* Over-view:

As we are learning Assembly Language in our course, so we decided to construct a calculator using Assembly code. As our project is calculator based we have defined various functions like Addition, Subtraction, Multiplication, Division, Percentage and power in it. we have introduced two new modes Increment and Decrement in our Calculator. Which are rarely found in any calculator.

We have use different mnemonics and commands especially the use of loops and jump statements were beneficial in creating functions in order to run the code in a simple and efficient way.

* Scope:

Assembly Language helps the programmer to write human readable codes which are nearest to machine language. As we have designed a calculator with two unique methods/functions Increment and decrement. We can also construct other distinctive functions too;

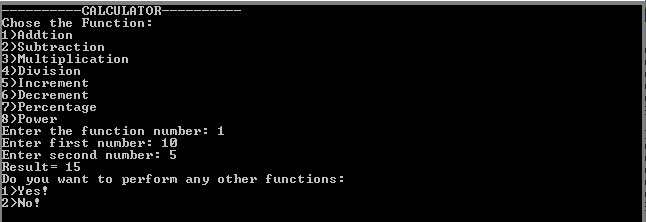
what was the purpose of creating this calculator?

The purpose for designing it was for the users to solve any type of problems. Rather than doing long calculations a user can simply use a calculator with unique methods to solve his problem within seconds. It is time efficient All this have been made easy because of Assembly language because,

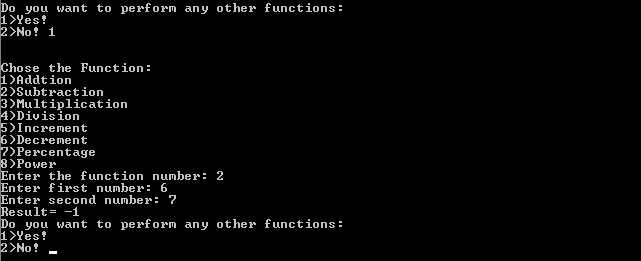
* It allows complex problems to run in simpler ways
* It is memory efficient.
* Requires less instructions to get the results.

Output result of different functions.

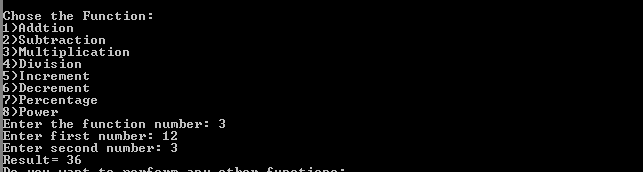
* Addition Function:



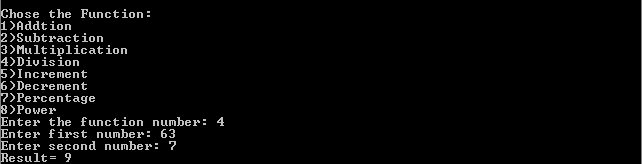
* Subtraction Function;



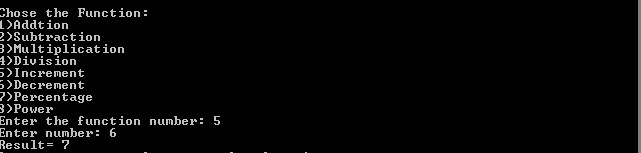
* Multiplication Function:



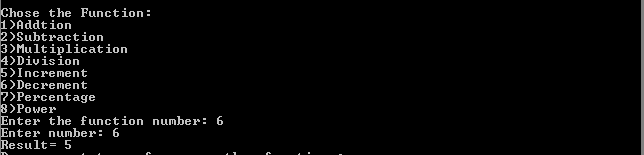
* Division Function:



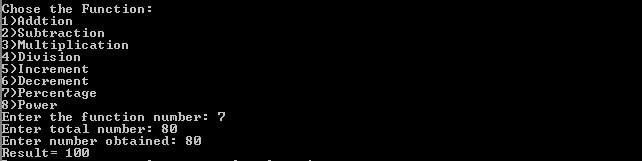
* Increment Function:



* Decrement Function:

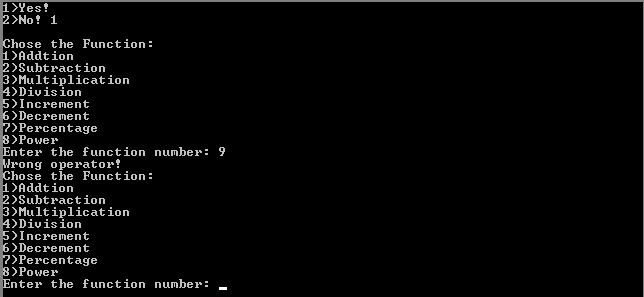


* Percentage Function:



* Power Function;



* In case if the user type a wrong operator value e.g (9)

The option of choosing the function will be asked again due to the jump condition.

* Conclusion:

The purpose of designing calculator was to do correct calculation efficiently. It should give user a relieve from doing the mental calculation and to need to rely on paper calculations.

**…………………………………………………………………………………………….**