

**A CLOSE LOOK
TO
THE
scurtY
PROGRAMMING
LANGUAGE**

By:Prafful Astle
Programmer-scurtY

scurtY

THE PROGRAMMING LANGUAGE OF NEXT GENERATION.

Programmed By: Prafful Astle

ABOUT THE AUTHOR



Prafful Astle is a 19 year old Computer Software programmer and

has state-of-the-art familiarity in the field of computers as he start using computers at the age of 5 and start programming at the age of just 14.

Mainly Prafful is a tech-Geek who want to get more and more information about technology .

He is well known for his own developed Programming language .

Prafful is currently perusing his bachelor's degree in computer science engineering and is working on his programming language to make Programming more understandable and easy.

The one and the only quote that Prafful uses while he Codes is :

"I f you do what you love, you'll never get bored of it "

PREFACE

Programming ComputerS now a days is a very useful task. Tons of students every year joins Engineering Colleges to become a Computer Engineer, There is a big reason behind it, About 90% of the total people admitted in computer science and engineering branch is there because of the vastness of the IT Sector now a days.

But sadly there are many people who don't even know what JAVA,C,C++ are.

when we tell them that they all are programming languages then they ask , "Whoa now what is programming language" so our teachers faces a large problem when they come to teach newest batch of every engineering college. And also when those people start learning it they starts taking it as a pressure and might their pressure cooker's (MIND) pressure releases after their external exams. huh., Passing the exam or only study to score high never gives out a perfect engineer. This is not their mistake they always work hard to be the best but the question is that "Is they are working hard in the right way??" the answer is No.

We all knows that food will not be perfect when we deep fry it in pre-hot oil.but it will be perfect when we fry it for the appropriate time on a mid flame.Similarly a student if want to become a CS engineer he/she have to prepare from the starting. But to start from the child age we don't have any easy language to work with. This is the reason behind the creation of **scurtY** Programming language. it have a short and easy syntax which gives the User what they want without any problem. Believe me it is really very easy to learn and when you get the result of your Program fast and Easily, you are going to love it. And according to me :

"I f you do what you love, you'll never get bored of it "

-Prafful Astle
Programmer of **scurtY**

ABOUT THIS BOOK

This book will work Both as the Reference book of **scurtY** **v0.1** and also as the installation guide and user manual of **scurtY** **v0.1** Integrated Development Program.

WHAT IS **scurtY** IDE ?


scurtY IDE is a Graphical User Interface Software for the programming language **scurtY** . It is also called the Integrated Development Environment of **scurtY** to create and run Computer program directly from the GUI IDE.

scurtY creates the programs which is directly executable without any further task it means, If your program is free from errors then it will be directly converted into ".exe" and can be run from the saved position.

Computer Programs created by **scurtY** is machine independent this means only caring the ".exe" file user can run it on any windows based machine from windows 95 to windows 8.1.

NOTE: **scurtY** is still developing and we need our users to report the bugs which they get while programming in **scurtY** because it will help us to make our software more open and free to use. And also it helps in expending the programming language.

Pre-Thanks.



CONTENTS:

• Introduction & Installation of scurtY IDE	08-09
• Getting Started With scurtY IDE	10-18
i. Getting Started	10
ii. Structure Of scurtY Programs	11
iii. The "SHOW()" Function	12
iv. Variable Types In scurtY	14
v. The "GRAB()" Function	17
• Understanding Operators & Expressions.	19-22
i. Revising The BODMAS rule in mathematics	19
ii. Operators In scurtY	19
iii. Creating Expressions in scurtY	21
• The" IF-ELSE " Statement	23-28
i. The Need Of Conditions In Programs	23
ii. The Simple IF-ELSE	23
iii. Programming time	24
• Working With "STRING"s .	29-32

i. What is STRING	29
ii. How To Tell scurtY About STRING	29
iii. Uses Of STRING	30
iv. The String Functions	30
• Working with LOOPS. (Creating The Guess My no. Game)	33-38
i. What Is LOOPING	33
ii. The "FROM" LOOP	33
iii. The "REPEAT-UNTILL" LOOP	35
iv. The STOPIT statement	36
v. The Guess My Number Game	36
• Data Variable In scurtY	39-43
i. What is temporary Data Storage	39
ii. How it works	39
iii. How to tell our program about data storage(declaration of data variable)-	40
iv. Use of Data Variable	40
v. Grabbing data from user(the "GRABDATA" function)	41
vi. showing data to user (the "SHOWDATA" function)	42

INTRODUCTION AND INSTALLATION OF scurtY

INTRODUCTION TO **scurtY**:

scurtY is a programming language which was developed in C & C++. The word scurtY comes from a Romanian word "scurt" which means Short & Précised.

As the name showing scurtY is a language which have short, précised and easy to use syntax.

scurtY is programming language which was created due to the problems faced by the people due to the large and tough syntax which is very tough to remember and that is why 90% of total errors comes in programming now a days is due to the incorrect syntax.

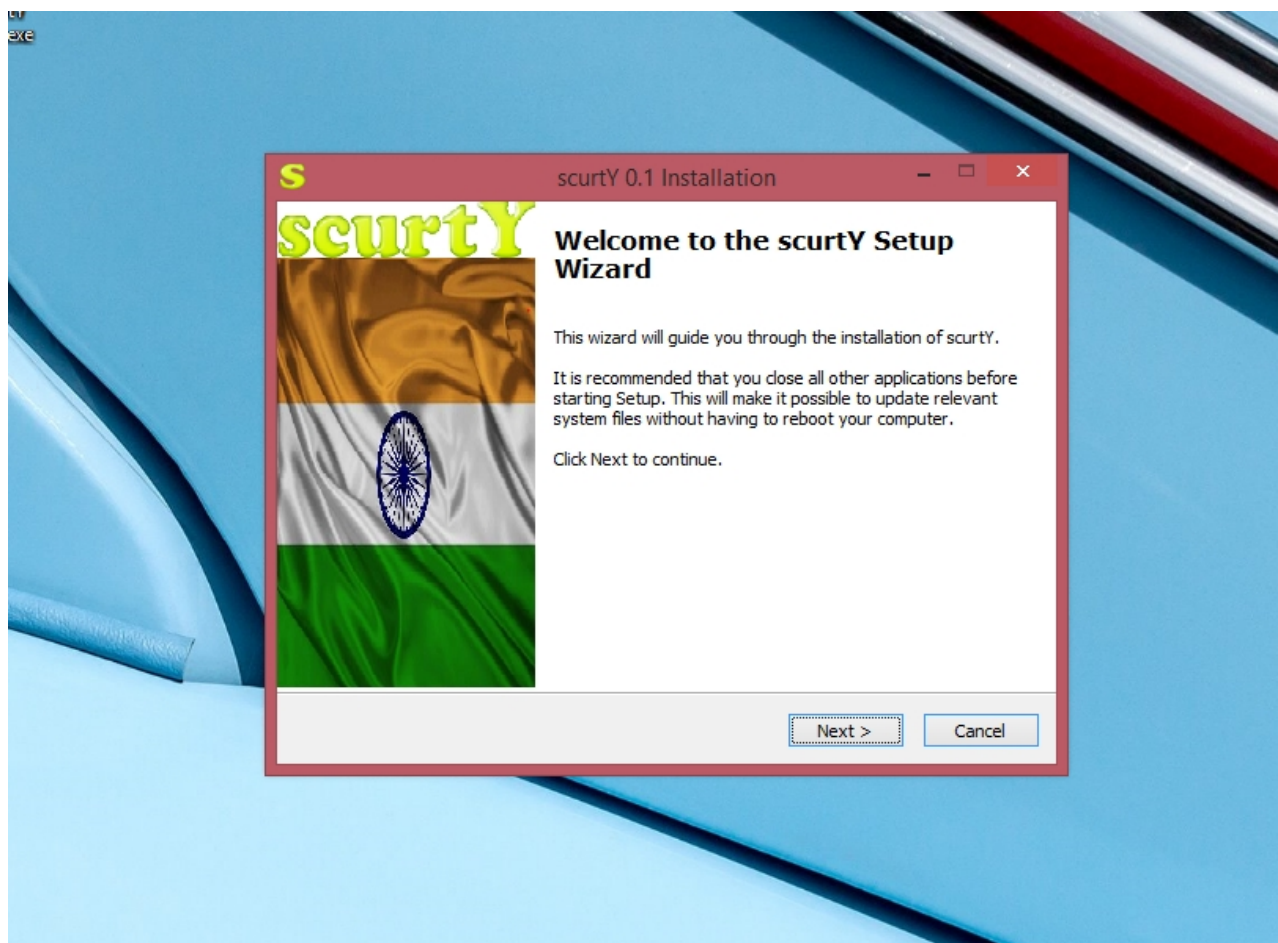
That is why in scurtY a very high level of English language is used to maintain the direct connection from computer memory to human memory.

INSTALLATION GUIDE

Here are some easy steps which can be used to setup **scurtY** in your Windows computer.

Step#1: Download **scurtY** v0.1 rom any website but I recommended to use its official website www.scurty.tk. (If you done this before just ignore this step).

Step#2: Now just double click on the "scurtY_Setup.exe" and click next.



Step#3: Now select the path where you want to install **scurtY**.

Step#4: Now click Next.

Step#5: Installation will take some time according to your cpu speed because it will decompress 31 mb to 104 mb.

Step#6: Click on Finish and you are done!!

*_*_*_*_*

GETTING STARTED WITH scurtyY IDE GETTING SLUBBED WITH IDE

TOOLBAR OF SCURTY IDE



In scurtyY IDE the toolbar is simple enough to understand but let us discuss the work of "compile and run" button. The compile and run button is the very first button in the toolbar and if user click it without saving the file then it opens a save dialog box because it is required to save the file before compiling it.

If your file is pre-saved then it first compile it and shows the errors one by one (if any) else it creates a ".exe" file which is the final output of your program.

