**UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE**

**(NAROWAL CAMPUS)**

**­­­­**

**NAME :**

**ABDULAHAD HUSSAIN**

**ROLL NO. :**

**549**

**SECTION:**

**A**

**SUBJECT:**

**PROGRAMING FUNDAMENTAL**

**LAB MANUAL**

**LAB #01:**

**Objective:**

C++ is most popularly used for building large software infrastructure and applications that run on limited resources, for example gaming programing, data structure. The main purpose of C++ programing was to add object oriented to the C programing.

**Data Type:**

* Integer (int)
* Floating (float)
* Character (char)
* String (str or text)
* Boolean (bool)
* Array

**Variable:**

Variables in C++ is **a name given to a memory location**. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location. All the operations done on the variable effects that memory location

**Operators:**

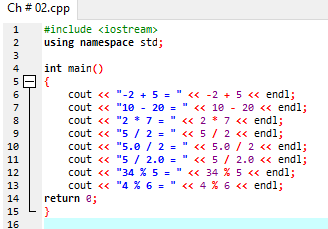
C++ arithmetic operators:

* Addition (+)
* Subtraction (-)
* Multiplication (\*)
* Division (/)
* Modulus operator (%)

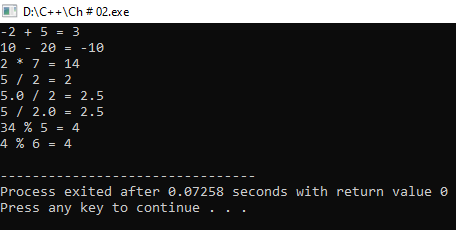
**Precedence of operator:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C operation | | | | Operation | | | | | Order of operation | | | | |
|  | () |  | |  | Paranthesis | |  | |  | The expression in parenthesis is | | |  |
|  |  | |
| evaluated first. |  | |
| In case of nested paranthesis the part of expression in the inner most parenthesis is | | |
| evaluated first. |  | |
| Unary + and - | | | | Addition & subtraction | | | | | From left to right | | | | |
|  | \*,%,/ | |  |  | Multiplication, Division, | | |  |  | From left to right | |  | |
|  | | Modulus |  | |  | |
| Binary + or - | | | | Addition  Subtraction | | | | | From left to right | | | | |

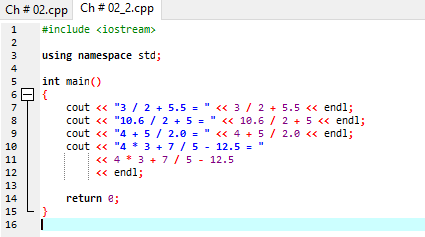
**Program No.1:**



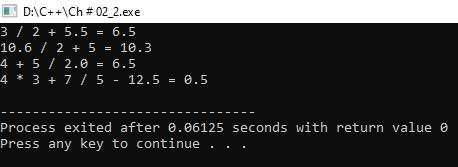
**Output:**

****

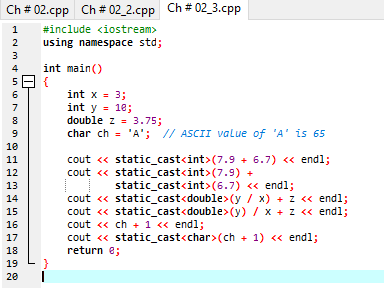
**Program No.2:**

****

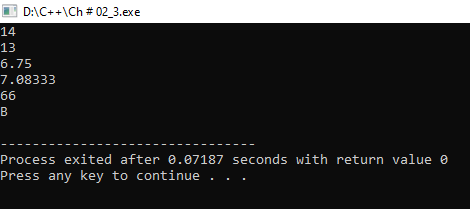
**Output:**

****

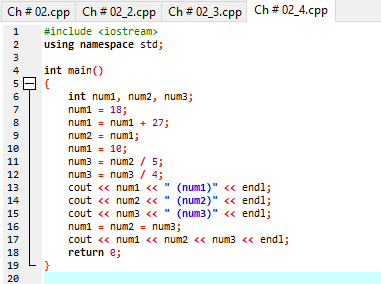
**Program No.3:**

****

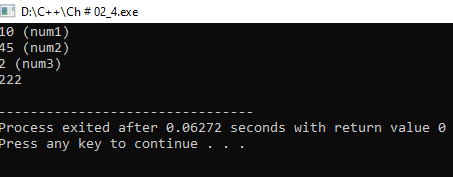
**Output:**

****

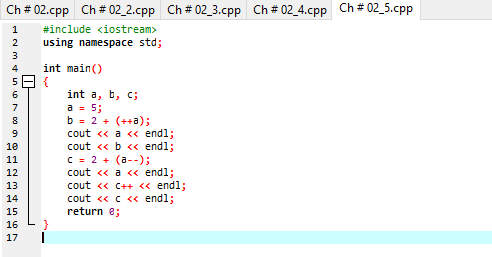
**Program No.4:**

****

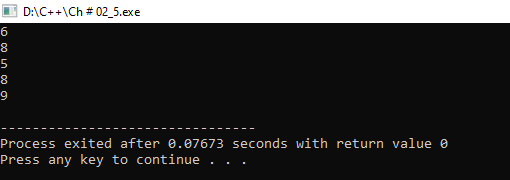
**Output:**

****

**Program No.5:**

****

**Output:**

****

**LAB #02:**

**Objective:**

C++ is most popularly used for building large software infrastructure and applications that run on limited resources, for example gaming programing, data structure. The main purpose of C++ programing was to add object oriented to the C programing.

**Data Type:**

* Integer (int)
* Floating (float)
* Character (char)
* String (str or text)
* Boolean (bool)
* Array

**Variable:**

Variables in C++ is **a name given to a memory location**. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location. All the operations done on the variable effects that memory location

**Operators:**

**C++ arithmetic operators**:

* Addition (+)
* Subtraction (-)
* Multiplication (\*)
* Division (/)
* Modulus operator (%)

**Precedence of operator:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C operation | | | | Operation | | | | | Order of operation | | | | |
|  | () |  | |  | Paranthesis | |  | |  | The expression in parenthesis is | | |  |
|  |  | |
| evaluated first. |  | |
| In case of nested paranthesis the part of expression in the inner most parenthesis is | | |
| evaluated first. |  | |
| Unary + and - | | | | Addition & subtraction | | | | | From left to right | | | | |
|  | \*,%,/ | |  |  | Multiplication, Division, | | |  |  | From left to right | |  | |
|  | | Modulus |  | |  | |
| Binary + or - | | | | Addition  Subtraction | | | | | From left to right | | | | |

**Program No. 1:**

**Statement:**

Get two integer from user as input and calculate and print their sum, difference, quotient and remainder.

**Alogarithm:**   **START**

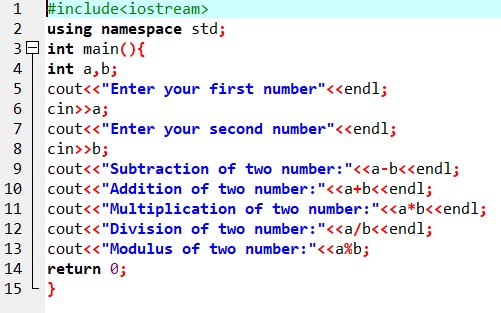
Step 1: Declare two variable a and b.

Step 2: Assign their data type that is integer.

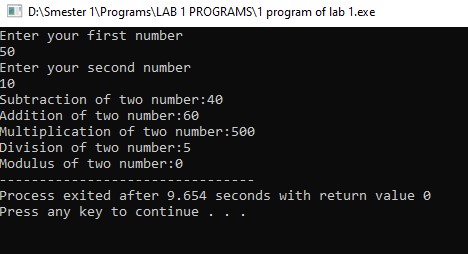
Step 3: Take value from user by using **“cin”** command. Step 4: Now apply arithmetic operation on a and b that is (+,-,\*,/,%).

Step 5: Print their result by using **“cout”** command**.**

**END**

**Programe:** 

**Output:**



**Program No. 2:**

**Statement:**

Ali’s basic salary is entered through the keyboard. If his house rent allowance is 40 percent of basic salary and transport allowance is 20 percent of basic salary. Then write a program that determines his gross salary.

**Alogarithm:**

**START**

Step 1: Declare four variable basic, house, transport and gross.

Step 2: Take basic salary from user and store in basic.

Step 3: Calculate 40% house rent using this formula (house = basic \* 0.4).

