12    for(int i=1; i<=5; ++i) //for first five rows  
13    {  
14        for(int j=1; j<=9-i; ++j) //spaces for first five rows  
15        {  
16            cout<<" ";  
17        }  
18        for(int k=1; k<=2*i-1; ++k) //for columns(five rows)  
19        {  
20            cout<<star;  
21        }  
22        cout<<endl;  
23    }  
24  
25  
26    //nested for loop  
27    for(int i=4; i>=1; --i) //last four rows  
28    {  
29        for(int j=1; j<=9-i; ++j) //spaces for last four rows  
30        {  
31            cout<<" ";  
32        }  
33        for(int k=1; k<=2*i-1; ++k) //for columns  
34        {  
35            cout<<star;  
36        }  
37        cout<<endl;  
38    }  
39    return 0;  
40 }
```

## OUTPUT:

```
*  
***  
*****  
*****  
*****  
*****  
***  
*  
  
-----  
Process exited after 0.08134 seconds with return value 0  
Press any key to continue . . .
```

**Q20:** Modify the program in Q19 to read an odd number in the range of 1 to 19 to specify the number of rows in the diamond, then display a diamond of appropriate size.

## CODE:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     char star;
6     int totalrows;
7     int halfrows;
8     cout<<"Enter total no of rows of diamond in odd number of range(1-19): ";
9     cin>>totalrows;
10
11    star='*';
12    halfrows=(totalrows/2);
13
14    for(int i=1; i<=halfrows; i++)
15    {
16        for(int j=1; j<=totalrows-i; ++j)
17        {
18            cout<<" ";
19        }
20        for(int k=1; k<=2*i-1; ++k)
21        {
22            cout<<star;
23        }
24        cout<<endl;
25    }
26
27    for(int i=totalrows-halfrows; i>=1; --i)
28    {
29        for(int j=1; j<=totalrows-i; ++j)
30        {
31            cout<<" ";
32        }
33        for(int k=1; k<=2*i-1; ++k)
34        {
35            cout<<star;
36        }
37        cout<<endl;
38    }
39    return 0;
40 }
```

## OUTPUT:

```
Process exited after 2.323 seconds with return value 0
Press any key to continue . . .
```

**Q21:** A right triangle can have sides that are all integers. A set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find all Pythagorean triples for side1, side2 and hypotenuse all no larger than 500. Use a triple-nested for loop that tries all possibilities.

## CODE:

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int hyptSquared;
6     int sidesSquared;
7
8     cout<<"\t\tPYTHAGOREAN TRIPLE\n";
9     for(int side1=1; side1<500; side1++)
10    {
11        for(int side2=1; side2<500; side2++)
12        {
13            for(int hypt=1; hypt<500; hypt++)
14            {
15                hyptSquared=hypt*hypt;
16                sidesSquared=side1*side1+side2*side2;
17
18                if(hyptSquared==sidesSquared&&hypt<=500)
19                {
20                    cout<<side1<<"\t"<<side2<<"\t"<<hypt<<endl;
21                }
22            }
23        }
24    }
25
26    return 0;
27 }
```

## OUTPUT:

PYTHAGOREAN TRIPLE		
3	4	5
4	3	5
5	12	13
6	8	10
7	24	25
8	6	10
8	15	17
9	12	15
9	40	41
10	24	26
11	60	61
12	5	13
12	9	15
12	16	20
12	35	37
13	84	85
14	48	50
15	8	17
15	20	25
15	36	39
15	112	113
16	12	20
16	30	34
16	63	65
17	144	145
18	24	30
18	80	82
19	180	181
20	15	25

**Q22:** Write a program that asks user to enter a sequence of 10 elements and then use insertion sort algorithm to sort the sequence in ascending order. Create a function named `insertion_sort()` that takes unsorted sequence as an input and return a pointer that points to the sorted sequence.

## OPTIONAL

**Q23:** Write a program that asks user to enter a sequence of 10 elements and then use selection sort algorithm to sort the sequence in ascending order. Create a function named `selection_sort()` that takes unsorted sequence as an input and return a pointer that points to the sorted sequence.

## OPTIONAL

**Q24:** Write a program that asks user to enter a sequence of 10 elements and then use bubble sort algorithm to sort the sequence in ascending order. Create a function named `bubble_sort()` that takes unsorted sequence as an input and return a pointer that points to the sorted sequence.

## CODE:

```
1 #include<iostream> //enables program to show data to screen
2 using namespace std;
3
4 // making bubble sort function
5 void bubblesort(double array[],int size)
6 {
7     cout<<"\t UNSORTED SEQUENCE\n";
8
9     //taking input from user in unsorted sequence
10    for(int index=0;index<size;index++)
11    {
12        cout<<"\nenter element no:"<<index<<"=";
13        cin>>array[index];
14    }
15
16    //swapp numbers
17    //sorting numbers in ascending order
18    for(int a=0; a<size; a++)
19    {
20        for(int b=0; b<(size-a-1); b++)
21        {
22            if(array[b]>array[b+1])
23            {
24                double temp=array[b];
25                array[b]=array[b+1];
26                array[b+1]=temp;
27            }
28        }
29
30        cout<<"\n\tSORTED SEQUENCE\n";
31
32        //displaying content of sorted array
33        for(int a=0; a<size; a++)
34        {
35            cout<<"\nelement no:"<<a<<"="<<array[a]<<endl;
36        }

```