Another silent feature of this button is that this can automatically save change your file whenever you click on it to compile.

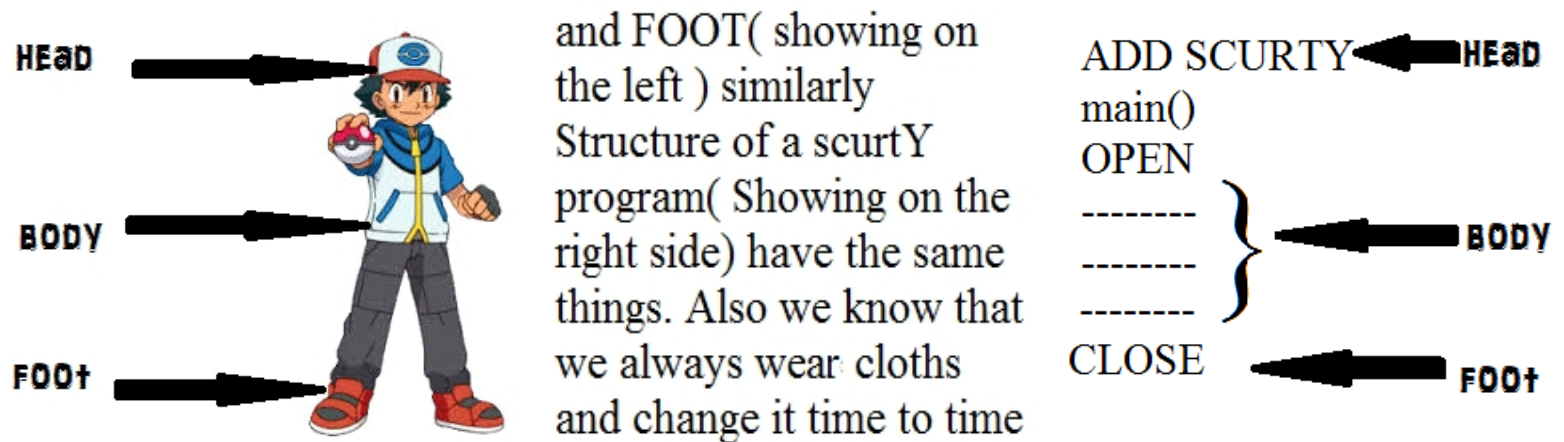
User can also compile,run,compile and run their program from the "BUILD" popup menu.

LETS START PROGRAMMING

Huh..after a long funky discussion let we start some serious Programming, hehe don't be so serious programming with scurtyY is as easy as boiling a cup of water. We just have to put the right thing at the right place and we will get the result as Quick as possible with scurty.

STRUCTURE OF scurtY PROGRAM

As we all know that our human body structure is consisting of A HEAD, A Body



but we never change our head and foot is it true? You know it is, similarly a scurtY program can be changed from its body, like we change our cloths to look different every time. In scurtY, Programmer can change its body to change its working but every time the head and the foot part will be the same.

Now, the Question arises in your mind that why we always "ADD SCURTY" in the program, Technically "ADD SCURTY" is the header file of the scurtY programming language which containing the functions used further in the programming.

Ohkay so now you got what "ADD SCURTY" is and why we use it always on the top.

NO?

In a very basic language "ADD SCURTY" is the signal to the scurtY compiler that the current file which is going to be compiled is related to scurtY programming language.

Now, in your mind there will be a question arises that what is that "*main()*" thing?

This is the *main()* function of scurtY program. Like our human body can do many fuctions at one time(viz., watching,eating etc) but the "*main*" function of our body is to breath if our body will failed to breath than every function of our body will stop working. Similarly although there are lots of functions present in the scurtY but the "*main()*" function of scurtY program will direct all of them.

Now, What about this "()" ?

Don't get panic this "()" is the symbol of function.

So just after the ADD SCURTY we will write the "*main()*" function as the syntax.

NOTE: The one and only thing which is in small letters in scurtY is that "*main()*" function and the remaining part of the program is in CAPITAL LETTERS.

Now, What about this "OPEN" and "CLOSE" ?

let we understand it but taking one more life example, If you want to write something in your copy you need to "OPEN" it, and when your work get finish you always close your copy. Am i right, You know i am, Similarly if you want to write something in the "*main()*" function then the "OPEN" & "CLOSE" will tell the scurty compiler that from where your coddng is started.

WHAT DID YOU LEARN TILL NOW?

So till now you learn about the "ADD SCURTY" header, the "*main()*" Function of the scurty program and how to OPEN and CLOSE it.

It is very easy i think, That is the advantage of scurtY that its *syntax* is too closer to the *english* language.

Now, Lets we move *a step ahead*.

THE "SHOW()" FUNCTION

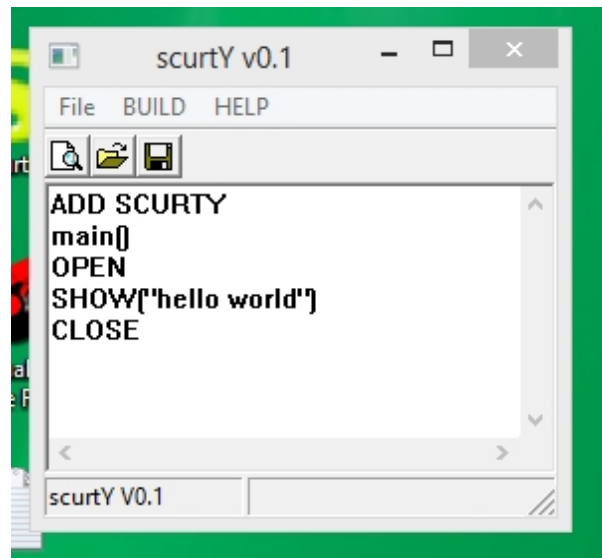
The "SHOW()" function is the SYNTAX used by scurtY program to show something on screen whatever you want to show on you program screen comes under this function "SHOW()" but remember to write it in capital letters always.

Here is the way to show something on screen:

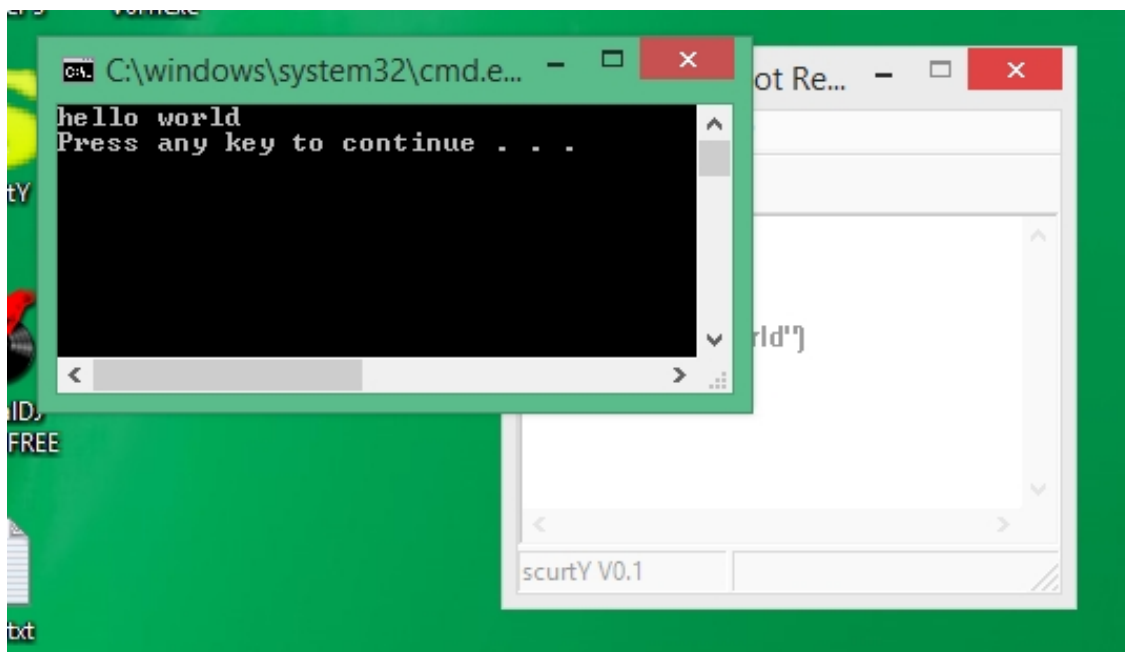
SHOW("anything")

here you see a **SHOW()** which is the function name and signature, and **anything** which is a string (a collection of many alphabets or characters) covered by double-quotes ("") which is the signature of strings.

Now, you are free for creating your very first scutY program, Here we go:



and then click on compile and run button, Save it as and again when you compile you will get the output like this:



Now go and check it out, Where you save your .scurtY program you will have a .EXE file of similar name as of your program.

Congratulations

You are a programmer now,

You had created your very first software which is directly runnable at anytime

anyplace.

isn't easy? I know it is,

okay so let we move further,

Creating your software is okay but it is required that it have to do something like before your software says "hello" to the world Now moving further let we discuss about some more useful things which help you to create some very useful softwares like a calculator etc.

VARIABLE TYPES IN scurtY

A variable, Did you know the meaning of this word "variable", A variable is something whose value is not fixed. It can vary time to time.

In scurtY variable is of four types viz., INTEGER variable, POINT variable. CHARACTER variable, STRING variable.

INTEGER VARIABLE: It is the variable type which have only integer value. Which means that the no -infinity to +infinity can store in this kind of variable but only integer type.

SYNTAX:

INT first_variable_name,Second_variable_name and so on.....

also you can tell the value of the variable to scurtY by variable_name followed by "=" followed by its value as an example,

INT abc=47,qwe=87,qs

in this variable declaration you tell the compiler that there is a variable of integer whose name is "abc" and who is having a value "47" also another variable "qwe" of value "87" and a variable "qs" whose value is unknown.

in this case when any value of a variable is unknown scurtY automatically gives it any bulk value.