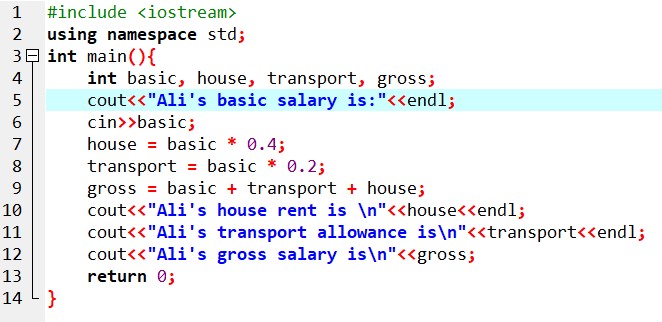
Step 4: Calculate 20% transport allowance by using this formula (transport = basic \* 0.2).

Step 5: Calculate gross salary by using this formula (gross = basic + house + transport).

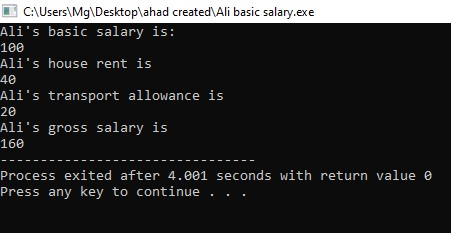
Step 6: Print value on screen.

**END**

**Program:**



**Output:**



**Program No 3:**

**Statement:**

Get an integer as input from

keyboard. Determine it is whether even or oddin nature.

**Alogarithm:**

**START**

Step 1: Declare three variables a, b and c.

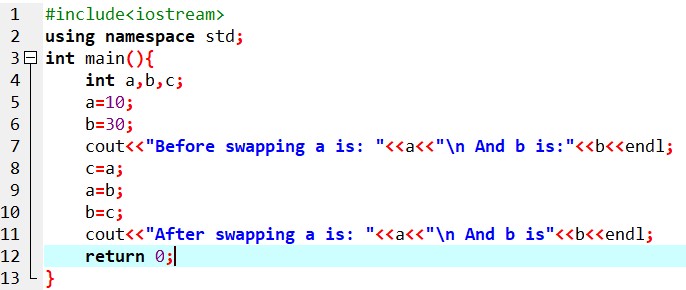
Step 2: Take two values from user and store them in a and b and third variable c is empty.

Step 3: Using operation c=a, a=b and b=c.

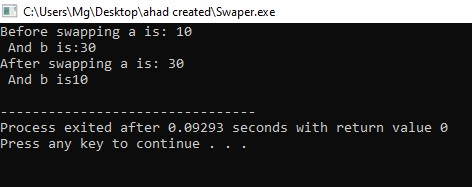
Step 4: Print the value of a and b.

**END**

**Program:**



**Output:**



**LAB #03:**

**I/O Library Header Files:**

In C++, I/O is a sequence of bytes, called a stream, from source to destination. Bytes are usually characters, unless program requires other types of information (images, sound)

Input stream: sequence of characters from an input device to the computer.

Output stream: sequence of characters from the computer to an output device

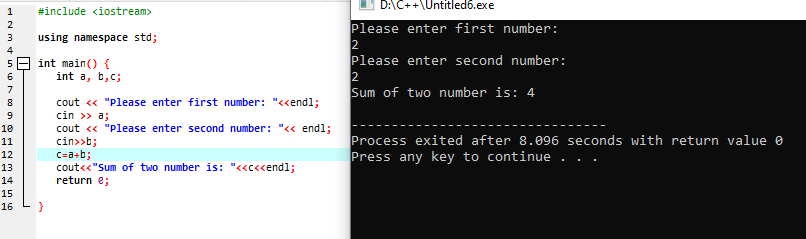
**Input And Output Function:**

The C++ standard libraries provide an extensive set of input/output capabilities which we will see in subsequent chapters. This chapter will discuss very basic and most common I/O operations required for C++ programming.

C++ I/O occurs in streams, which are sequences of bytes. If bytes flow from a device like a keyboard, a disk drive, or a network connection etc. to main memory, this is called **input operation** and if bytes flow from main memory to a device like a display screen, a printer, a disk drive, or a network connection, etc., this is called **output operation**.

**Cin:**

The predefined object **cin** is an instance of **iostream** class. The cin object is said to be attached to the standard input device, which usually is the keyboard. The **cin** is used in conjunction with the stream extraction operator, which is written as >> which are two greater than signs as shown in the following example.

****

**Cin.get():**

The cin.get () C++ function is used to access the character array. This is a very basic function in the c++ programming language that is used to get input from the user. The cin.get() C++ function also includes the white space characters into it.

**cin.ignore():**

The cin.ignore() function is used to ignore or clear one or more characters from the input buffer. 

**cin.peek():**

**cin.peek()** returns the next input character without taking it out of the input stream.

**cin.putback():**

**cin.putback()**, as the name suggests, puts a character back to the beginning of the input stream. One thing noteworthy is that you must read a character before you use this function.

**Manipulator:**

Manipulator are instructions to the output stream that modify the output in various.

**Manipulators** are helping functions that can modify the input/output stream. It does not mean that we change the value of a variable, it only modifies the I/O stream using insertion (<<) and extraction (>>) operators.

**set precision():**

set precision in C++ is a manipulator used to set the precision of the floating-point number after the decimal in the output stream. Set precision() function is defined inside the header file. set precision in C++ works to prevent the loss of information.

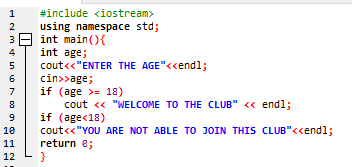
**LAB #04:**

**Conditional Statement:**

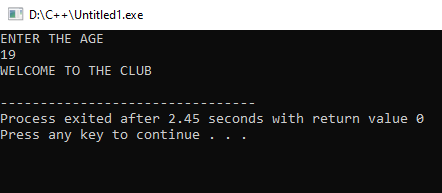
**IF Statement:**

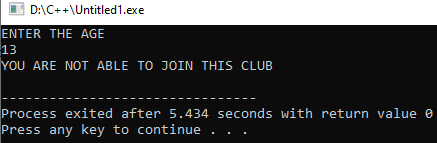
The if statement is the simplest of the decision statement.

**Program No. 1:**

****

**Output:**

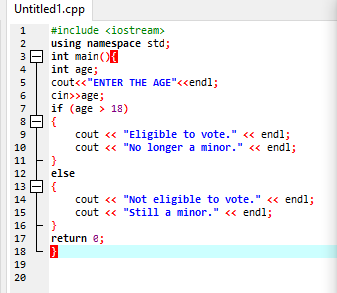
****

****

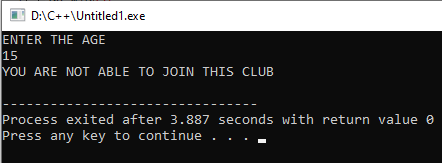
**If….else staement:**

The if statement lets you do something if a condition is true. If it is not true, nothings happens. But suppose we want to do one thing if a condition is true, and do something else if it’s false. That’s where the if…else statement comes in. It consists of an if statement, followed by a statement or block of statements, followed by the keyword else, followed by another statement or block of elements.