```
37 L }
38
39 #define elements 10
40
41
42 int main()//function main begins program execution
43 {
44 //declaration of array
45 double my_array[elements];
46 //recalling bubblesort function
47 bubblesort(my_array,elements);
48 return 0;
49 }
```

## OUTPUT:

```
UNSORTED SEQUENCE
enter element no:0=8873
enter element no:1=858
enter element no:2=585
enter element no:3=586
enter element no:4=
252
enter element no:5=203
enter element no:6=15635
enter element no:7=5156
enter element no:8=1456
enter element no:9=213
SORTED SEQUENCE
element no:0=203
element no:1=213
element no:2=252
```

```
element no:3=585  
element no:4=586  
element no:5=858  
element no:6=1456  
element no:7=5156  
element no:8=8873  
element no:9=15635
```

**Q25:** Write a program that inputs an integer, separates the integer into its digit and prints them separated by three spaces each. [Hint: Use vectors to store the separated digits]

## CODE:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int main()
5 {                                     //INTEGER SEPARATOR FOR FOUR DIGIT INTEGER
6     int a,num1,num2,num3,num4;
7     cout<<"Enter four digit integer ";
8     cin>>a;
9
10    num1=a/1000;
11    a=a%1000;
12    num2=a/100;
13    a=a%100;
14    num3=a/10;
15    a=a%10;
16    num4=a;
17    vector<int> b;
18    b.push_back(num1);
19    b.push_back(num2);
20    b.push_back(num3);
21    b.push_back(num4);
22    for (int i = 0; i < b.size(); i++)
23    {
24        cout<<b[i]<<"   ";
25    }
26    return 0;
27 }
```

## OUTPUT:

```
Enter four digit integer 4567
4 5 6 7
-----
Process exited after 8.188 seconds with return value 0
Press any key to continue . . .
```

# CODE:

```
1 #include<iostream>
2 using namespace std;
3 void insert_element(int arr[],int size)
4 {
5     for(int i=0;i<size; i++)
6     {
7         cout<<"Element of index no" << i << endl;
8         cin>>arr[i];
9     }
10    cout<<endl;
11 }
12 void index_element_replace(int arr[],int size)
13 {
14     cout<<endl;
15     int index;
16     cout<<"Enter the index no which you want to replace" << endl;
17     cin>>index;
18     cout<<"Enter a new value for index" << endl;
19     cin>>arr[index];
20 }
21 void index_find_replace(int arr[],int size)
22 {
23     cout<<endl;
24     int flag;
25     int element;
26     cout<<"enter element to search and change its value: ";
27     cin>>element;
28     cout<<endl;
29     for(int i=0; i<size; i++)
30     {
31         if(arr[i]==element)
32         {
33             cout<<"number " <<element<<" found at index " <<i;
34             cout<<endl;
35             cout<<"\nnow enter new value for this index: ";
36             cin>>arr[i];
37         }
38     }
39     cout<<endl;
40 }
41
42 void index_element_remove(int arr[],int size)
43 {
44     int index;
45     cout<<"enter index which you want to delete: ";
46     cin>>index;
47     for(int i=index; i<size; i++)
48     {
49         int temporary=arr[i];
50         arr[i]=arr[i+1];
51         arr[i+1]=temporary;
52     }
53 }
54 }
```

```
55
56
57 void find_element(int arr[],int size)
58 {
59     int t;
60     int element;
61     cout<<"\nEnter element to search: ";
62     cin>>element;
63     cout<<endl;
64     for(int i=0; i<size; i++)
65     {
66         if(arr[i]==element)
67         {t==1;
68             cout<<"number "<<element<<"found at index "<<i;
69             cout<<endl;
70             break;
71         }
72     }
73 }
74
75 void display(int arr[],int size)
76 {
77     cout<<"\t\tCONTENTS OF ARRAY";
78     for(int i=0; i<size; i++)
79     {
80         cout<<"\nelement no."<<i<< "="<<arr[i]<<endl;
81     }
82 #define elements 10
83
84
85 int main()
86 {
87     int array[elements];
88
89     insert_element(array,elements);
90     display(array,elements);
91     index_find_replace(array,elements);
92     display(array,elements);
93     index_element_replace(array,elements);
94     display(array,elements);
95     find_element(array,elements);
96     index_element_remove(array,elements);
97     display(array,elements-1);
98
99
100    return 0;
101 }
```

# OUTPUT:

```
Element of index no 0  
2  
Element of index no 1  
4  
Element of index no 2  
6  
Element of index no 3  
8  
Element of index no 4  
10  
Element of index no 5  
12  
Element of index no 6  
14  
Element of index no 7  
16  
Element of index no 8  
18  
Element of index no 9  
20  
  
        CONTENTS OF ARRAY  
element no.0=2  
  
element no.1=4  
  
element no.2=6  
  
element no.3=8
```

```
element no.4=10  
element no.5=30  
element no.6=14  
element no.7=16  
element no.8=18  
element no.9=30  
  
enter element to search: 18  
  
number 18found at index 8  
enter index which you want to delete: 4  
        CONTENTS OF ARRAY  
element no.0=2  
  
element no.1=4  
  
element no.2=6  
  
element no.3=8  
element no.4=30  
element no.5=14
```

```
element no.6=16
```

```
element no.7=18
```

```
element no.8=30
```

```
-----  
Process exited after 145.9 seconds with return value 0
```

```
Press any key to continue . . . ■
```