POINT VARIABLE: It is the variable type which have integer as well as a value

which contains a decimal point.

SYNTAX:

POINT first_variable_name,Second_variable_name and so on.....

also you can tell the value of the variable to scurty by variable_name followed by "=" followed by its value as an example,

POINT abc=47.4,qwe=8.7,qs

in this variable declaration you tell the compiler that there is a variable of point whose name is "abc" and who is having a value "47.4" also another variable "qwe" of value "8.7" and a variable "qs" whose value is unknown.

in this case when any value of a variable is unknown scurtY automatically gives it any bulk value.

CHARACTER VARIABLE: It is the variable type which have only a single character value. Which means that any alphabet small or in caps any no as a character or any symbol can store in this kind of variable but only one at a time.

SYNTAX:

CHAR first_variable_name,Second_variable_name and so on.....

also you can tell the value of the variable to scurty by variable_name followed by "=" followed by its value **in single quotes (' ')** as an example,

CHAR abc='h',qwe='e',qs

in this variable declaration you tell the compiler that there is a variable of characters whose name is "abc" and who is having a value "47" also another variable "qwe" of value "87" and a variable "qs" whose value is unknown.

in this case when any value of a variable is unknown scurtY automatically gives it any bulk value.

STRING VARIABLE: It is the variable type which have a string(group of characters) as its value.

SYNTAX:

STRING first_variable_name,Second_variable_name and so on.....

also you can tell the value of the variable to scurty by variable_name followed by "=" followed by its value **in Double quotes (" ")** as an example,

STRING abc="hello",qwe="world",qs

in this variable declaration you tell the compiler that there is a variable of words whose name is "abc" and who is having a value "hello" also another variable "qwe" of value "world" and a variable "qs" whose value is unknown.

in this case when any value of a variable is unknown scurtY automatically gives it any bulk value.

in a simple and easy way scurtY have 4 kind of variables:

#1. INT who stores integers only.

#2. POINT which stores integer as well as decimal value.

#3. CHAR which stores a singular character value in (' ').

#4. STRING which stores a group of character as its value in (" ").

Now, we know how to declare the variables and how to store values at the declaring time. and now we are going to learn how to change its value:

to change the value of any variable we use this SYNTAX

variable_name=0+(new_value OR any_other_variable)

as an example,

abc=0+qwe :It changes the value of *abc* to *qwe*

OR

abc=0+12 :It changes the value of *abc* to **12**

Character and string literals have certain peculiarities, like the escape codes. These are special characters that are difficult or impossible to express otherwise in the source code of a program, like newline (\n) or tab (\t). All of

them are preceded by a backslash (\). Here you have a list of some of such escape codes:

\n	newline
\r	carriage return
\t	tab
\v	vertical tab
\b	backspace
\f	form feed (page feed)
\a	alert (beep)
\'	single quote (')
\"	double quote (")
\?	question mark (?)
\\	backslash (\)

For example:

```
'\n'  
'\t'  
"Left \t Right"  
"one\ntwo\nthree"
```

Now, you know how to use variables in scurtY programming language. Now we are moving more further, sometimes it is very necessary to get something from the user of your software, So in the next step we are going to do this also.

THE "GRAB()" FUNCTION

The standard input device is usually the keyboard. Handling the standard input in scurtY is done by the "GRAB()" function, as the name showing this function have the ability to catch what user type on the screen through keyboard.

SYNTAX:

GRAB(variable_name)

Here the variable used have to be declared before its use. Also if you want to add some message before grabbing the value from the user GRAB() can be used as follows:

GRAB('message',variable_name)

Also you can grab a no of variables at a single time by using GRAB() function like this:

GRAB('message',variable_name1,variable_name2) and so on.

EXPENDING THE SHOW() FUNCTION

As now you got that how to use variable in scurtY we can now expend The SHOW() function to show variables on the screen also with the messages.

To show variables with the appropriate messages you can use the SHOW() function as follows:

SHOW('message',variable_name1,variable_name2)

or you can use any combination of Message and variables followed by the comma(.). like

SHOW('message',variable_name1,'message',variable_name2)

Note:This property is only present in the SHOW() function not in the GRAB() function. in GRAB() function you can only show a message before grabbing the variable value.



UNDERSTANDING OPERATORS AND EXPRESSIONS

REVISING THE BODMAS RULE

BODMAS, hope everybody in this world is familiar with it. For those who are not from this world, BODMAS is a sequential pattern of solving mathematics expressions. It stands For

B-brackets

O-of

D-divide

M-multiply

A-addition

S-subtraction

According to this we solve normal mathematical expression which contains $()$, $/$, x , $+$, $-$. Also here in scurty you can solve mathematical expressions by using this BODMAS rule.

OPERATORS IN scurtY

'/' , '*' , '-' , '+' , '(' , ')' etc these are called as Operators. Almost every calculation has been done from these operators. In scurty they are very useful, let us discuss how we can use them in our programming.

There are four kind of operators in scurtY:

- Arithmetic operators ($+$, $-$, $*$, $/$, $\%$)

- Increase and decrease (++, --)
- Conditional Operators (==, !=, >, <, >=, <=)
- Logical operators (AND, OR)

Arithmetic Operators:

The five arithmetical operations supported by the scurtY language are: + addition - subtraction * multiplication / division % modulo

Operations of addition, subtraction, multiplication and division literally correspond with their respective mathematical operators. The only one that you might not be so used to see is modulo; whose operator is the percentage sign (%). Modulo is the operation that gives the remainder of a division of two values. For example, if we write:

```
a = 11 % 3
```

the variable a will contain the value 2, since 2 is the remainder from dividing 11 between 3.

Increase and decrease (++, --) :

Shortening even more some expressions, the increase operator (++) and the decrease operator (--) increase or reduce by one the value stored in a variable. They are equivalent to +=1 and to -=1, respectively. Thus:

```
c=c+1 and c++
```

are all equivalent in its functionality: they increase by one the value of c.

Conditional Operators (==, !=, >, <, >=, <=)

In order to evaluate a comparison between two expressions we can use the relational and equality operators. The result of a relational operation is a Boolean value that can only be true or false, according to its Boolean result.

We may want to compare two expressions, for example, to know if they are equal or if one is greater than the other is. Here is a list of the relational and equality operators that can be used in scurtY

== Equal to
!= Not equal to
> Greater than
< Less than
>= Greater than or equal to
<= Less than or equal to

Here there are some examples:

(7 == 5) // evaluates to false.
(5 > 4) // evaluates to true.
(3 != 2) // evaluates to true.
(6 >= 6) // evaluates to true.
(5 < 5) // evaluates to false.

Of course, instead of using only numeric constants, we can use any valid expression, including variables.

These operators are very helpful in if statements.

Logical operators (AND, OR) :

The logical operators **AND** and **OR** are used when evaluating two expressions to obtain a single relational result. The operator **AND** corresponds with Boolean logical operation AND. This operation results true if both its two operands are true, and false otherwise. The following panel shows the result of operator **AND** evaluating the expression **a AND b**:

a	b	a AND b
true	true	true
true	false	false
false	true	false
false	false	false

The operator **OR** corresponds with Boolean logical operation OR. This operation results true if either one of its two operands is true, thus being false only when both operands are false themselves. Here are the possible results of a **OR** b:

a	b	a OR b
true	true	true
true	false	true
false	true	true
false	false	false

For example:

`((5 == 5) && (3 > 6)) // evaluates to false (true && false).`

`((5 == 5) || (3 > 6)) // evaluates to true (true || false).`

CREATION OF EXPRESSIONS

Creating expression in scurty is not a big tough task. as we discussed earlier in this book that scurty make you do thing very easy, simple and fast now you are going to understand why.

you can create expressions in a very easy way like,

```
INT a=5,b=7,c=8,sum=0
```

```
sum=a+b+c
```

```
SHOW("total is=",sum)
```

this codes shows the sum of a,b,c similarly you can create your own expression for any formula like,

$$e=mc^2$$

the expression is $e=m*c*c$

$$\text{and S.I.} = \frac{PRT}{100}$$

the expression in $si=(P*R*T)/100$ and so on.



THE IF-ELSE STATEMENT

NEED OF CONDITIONS IN PROGRAM

Conditions, means the result will be in true or false. In the last chapter we studied about Conditional operators which give such true or false result.

Suppose that if you are developing a game and you only want to show the message of *game over* when the game is really over else you want to continue the game, This can be done only by conditions and here your condition is :

if game_over is TRUE

show game over

else nothing will happen.

here in simple english we use if-else and hence keeping in mind the english language scurty also uses IF-ELSE statement to check weather the condition is true or not.

THE SIMPLE IF ELSE

The IF keyword is used to execute a statement or block only if a condition is fulfilled. Its form is:

IF(condition) statement

Where condition is the expression that is being evaluated. If this condition is true, statement is executed. If it is false, statement is ignored (not executed) and the program continues right after this conditional structure. For example, the following code fragment shows x is 100 only if the value stored in the x variable is indeed 100:

```
IF (x == 100)  
SHOW("x is 100")
```

If we want more than a single statement to be executed in case that the condition is true we can specify a block using braces OPEN and CLOSE:

```
IF(x == 100)  
OPEN  
SHOW( "x is ",x)  
SHOW("you got it")  
CLOSE
```

We can additionally specify what we want to happen if the condition is not fulfilled by using the keyword ELSE. Its form used in conjunction with IF is:

```
IF (condition)  
statement1  
ELSE  
statement2
```

For example

```
IF(x == 100)  
SHOW("x is 100")  
ELSE  
SHOW("x is not 100")
```

prints on the screen x is 100 if indeed x has a value of 100, but if it has not -and only if not- it prints out x is not 100

NOTE: we can only use expressions with conditional operators to check true or false.