**Program :**

****

**Output :**

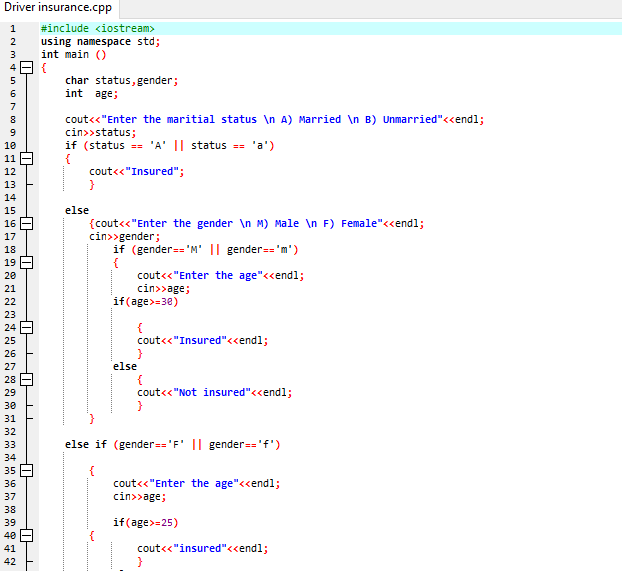
****

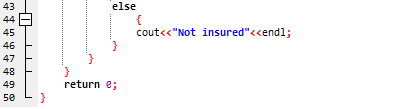
**If….else if statement:**

The nested if...else statements in the ADIFELSE program look clumsy and can be hard—for humans—to interpret, especially if they are nested more deeply than shown. However, there’s

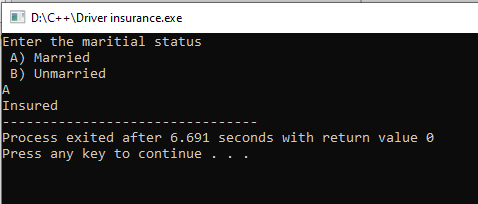
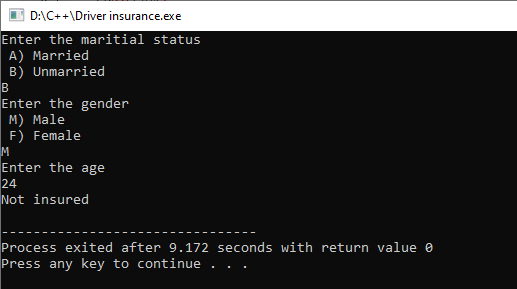
another approach to writing the same statements.

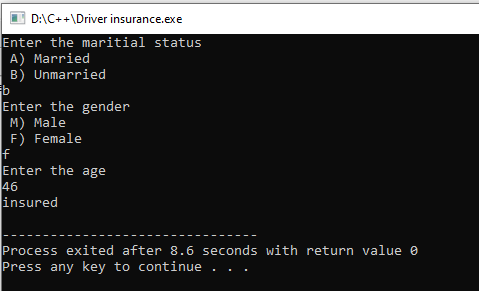
**Program:**

****

****

**Output:**

****

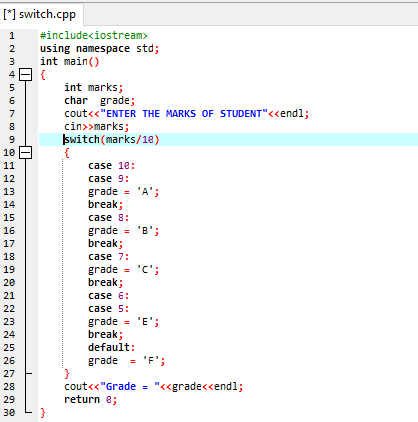
****

**Switch Statement:**

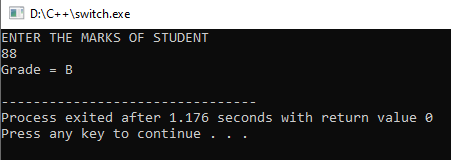
If you have a large decision tree, and all the decisions depend on the value of the same variable,

you will probably want to consider a switch statement instead of a ladder of if...else or else if constructions.

**Program:**

****

**Output:**

****

**LAB #05:**

**Program No. 1:**

Write a program to convert temperature from fahrenheit to celsius degree.

**Algorithm:**

1. Declare two variables of double datatype

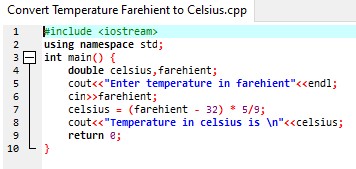
2. Ask the value from user and prompt message that user should enter the temperature in fahrenheit.

3. Convert the Fahrenheit into Celsius using following formula:

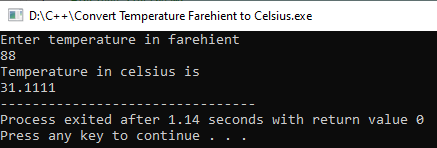
Celsius = (fahrenhiet - 32) \* 5 / 2

4. Display the given value with fahrehient unit and converted value in celsius

**Source Code:**



**Output:**



**Program No. 2:**

A company insures its driver in following three cases:

1. If driver is married
2. If driver is unmarried and male and his age >30
3. If driver is unmarried and female and her age >25

In all other case the driver is not insured. Write a program that take the marital status, gender and age of driver as input from keyboard and determines whether he is insured or not.

**Algorithm:**

1. Declare three variables two of character datatype and one of integer datatype

2. Prompt a message and ask his/her marital status.

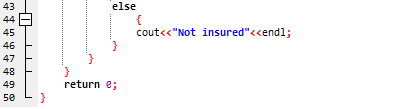
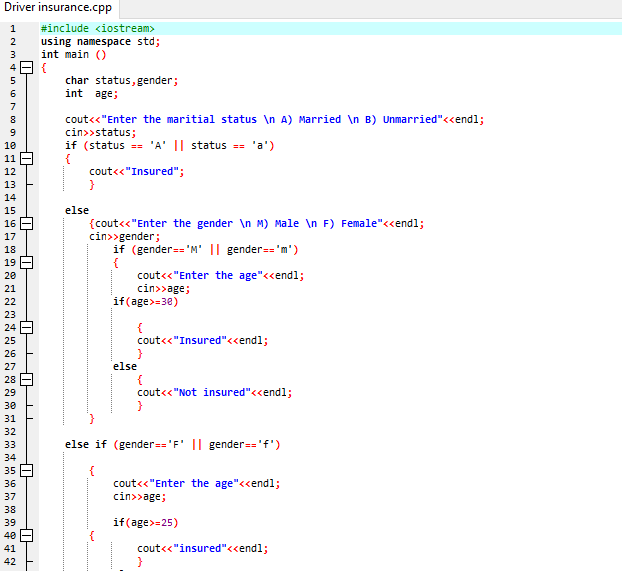
3. If user is married then display a message that he/she is insured.

4. If user is not married than ask gender by displaying a message. After that prompt a message and ask users age.

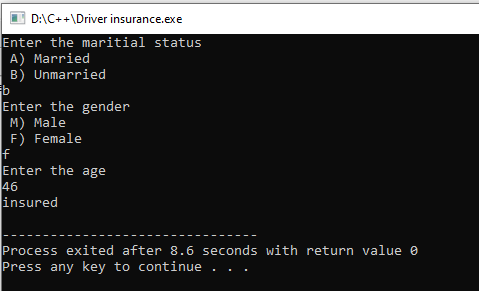
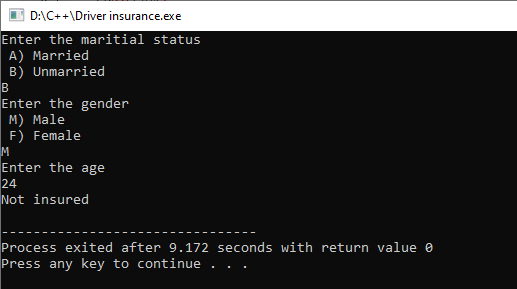
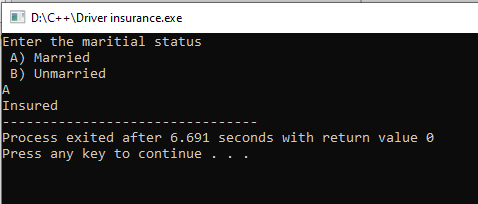
5. If user is female and her age is greater than 25 than display a message that she is insured otherwise show a message which tell her that she is not insured.

6. If user is male and his age is greater than 30 than display a message that he is insured otherwise show a message which tell her that he is not insured.

**Source Code:**



**Output:**

****

**LAB #06:**

**Program No. 1:**

In a tabular format, print the integers from 0 to 9, their squares and their cubes.

Num square cubes

0 0 0

1 1 1

2 4 8

**Algorithm:**

The algorithm to solve a above problem is as follows:

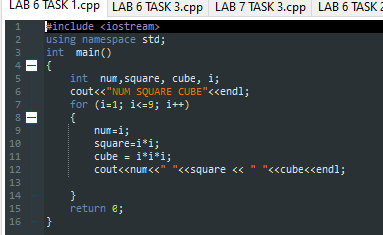
I. First run a loop from 0 to 9.

II. Multiple a iteration variable with itself and store a result into a third variable called square.

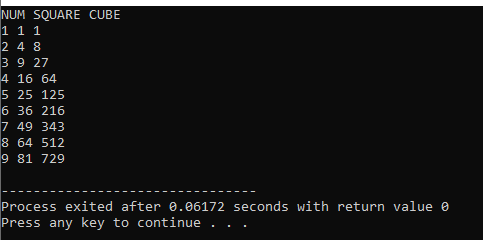
III. Multiple a iteration variable with third variable which is square and store result into a variable called cube.