THE NESTED IF ELSE

The IF + ELSE structures can be concatenated with the intention of verifying a range of values. The following example shows its use telling if the value currently stored in x is positive, negative or none of them (i.e. zero):

```
IF(x > 0)
SHOW("x is positive")
ELSE IF(x < 0)
SHOW("x is negative")
ELSE
SHOW("x is 0")
```

Remember that in case that we want more than a single statement to be executed, we must group them in a block by enclosing them in OPEN-CLOSE.



Okay so finally the time is come when you can start some sort of programming.

IF-ELSE is very useful in programming and also it is called the base of programming, let we start from the easiest and increase the level program by program, in this programming section there are 3 sections beginner, Moderate and Expert which total contains 6 programs 3 of them are solved and remaining 3 are unsolved. I placed them alternatively and it is on you to understand the example program and solve the task by your own.

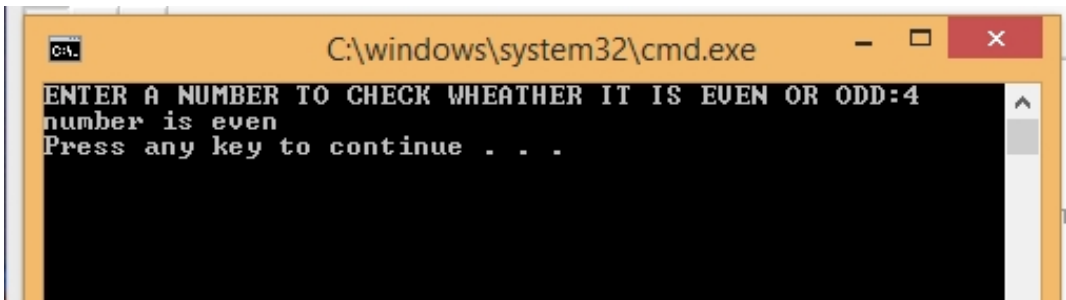
LEVEL 1: BEGINEER

EXAMPLE: Write a software in scurtY which can find the entered number is

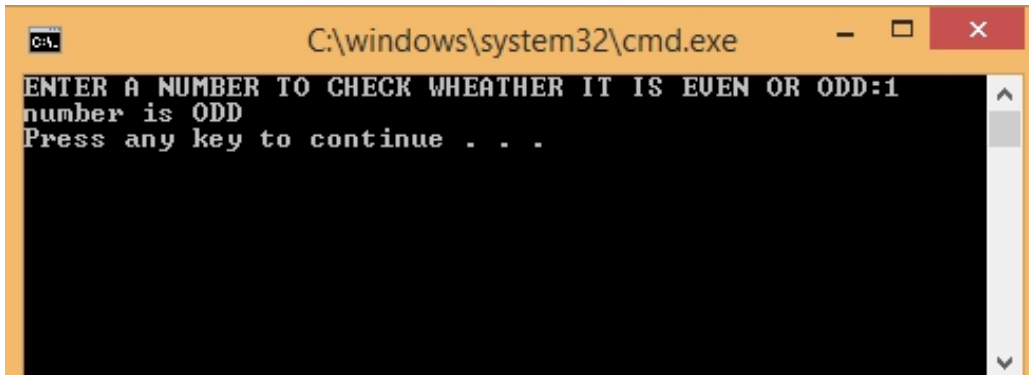
even or odd.

```
ADD SCURT Y
main()
OPEN
INT number=0
GRAB("ENTER A NUMBER TO CHECK WHEATHER IT IS EVEN OR ODD:",number)
number=number%2
IF(number==0)
SHOW("number is even")
ELSE
SHOW("number is ODD")
CLOSE
```

and here is the test output #1



test output #2

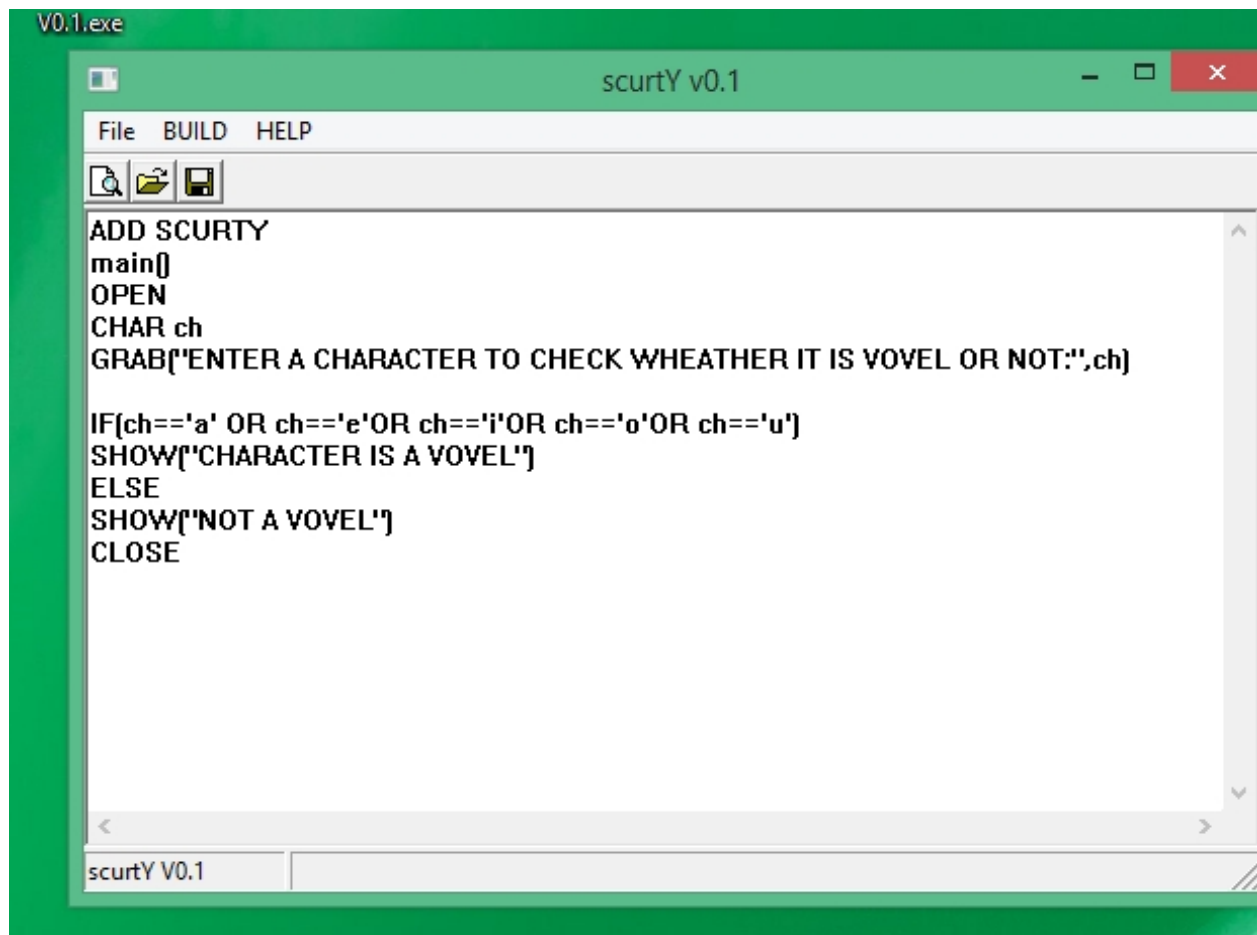


TASK: Write a software in scurtY to check whether the entered number is divisible by 7 or not.

hints: grab a no. > check if it gives rimender when divides by 7 >if yes then not divisible else >divisible

LEVEL 2: MODERATE

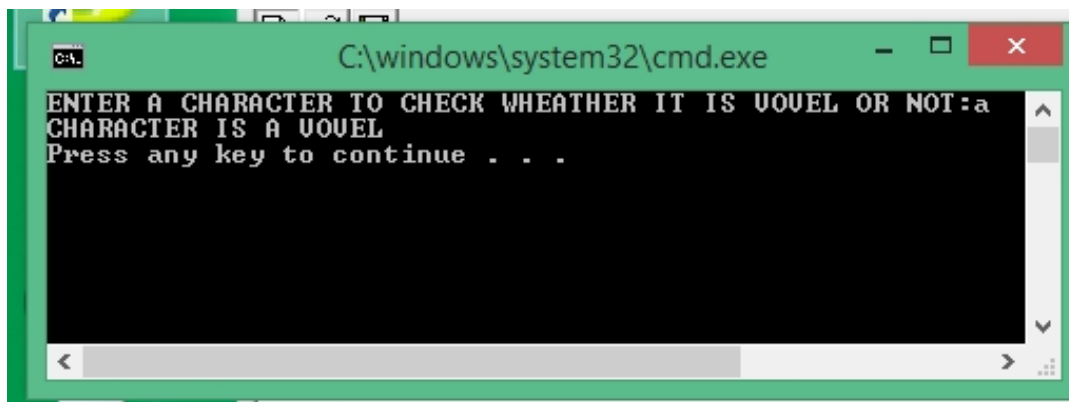
EXAMPLE: EXAMPLE: Write a software in scurY to check whether the entered character is a vowel or not.



```
V0.1.exe
scurY v0.1
File BUILD HELP
ADD SCURTY
main()
OPEN
CHAR ch
GRAB["ENTER A CHARACTER TO CHECK WHEATHER IT IS VOVEL OR NOT:",ch)

IF[ch=='a' OR ch=='e'OR ch=='i'OR ch=='o'OR ch=='u']
SHOW["CHARACTER IS A VOVEL"]
ELSE
SHOW["NOT A VOVEL"]
CLOSE
```

and here is the output

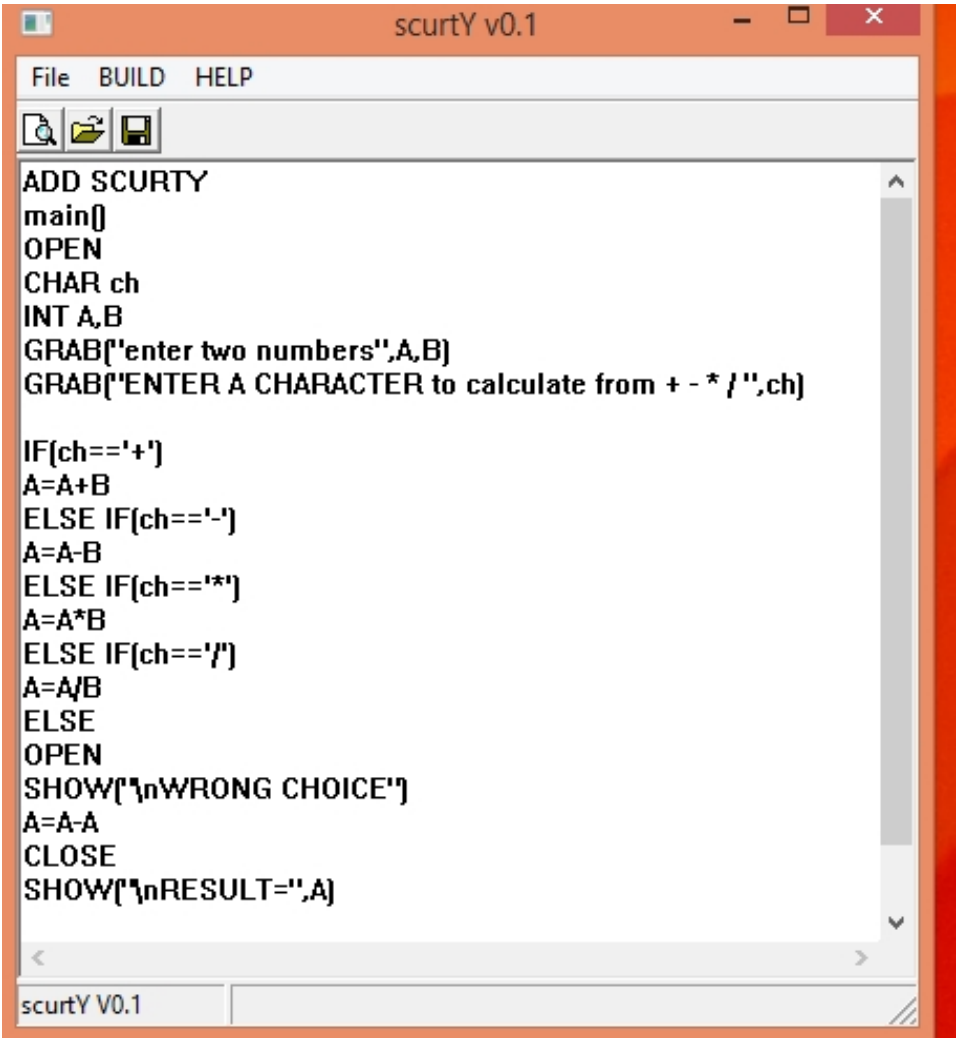


```
C:\windows\system32\cmd.exe
ENTER A CHARACTER TO CHECK WHEATHER IT IS VOVEL OR NOT:a
CHARACTER IS A VOVEL
Press any key to continue . . .
```

TASK: Write a software in scurY to whether the entered character is consonant or not.

LEVEL 3: EXPERT

EXAMPLE: Write a simple calculator Software in scurtY.



The screenshot shows a window titled "scurtY v0.1" with a menu bar containing "File", "BUILD", and "HELP". Below the menu bar are three icons: a magnifying glass, a folder, and a document. The main area is a text editor displaying the following code:

```
ADD SCURTY
main()
OPEN
CHAR ch
INT A,B
GRAB("enter two numbers",A,B)
GRAB("ENTER A CHARACTER to calculate from + - * / ",ch)

IF(ch=='+')
A=A+B
ELSE IF(ch=='-')
A=A-B
ELSE IF(ch=='*')
A=A*B
ELSE IF(ch=='/')
A=A/B
ELSE
OPEN
SHOW("\nWRONG CHOICE")
A=A-A
CLOSE
SHOW("\nRESULT=",A)
```

The status bar at the bottom left shows "scurtY V0.1".

Now your task is to Modify the last software by adding more functions like decimal calculation and reminder finder. also you need to find the correct output.



WORKING WITH STRINGS

WHAT IS A STRING

String, which is discussed earlier in this book is the collections of characters to make a word or whole sentence.

HOW TO TELL SCURTY ABOUT STRING

To tell scurty about strings you need to use string variable which is discussed earlier also.

STRING VARIABLE: It is the variable type which have a string(group of characters) as its value.

SYNTAX:

STRING first_variable_name,Second_variable_name and so on.....

also you can tell the value of the variable to scurty by variable_name followed by "=" followed by its value in **Double quotes** (" ") as an example,

STRING abc="hello",qwe="world",qs

in this variable declaration you tell the compiler that there is a variable of words whose name is "abc" and who is having a value "hello" also another variable "qwe" of value "world" and a variable "qs" whose value is unknown.

in this case when any value of a variable is unknown scurtY automatically gives it any bulk value.

a string can be used in various ways like : to store somebodies name and information etc, to store answers of questions in a KBC game.etc etc

THE STRING FUNCTIONS

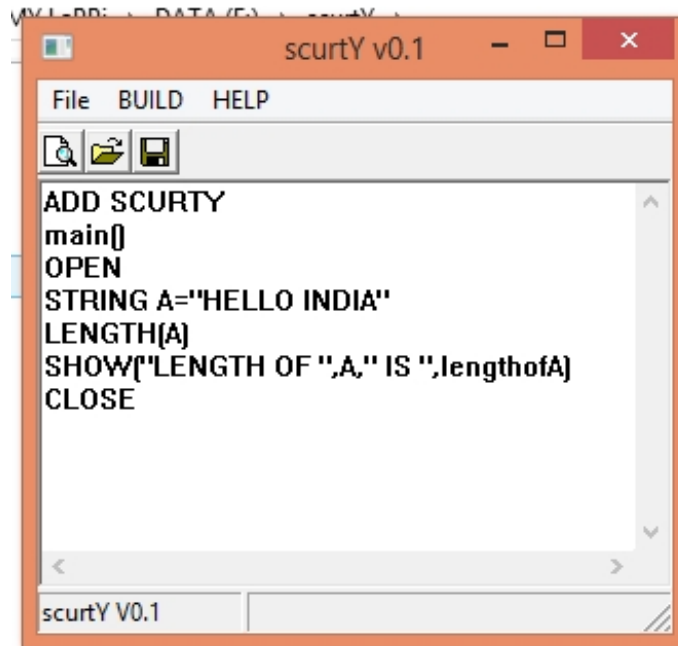
THE *LENGTH()* FUNCTION:

The length function is the most useful and advance function in scurty programming.

the length function have the ability to find the length of a string (no. of characters in it) and also it creates a variable called the "lengthofvariable_name" which we can use anytime anyplace.

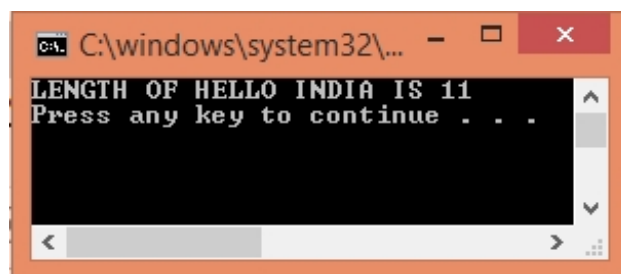
SYNTAX:

LENGTH(variable_of_string)



```
ADD SCURTY
main()
OPEN
STRING A="HELLO INDIA"
LENGTH(A)
SHOW["LENGTH OF ",A," IS ",lengthofA]
CLOSE
```

WHICH GIVES THE OUTPUT



```
C:\windows\system32\...
LENGTH OF HELLO INDIA IS 11
Press any key to continue . . .
```

WE can use "*lengthofvariable_name*" at any place as it works as a normal variable.

THE *COPYSTRING()* FUNCTION:

The *COPYSTRING()* function in scurty language used to copy second string to first pre-defined string variable, it can also copy a constant string to the first pre-defined string SYNTAX as follows:

COPYSTRING(variable,variable)

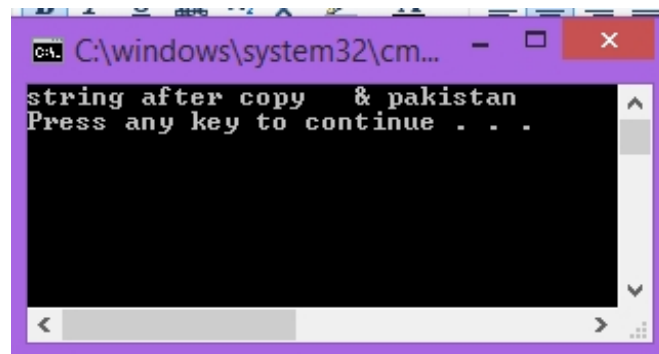
or

COPYSTRING(variable,"constant string")

NOTE:It replaces the first string from second.



which produces the output



THE *JOIN()* FUNCTION:

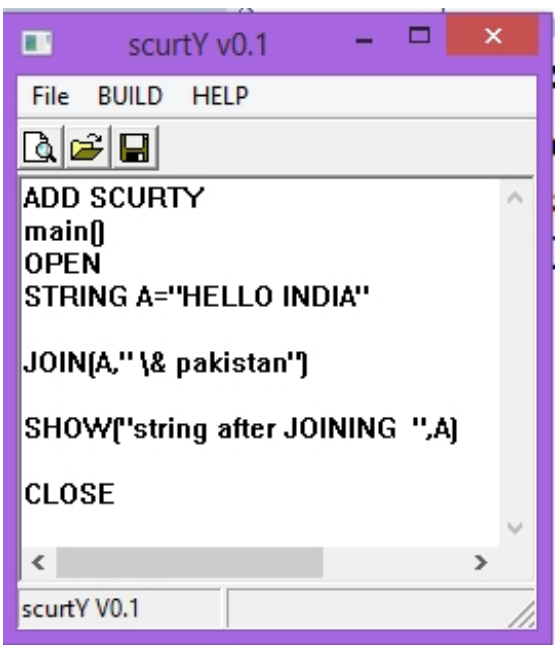
The *JOIN()* function in scurty language used to JOIN second string to first pre-defined string variable, it can also JOIN a constant string to the first pre-defined

string SYNTAX as follows:

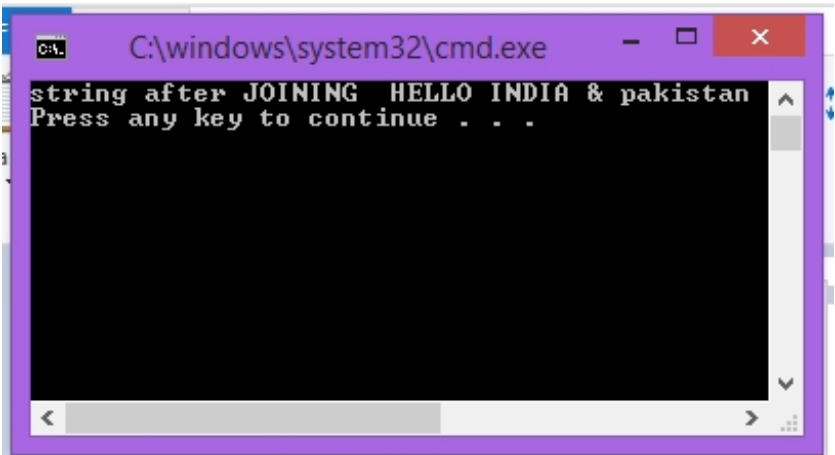
JOIN(variable,variable)

or

JOIN(variable,"constant string")



which gives the output



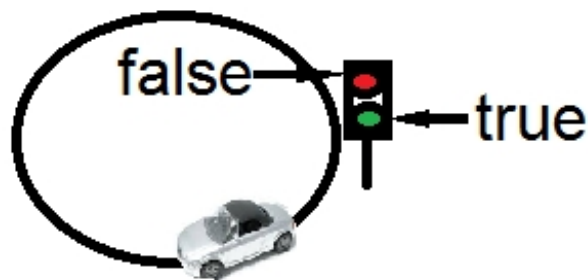
WORKING WITH LOOPS

WHAT IS A LOOP

Generally, A loop is a round path. In programming also, but here we have a condition. A condition which is required to be fulfilled every time the loop runs, it is like a round racing track which has a certain condition of driving by the speed of 40. If your speed increases then your car will stop.



NORMAL LOOP



PROGRAMMING LOOP

There are basically Two kinds of loop:

1. Entry control loop.(Which test condition before the start of loop).
e.g.FROM loop
- 2.Exit control loop (Which test the condition after execution of loop).
e.g. REPEAT UNTIL loop.

THE FROM TO DO LOOP

This loop is a kind of entry control loop which test the condition first if the condition is true then the other thing will happen otherwise not. The syntax of FROM-TO-DO loop is as follows:

FROM *initialization* **TO** *condition* **DO** *upgradation* *statement*

Its main function is to repeat statement while condition remains true. But in addition, the from loop provides specific locations to contain an initialization statement and an increase statement. So this loop is specially designed to perform a repetitive action with a counter which is initialized and increased on each iteration.

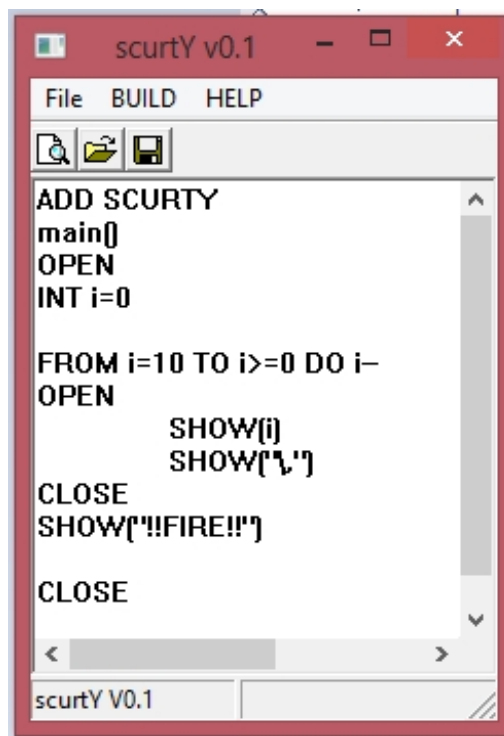
It works in the following way:

1. initialization is executed. Generally it is an initial value setting for a counter variable. This is executed only once.
2. condition is checked. If it is true the loop continues, otherwise the loop ends and statement is skipped (not executed).
3. statement is executed. As usual, it can be either a single statement or a block enclosed in OPEN & CLOSE.
4. finally, whatever is specified in the increase field is executed and the loop gets back to step 2.

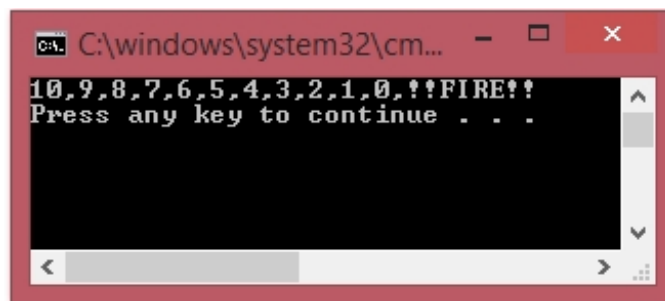
GRAPHICALLY



let see how it helps in programming with the help of example:



which gives the output



THE REPEAT UNTILL LOOP

It is a Exit control loop and its SYNTAX is:

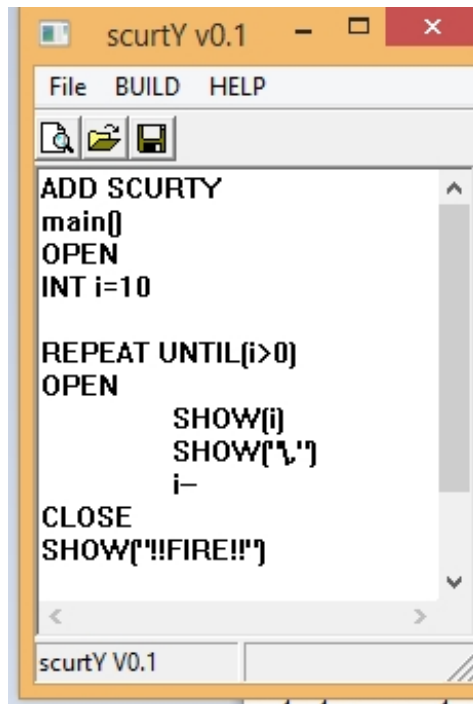
REPEAT UNTIL(*condition*)

statement

Its functionality is exactly the same as the FROM TO DO loop, except that condition in the REPEAT UNTIL loop is evaluated after the execution of statement instead of before, granting at least one execution of statement even if

condition is never fulfilled.

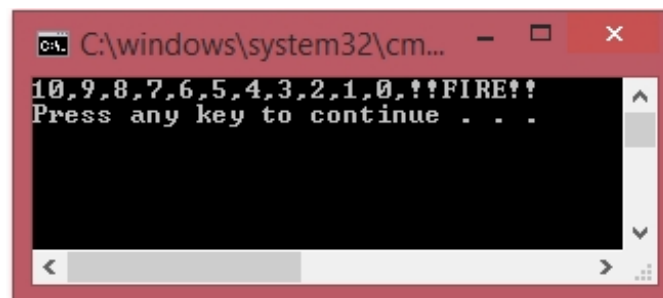
Let we make the same software with REPEAT UNTIL loop



```
File BUILD HELP
ADD SCURTY
main()
OPEN
INT i=10

REPEAT UNTIL[i>0]
OPEN
    SHOW(i)
    SHOW("\n")
    i--
CLOSE
SHOW("!!FIRE!!")
```

which gives the similar output



```
C:\windows\system32\cm...
10,9,8,7,6,5,4,3,2,1,0,!!FIRE!!
Press any key to continue . . .
```

We can also use both loops any time at any place and also we can use the nested loop.