IV. For formatting we use setw() function.

**Source code:**



**OUTPUT:**



**Flow Chart:**

**END**

Num, Square, Cube

Num = i;

Square=i \* i;

Cube = i \* i \* i;­

Condition

for (i=0; i<=9; i++)

NUM, SQUARE, CUBE

**START**

**FALSE**

**TURE**

**Program No. 2:**

Write a program that take grades secured by the students of a class as input and print description of the grade of a student. If the student has grade ‘A’ we print ‘Excellent’ and 'Very good', 'good', 'poor' and 'fail' for grades B, C, D, and F respectively. Note: use switch statement.

**Algorithm:**

The algorithm of above problem is as follows:

I. Prompt a message and get grades from user

II. Make condition cases in switch statement

‘A’ Excellent

‘B’ Very Good

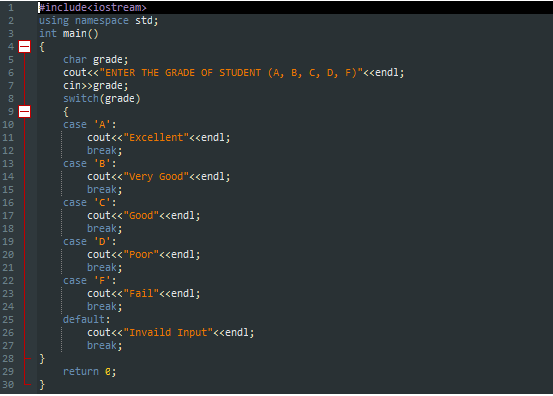
‘C’ Good

‘D’ Poor

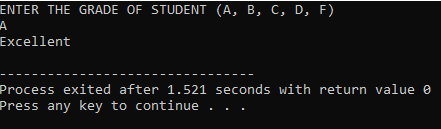
‘F ’ Fail

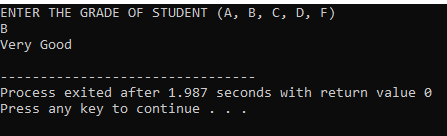
III. After checking condition print out the respective description which matches the condition.

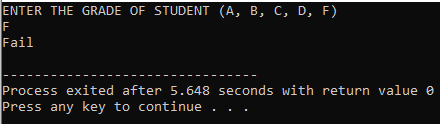
**Source Code:**



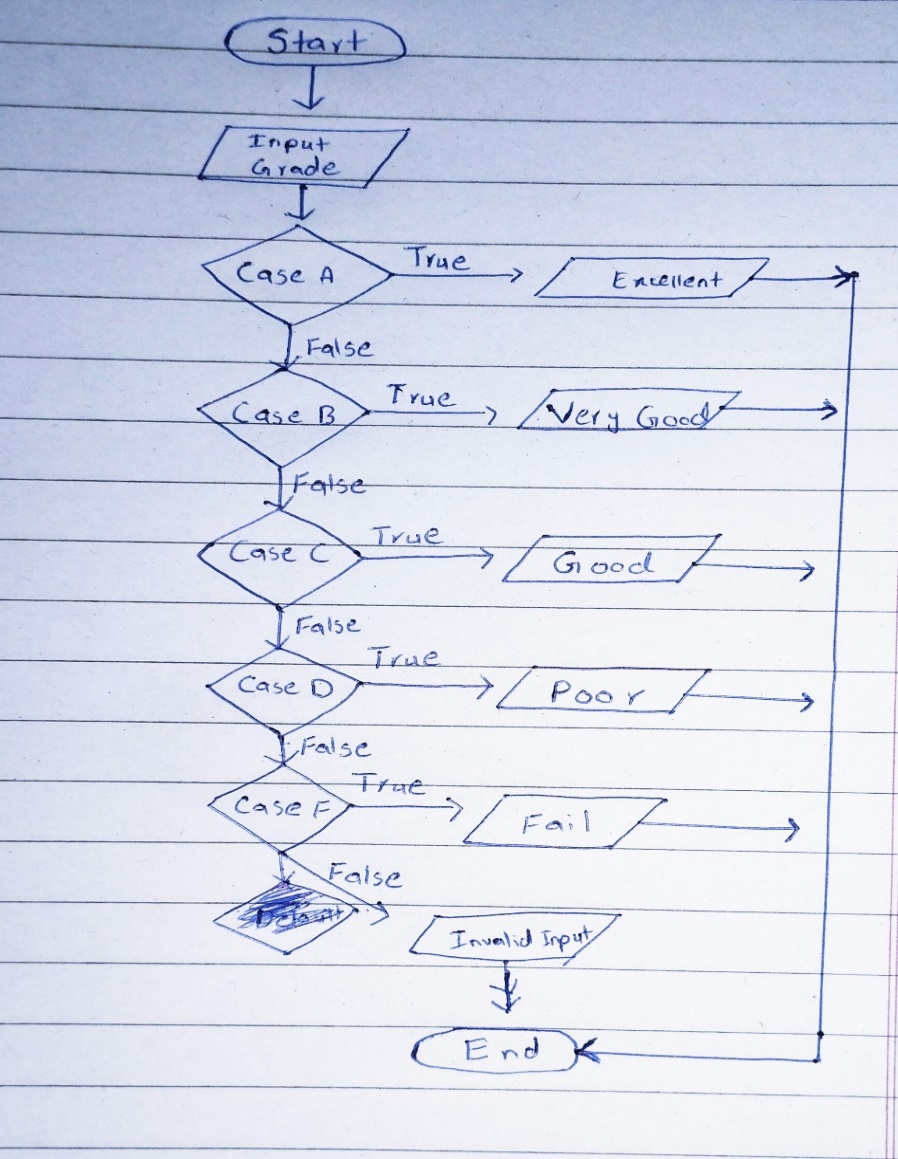
**Output:**

****

****

****

**Flow Chart:**

****

**Program No. 3:**

In a company, there are deductions from the salary of the employees for a fund. The deductions rules are as follows: If salary is less than 10,000 then no deduction If salary is more than 10,000 and less than 20,000 then deduct Rs. 1,000 as fund If salary is equal to or more than 20,000 then deduct 7 % of the salary for fund. Take salary input from user and after appropriate deduction show the net payable amount. Note: use switch statement.

**Algorithm:**

The steps to solve above problem is as follows:

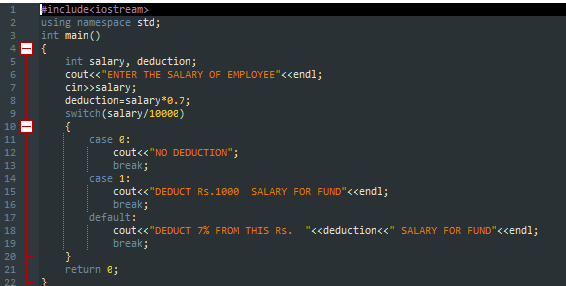
I. Prompt a message a take salary from user

II. Divide the salary with 10000 in switch expression

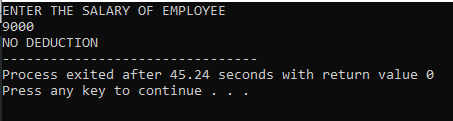
III. Make cases for conditions. For case 0 there will be no deduction. For case 1 the deduction of 1000 is happen and for case 2 and default case the deduction of 7% takes place

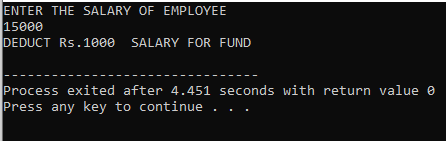
IV. Display a net value after deduction from salary