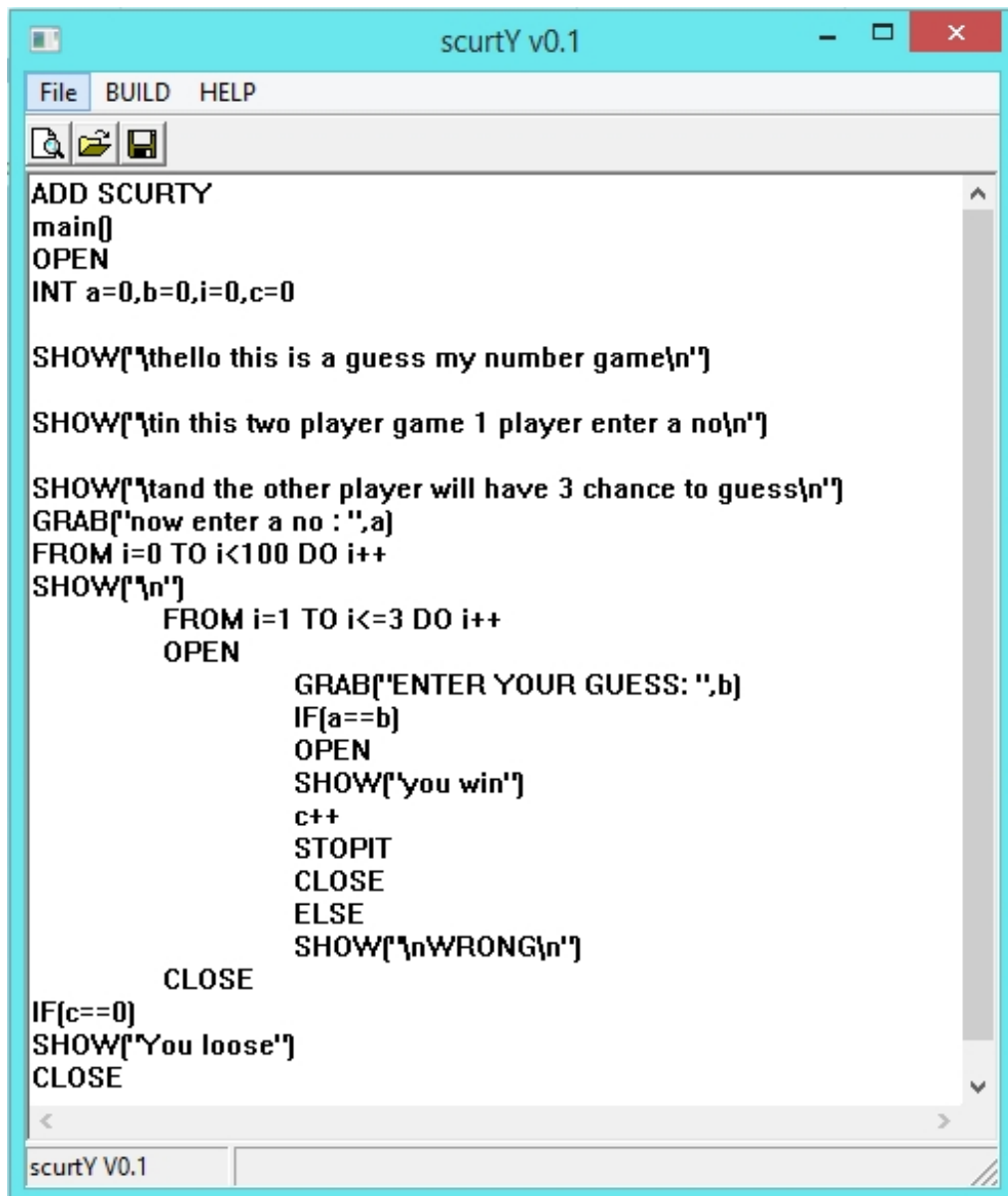
THE STOPIT STATEMENT

The STOPIT statement is very useful in the world of programming, According to the name it will stop the further processing of any loop or if-else statement.

just simply write : STOPIT

anywhere in the loop body or if-else body.

THE GUESS MY NUMBER GAME



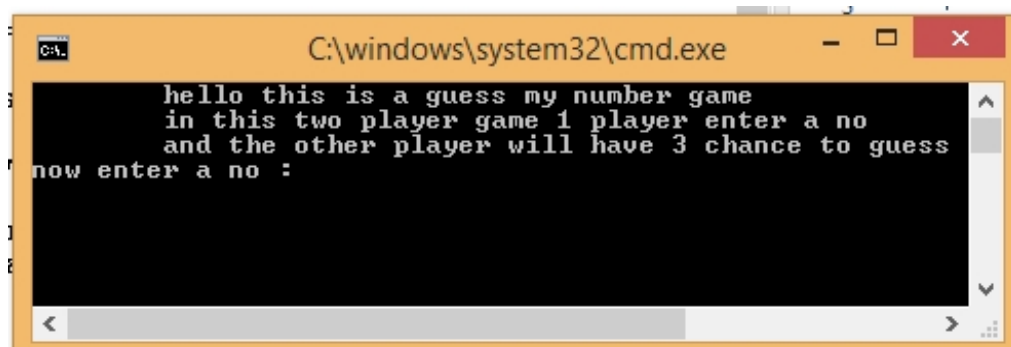
```
File BUILD HELP
ADD SCURTY
main()
OPEN
INT a=0,b=0,i=0,c=0

SHOW["\thello this is a guess my number game\n"]

SHOW["\tin this two player game 1 player enter a no\n"]

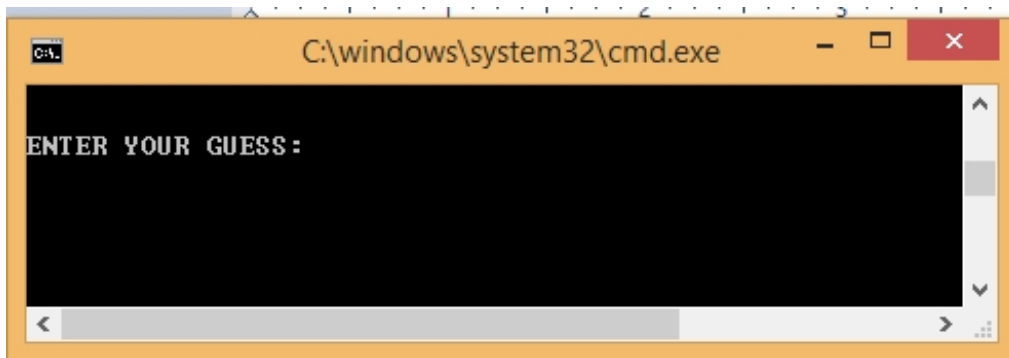
SHOW["\tand the other player will have 3 chance to guess\n"]
GRAB['now enter a no : ',a]
FROM i=0 TO i<100 DO i++
SHOW["\n"]
    FROM i=1 TO i<=3 DO i++
    OPEN
        GRAB['ENTER YOUR GUESS: ',b]
        IF[a==b]
        OPEN
        SHOW['you win']
        c++
        STOPIT
        CLOSE
        ELSE
        SHOW["\nWRONG\n"]
    CLOSE
IF[c==0]
SHOW["You loose"]
CLOSE
```

here is the output show first:

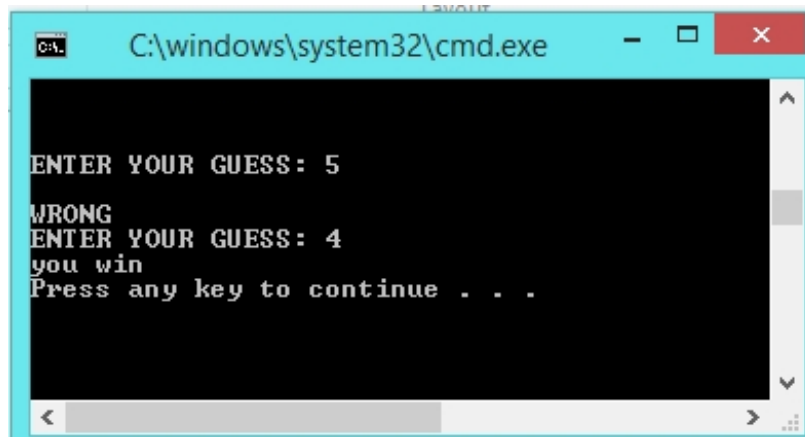


```
CA. C:\windows\system32\cmd.exe
hello this is a guess my number game
in this two player game 1 player enter a no
and the other player will have 3 chance to guess
now enter a no :
```

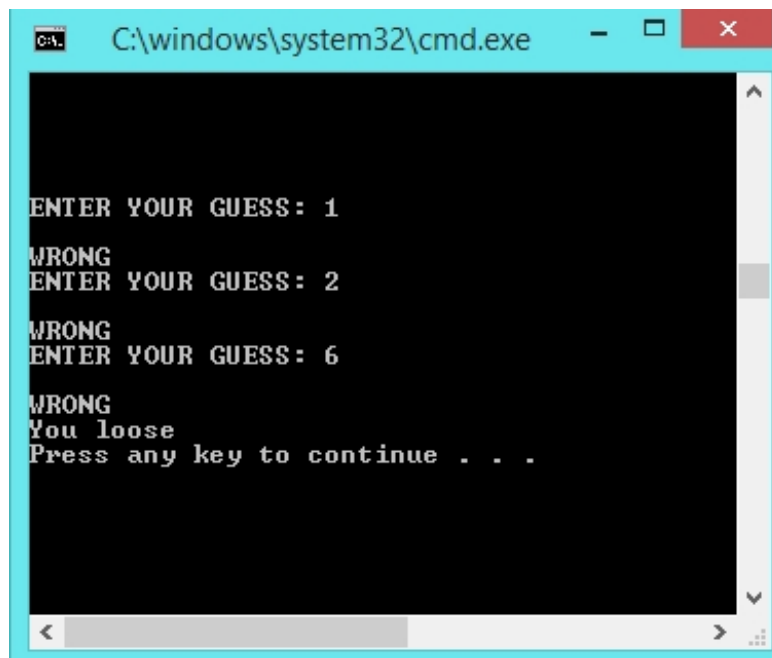
let user enter :4



let 2 user enter 5 and then 4



if user fails to guess:



Now your task is to create this game for character guessing and also introduce a third player in this game.



DATA VARIABLE IN scurtY

WHAT IS TEMPORARY DATA STORAGE

Temporary Data is that data which is collected by a software when it runs, Normally in small software all of the data is stored in an array.

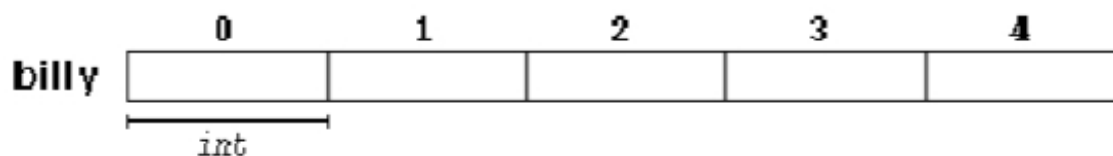
An array is a collection of similar kind of data types.

Here in scurtY, we call an array as a DATAVARIABLE. The main reason behind it is that it is really a data variable which stores a lot of data in a single variable, like if you want to store computer science marks of five students you will create five integer variables but here, you can also use a data variable to store these five values.

HOW DATA VARIABLE WORKS

Let we start with an example like,

A DataVariable to contain 5 integer values of type INT called billy could be represented like this:



where each blank panel represents an element of the DATAVARIABLE, that in this case are integer values of type INT. These elements are numbered from 0 to 4 since in DATAVARIABLES the first index is always 0, independently of its

length.

DECLARATION FOR A DATAVARIABLE

Like a regular variable, A DATAVARIABLE must be declared before it is used. A typical declaration for a DATAVARIABLE in scurY is:

DATAOFINT *nameofvariable.size* : for integer type datavariabile.

DATAOFPOINT *nameofvariable.size* : for decimal type datavariabile.

DATAOFCHAR *nameofvariable.size* : for single character type datavariabile.

where *nameofvariable* is a valid identifier and the *size* field (which is always have a dot '.' before it), specifies how many of these elements the DATAVARIABLE has to contain.

Therefore, in order to declare an array called billy as the one shown in the above diagram it is as simple as:

DATAOFINT billy.5

NOTE: The size field after dot (.) which represents the number of elements the DATAVARIABLE is going to hold, must be a constant value, since DATAVARIABLE are blocks of non-dynamic memory whose size must be determined before execution. In order to create DATAVARIABLE with a variable length dynamic memory is needed.

USE OF A DATAVARIABLE

DATAVARIABLEs can be used normally as a variable in any expression etc. The way of using it is,

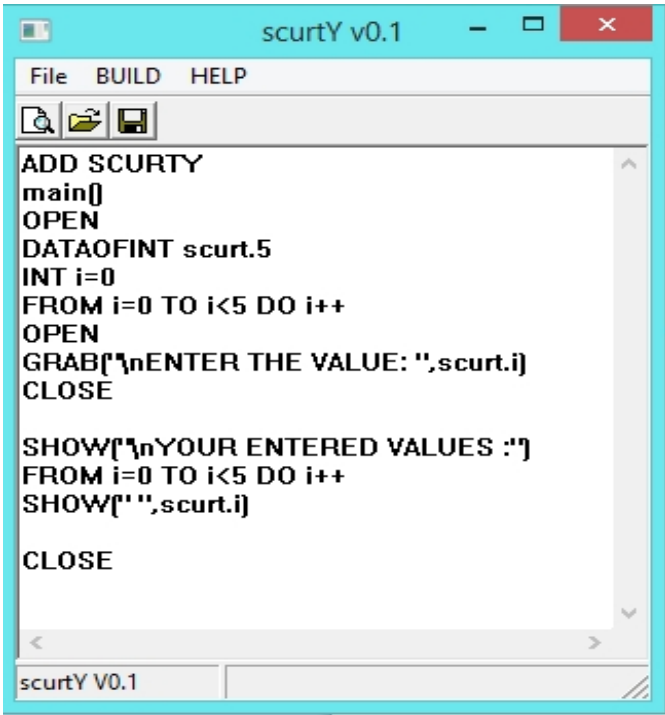
name●element_position

like if you have a data variable of name "marks" and which have stored 5 elements in it and you want to use the marks at 2 position so you just have to write :

marks.2

Note:scurtY is still developing so in this version we are not able to use data variables in if-else and loops directly we can use a secondary variable to store the value of that data variable. In the next update these bugs will be fixed.

now let see how we can grab data from user in a data variable and how to show them.

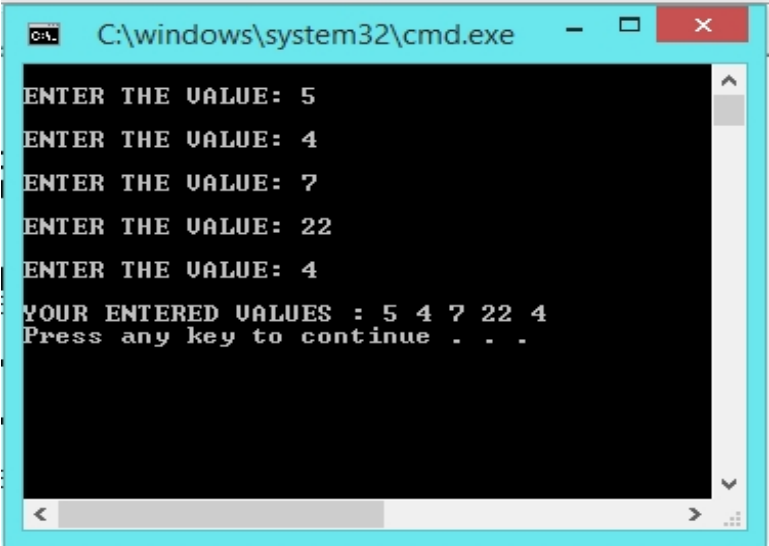


```
File  BUILD  HELP
[Icons]
ADD SCURT Y
main()
OPEN
DATAOFINT scurt.5
INT i=0
FROM i=0 TO i<5 DO i++
OPEN
GRAB("%nENTER THE VALUE: ",scurt.i)
CLOSE

SHOW("%nYOUR ENTERED VALUES :")
FROM i=0 TO i<5 DO i++
SHOW("%",scurt.i)

CLOSE
```

AND THE OUTPUT



```
C:\windows\system32\cmd.exe
ENTER THE VALUE: 5
ENTER THE VALUE: 4
ENTER THE VALUE: 7
ENTER THE VALUE: 22
ENTER THE VALUE: 4
YOUR ENTERED VALUES : 5 4 7 22 4
Press any key to continue . . .
```

Although grabbing and showing with the help of loops gives us more flexibility but we also have some pre defined functions to do this task.

THE GRABDATA FUNCTION

Grabbing data from the user is more easy with this function present in scurtY language. To grab data from zero upto a limit of elements you just have to write :

GRABDATA(*variable_name.size*)

it will automatically grab data from user without writting a loop for it.

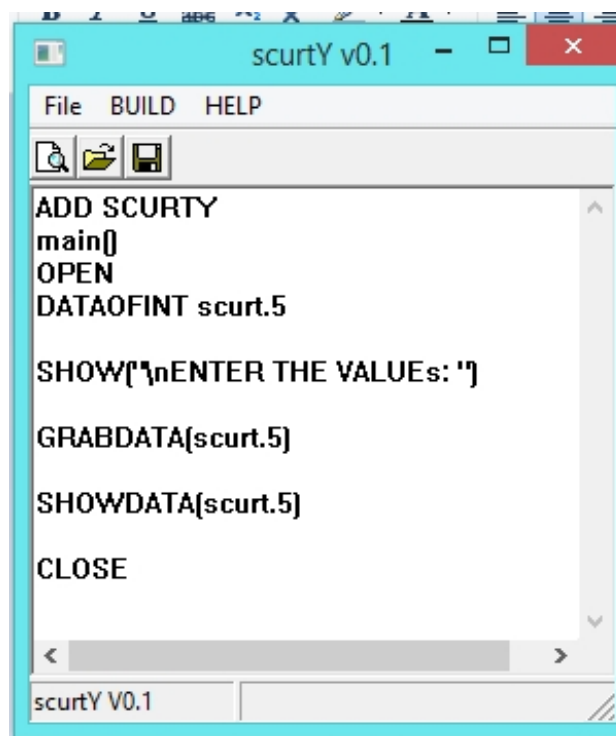
THE SHOWDATA FUNCTION

Showing data to the user is more easy with this function present in scurtY language. To show data from zero upto a limit of elements you just have to write :

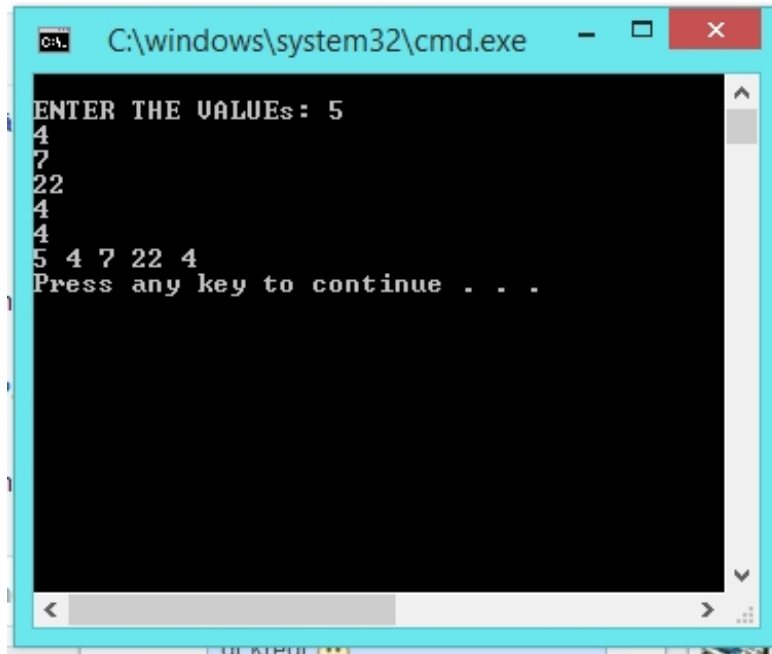
SHOWDATA(*variable_name.size*)

it will automatically Show data to user without writing a loop for it.

now let us see how easy the last software will be:



which produces the output which is same but don't have messages.



```
C:\windows\system32\cmd.exe

ENTER THE VALUEs: 5
4
7
22
4
4
5 4 7 22 4
Press any key to continue . . .
```

now your task is to create a software which stores the data of 10 students their marks in CS, their attendance, thier roll no and print the details related to that roll no.

BEST OF LUCK HAPPY CODDING



THANKS FOR READING