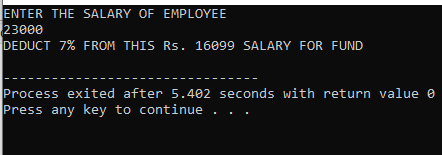
**Source Code:**

****

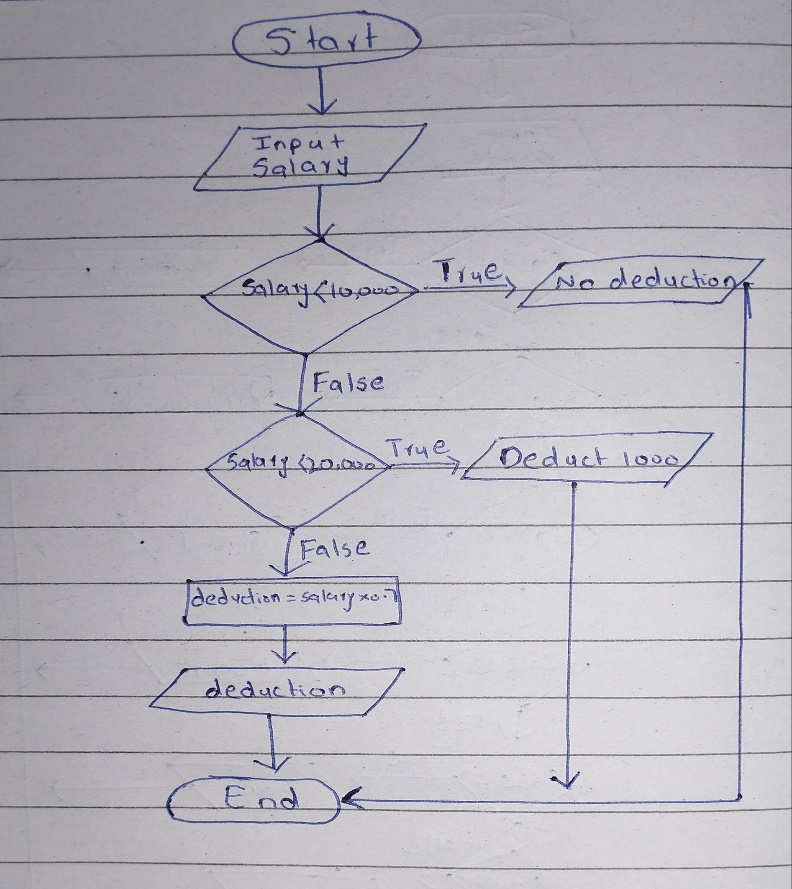
**Output:**

****

****

****

**Flow Chart:**

****

**LAB #07:**

**Program No. 1:**

**Statement:**

#include <iostream>

using namespace std;

int main()

{

int x;

for (x =5; x <= 50; x = x+5)

{

// x = x + 5 could also be written x += 5

cout<< "Loop counter value is " << x << ".\n";

}

return 0;

}

Analyze above code and Write output.

**Output:**

Loop counter value is 5

Loop counter value is 10

Loop counter value is 15

Loop counter value is 20

Loop counter value is 25

Loop counter value is 30

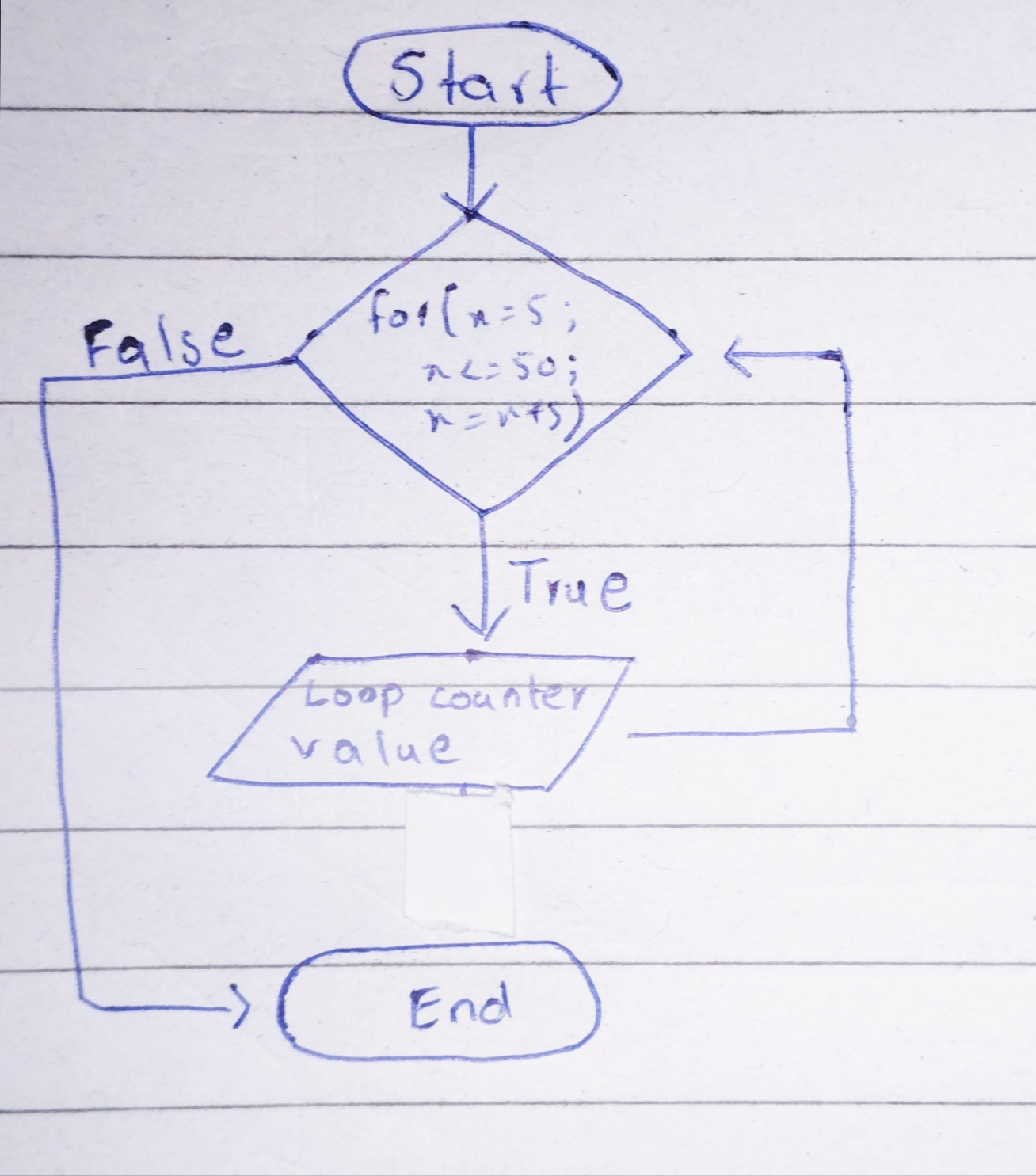
Loop counter value is 35

Loop counter value is 40

Loop counter value is 45

Loop counter value is 50

**Flow Chart:**

****

**Program No. 2:**

**Statement:**

Write a program that finds running sum of first 25 natural number.

**Algorithm:**

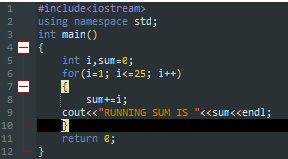
The Algorithm is as follows:

I. Start loop from 1 and it will run until iterator variable value becomes equal to 25. There should be a simple increment in iterator variable.

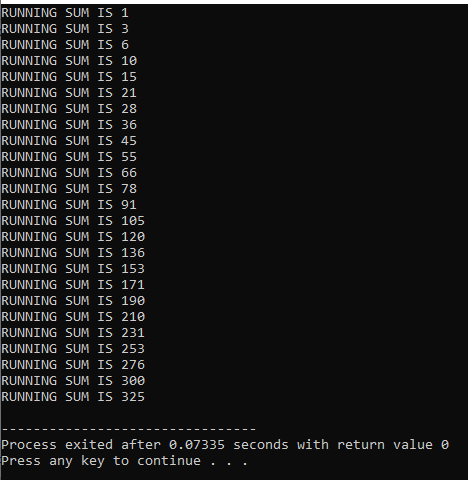
II. Make a global variable name sum which value will be zero at decleration and store the iterator variable value in it when the execution of loop body take place.

III. Display the value of result variable in every execution of loop body.

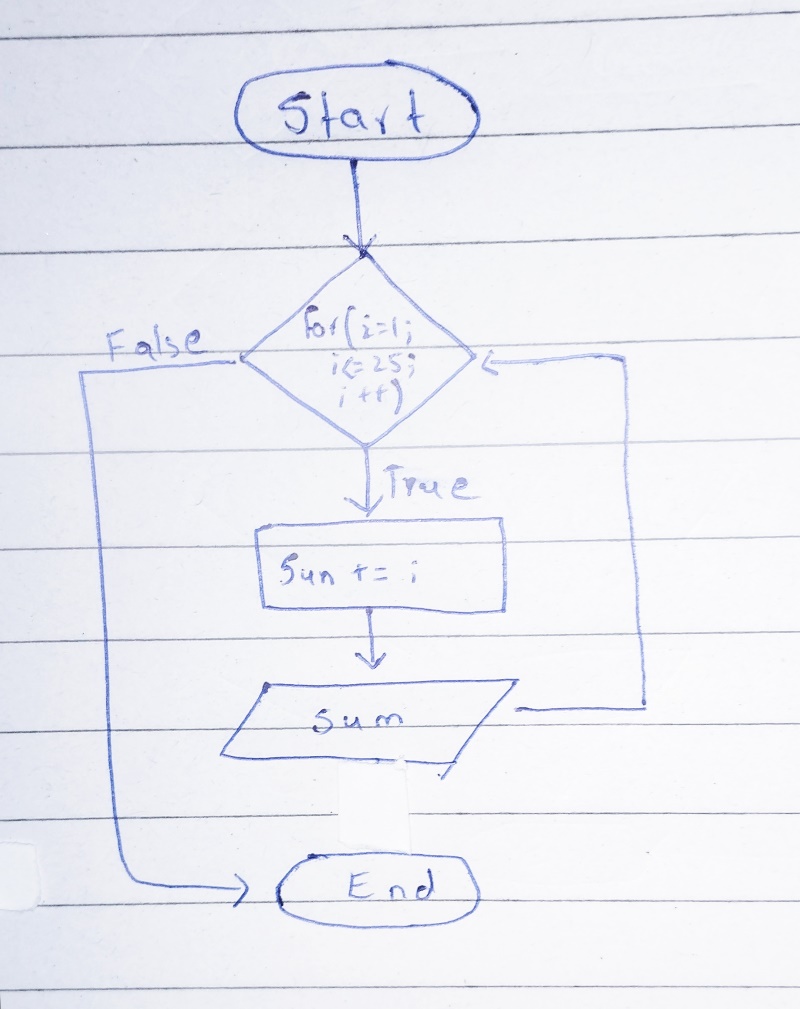
**Source Code:**

****

**Output:**

****

**Flow Chart:**

****

**Program No. 3:**

**Statement:**

Write a program to get an integer from user as input and print its multiplication table.

**Algorithm:**

The algorithm to do above project is as follows:

I. Prompt a message and get a number for its table

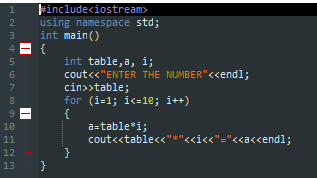
II. Prompt a message and ask the start of table

III. Prompt a message and get the limit and end point of table

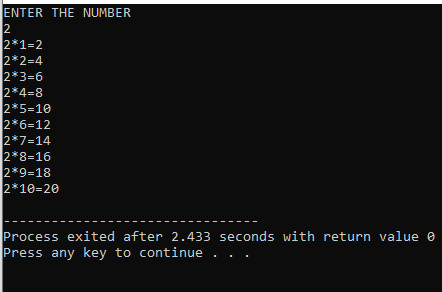
IV. Use for loop, limit as condition and start as initialization and take simple increment of one

V. Print the statement inside the loop body in such way that shows a table pattern

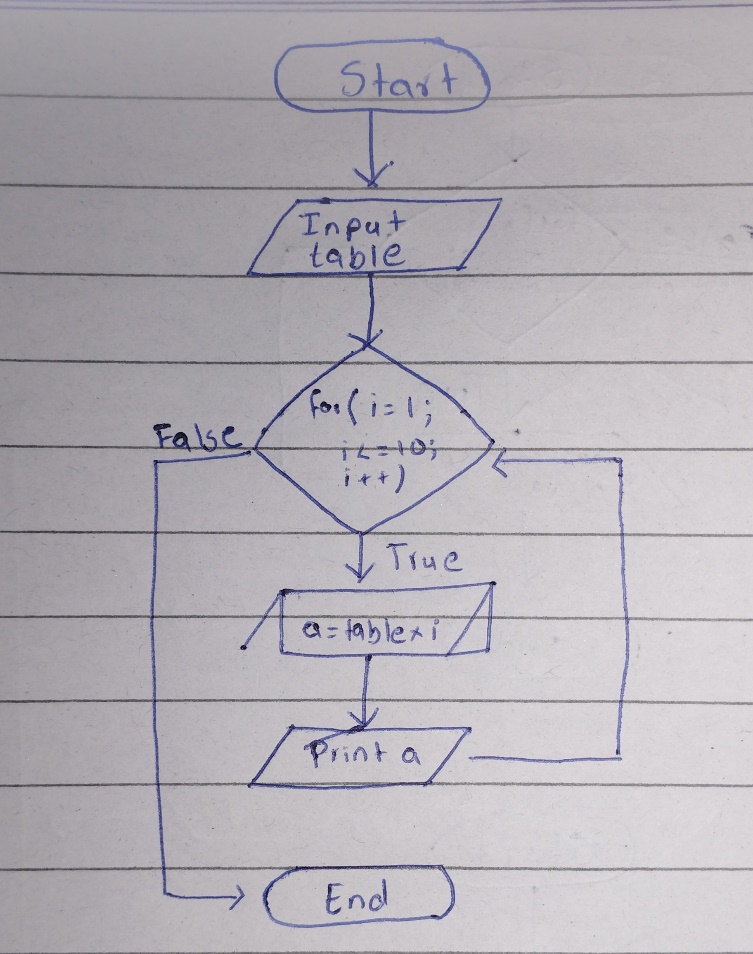
**Source Code:**

****

**Output:**

****

**Flow Chart:**

****

**LAB #08:**

**Task No. 01:**

**Statement:**

Write a program that reads in the size of the side of a square then prints a hollow square of that size out of asterisks and blanks.

**Algorithm:**

The steps to solve the above problem are as follows:

1. Prompt a message and takes several rows and columns in an integer number.

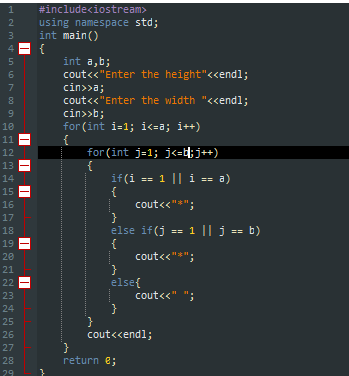
2. To print an asterisk pattern we use the nested loop concept.

3. To print columns we use an outer loop up to a number of columns entered by the user.

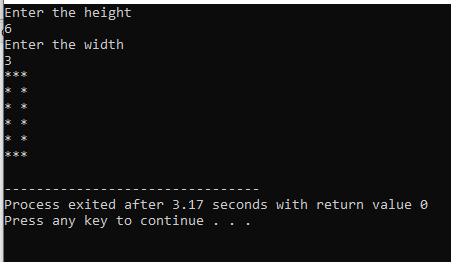
4. To print rows we use an inner loop up to a number of rows which entered by the user.

5. We will print an asterisk for inner and outer loops only when the value of iteration variables becomes equal to 1 or the number of rows or columns.

**Source Code:**

****

**Output:**

****

**Task 02:**

**Statement:**

Write a program to determine whether a number is palindrome /armstrong or not.

**Algorithm:**

The algorithm to solve above problem is given as follows:

1. Prompt a message and take a number from user

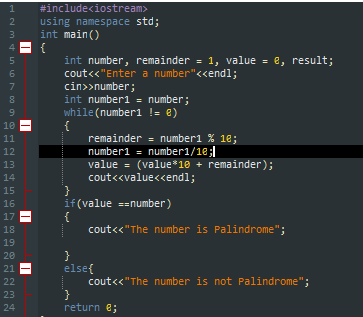
2. Store the value in another variable

3. Run a loop until other variable value becomes not equal to zero and in this loop save the remainder in a variable. The remainder is save in another variable which is known as the reverse number and there will be updation occur in such way that the number is divide by 10. This loop is used to reverse a number.

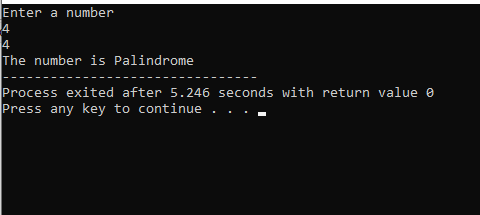
4. Check the reverse number with real number if they are equal than such number will be palindrome number otherwise display a message

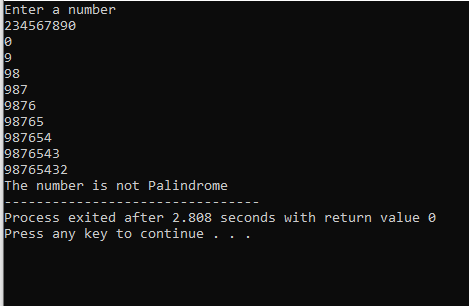
that the number is not palindrome number

**Source Code:**

****

**Output:**

****

****

**Flow Chart:**

****

**LAB #09:**

**Program No. 1:**

**Statement:**

Get division and number of subjects failed of student as input from keyboard and determine using the following chart that how many grace marks are to be awarded to him/her per subject failed.

|  |  |  |
| --- | --- | --- |
| **Division** | **No. Of subjects failed** | **Grace marks** |
| 1st | >3 | No grace marks |
|  | <=3 | 5 marks per subject |
| 2nd | >2 | No grace marks |
|  | <=2 | 4 marks per subject |
| 3rd | >1 | No grace marks |
|  | =1 | 5 grace marks |

**Algorithm:**

The steps to solve the above problem are as follows:

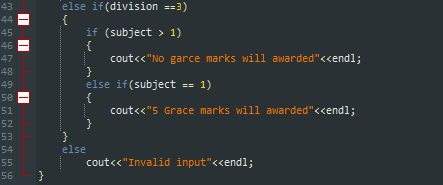
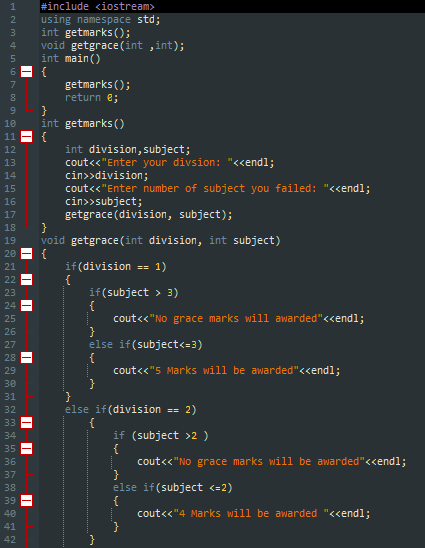
1. Prompt a message and take the division and number of subjects the student failed.

2. Do this by calling a function of gracemarks and call getgrace function in gracemarks function. We give the division and number of subjects as arguments to the second function.

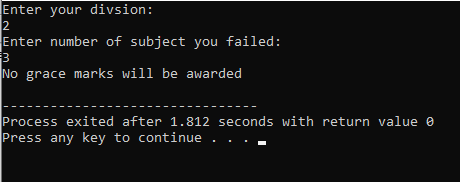
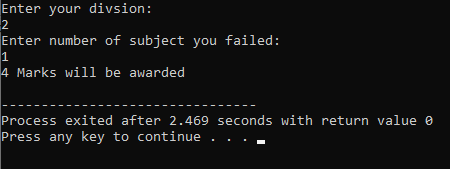
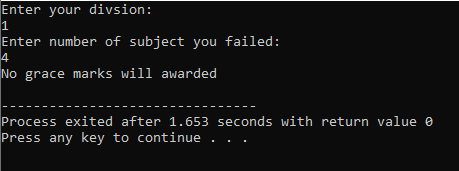
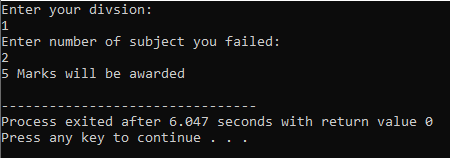
3. The second function determines and prints the statements according to the above table.

4. At the end we print the marks statement to tell the student how many marks he/she will get as grace marks.

**Source Code:**

****

**Output:**

****

**Program No. 2:**

**Statement:**

Get a positive integer from user as input and determine its factorial.

**Algorithm:**

The algorithm of above problem is as follows:

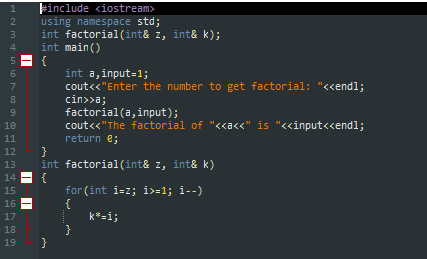
1. Prompt a message and take a number from user for its factorial

2. Give the reference of number variable and result variable to function for calculating factorial

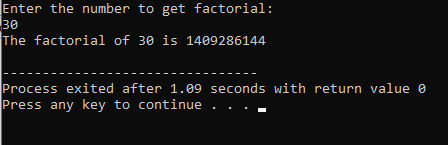
3. Factorial function take the number and use iterations to find the factorial and store the result in the result variable

4. At the end in the main function print the statement which print the number and its factorial

**Source Code:**

****

**Output:**

****

**LAB #10:**

**Program No. 01:**

**Statement:**

Write a program that inputs temperatures of seven

days of a week and calculates its average.

**Algorithm:**

The algorithm to solve above problem is as follows:

1. Make an integer array of 7 size and take values of it from user using iteration method.

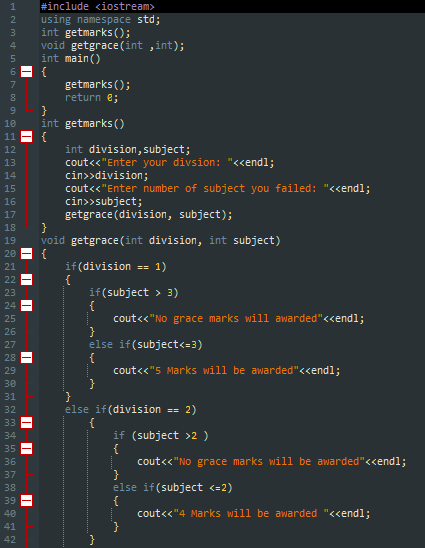
2. Store the input values into a variable which sum up the input value with previous input

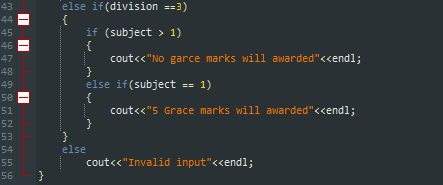
value. This can also possible using iteration approach

3. Divide the sum on 7 and store the result into another variable.

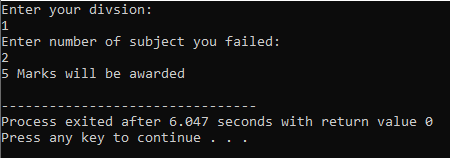
4. At the end print out the statement that average of 7 days is as follows

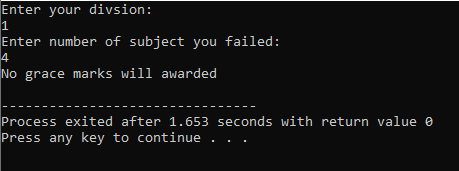
**Source Code:**

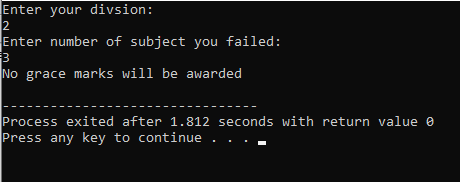
****

****

**OUTPUT:**

****

****

****

**Program No. 2:**

**Statement :**

Write a C++ program to copy in reverse order the elements of an integer 1-D array into another array of same kind.

**Algorithm:**

The algorithm to solve above problem is as follows:

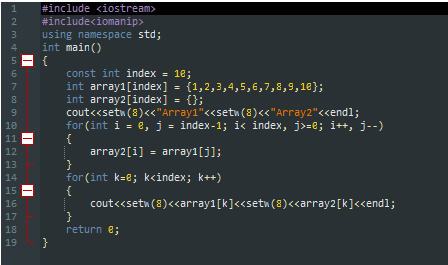
1. Prompt a message and takes the value of first array from user.

2. Store the value one by one into an array.

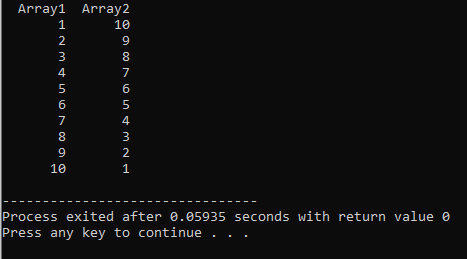
3. Use the loop approach to store the value of entries into another array into reverse order.

4. At the end print the first and second array.

**Source Code:**

****

**Output:**

****

**LAB #11:**

**Task No. 01:**

**Statement :**

Write a program that input marks of 5 subjects of a student and calculate its average and grade in a function and array respectively

1. Calculate average by referance
2. Show grade by referance(A,B,C)

**Algorithm:**

The steps to solve the above problem are as follows:

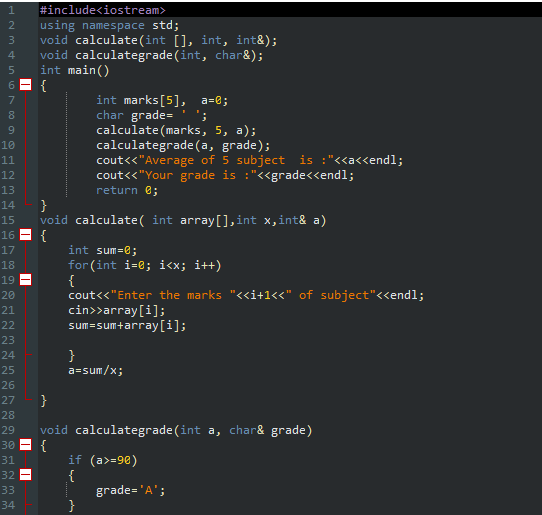
1. Make an array of the length of 5 and prompt a message and take all entries in sequence and store it in an array

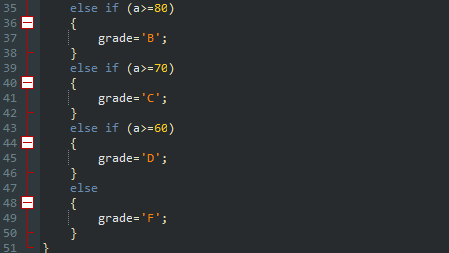
2. Call a function for calculating the average of marks. This function stores the sum of all entries in a variable and divides the sum by five to calculate an average of marks. This functions take the reference of a variable and change its value.

3. Call a function after calculating the average for determining the grade. This function takes the average and a reference variable and changes the value of the variable for a grade.

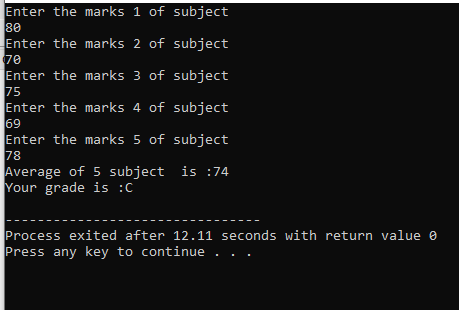
4. At the end print out the determined grade and average of the student

**Source Code:**

****

****

**Output:**

****

**Task No. 02:**

**Statement:**

Write a program that search the smallest element in a 2d array by column.

**Algorithm:**

The algorithm to solve the above problem is as follows:

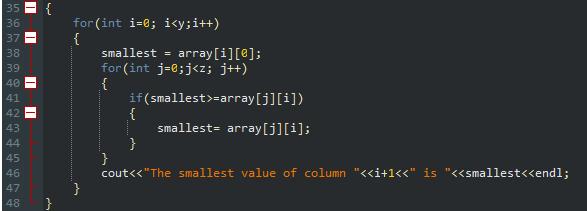
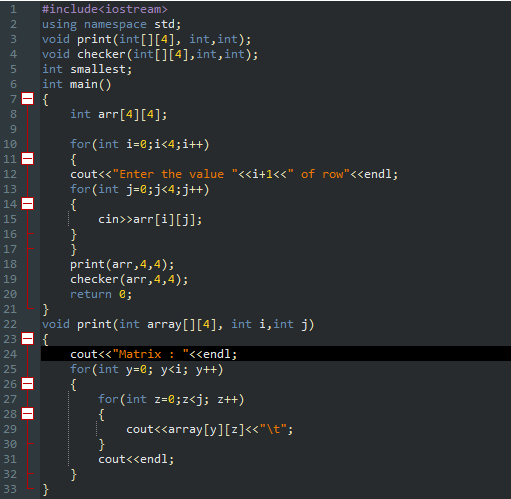
1. Prompt and message and take entries of each row one by one.

Use a loop to take a value and store it in a 3 by 3 integer array. Also, use a loop to print and check the values.

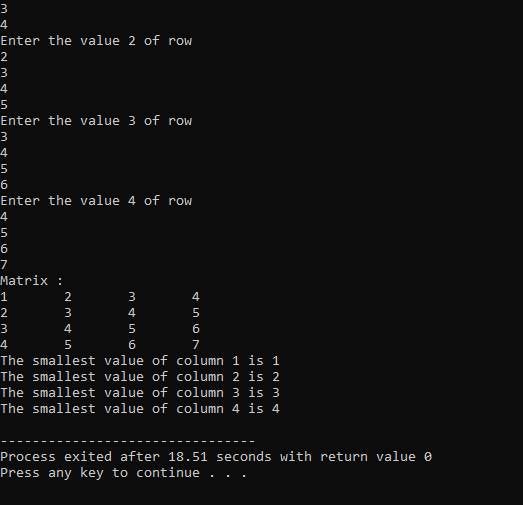
2. Print values of a two-dimensional array in form of a matrix.

3. Use a loop to check values. Start from the first vertical line and check the smallest element in that vertical column. Repeat this process for all vertical columns. And at the end print the smallest value of all vertical columns.

**Source Code:**

****

**Output:**

****

**LAB #12:**

**Program No 01:**

Write a program that will search a name from array of strings using linear search function**.**

**Algorithm:**

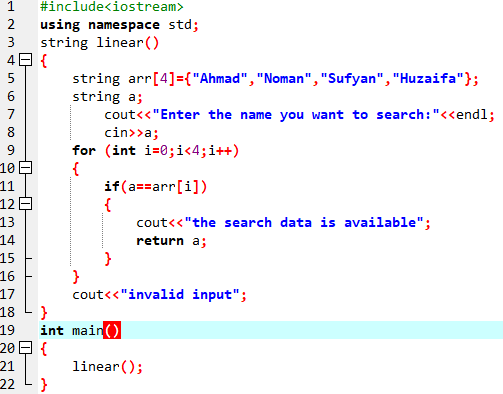
The algorithm is as follows:

1. Prompt a message and ask a name from the user to check whether it is in an array of strings or not.

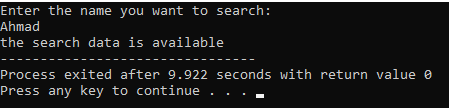
2. Make an array of some size and store some names in it.

3. Make a function name linearSearch which takes the array, user-entered name, and size of the array as parameters. The function uses a loop to check the condition and the condition is as if the value becomes equal so stop the loop there and give some return value which tells it is in our array otherwise return some value after the completion of the loop that indicates that it is not in our array

**Source Code:**



**OUTPUT:**



**Program No 02:**

Write a program to sort given numeric data of an array.

**Algorithm:**

The algorithm is as follows:

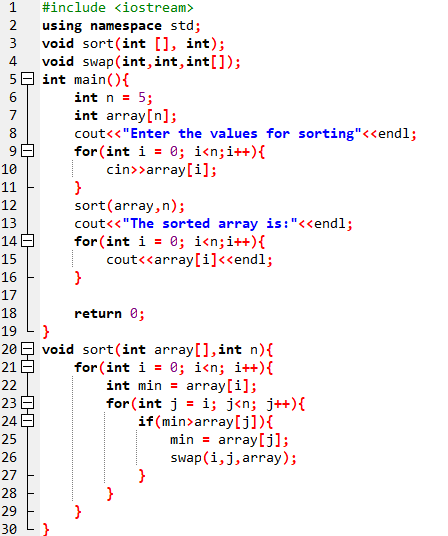
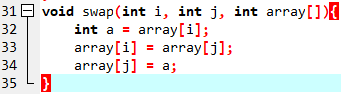
1. Prompt a message and take the value of the array from the user.

2. Run a nested loop for sorting, an outer loop for taking the value of the array, and an inner for checking the value, and in the inner loop, we use the swap function for swapping for doing the sort.

3. Take the first value as a minimum value of the array and check whether it is minimum or not if yes do no sort and jump to the next index otherwise swap it with the next one and the outer loop will go to the next index value and do same operations on it. These operations will do on all indexes of the array.

4. At the end print the sorted array.

**Source Code:**

**OUTPUT:**

