MODULE 4: CONDITIONAL STATEMENTS, ARRAYS AND LOOPS

ARRAY

* A homogeneous collection of data stored contiguously in memory and referenced under one identifier.

Declaraction of an Array

Data\_type array\_identifier[];

Examples:

int num[1000];

float size[MAX]; /\* MAX is a defined constant \*/

char letters[] = {‘a’, ‘b’, ‘c’}; /\* initializing the array with characters a, b and c and automatically, the number of elements in the array is 3 \*/

Indexing and accessing array elements

Int array[] = {7,4,3,8,400};

Index 0 1 2 3 4

array 7 4 3 8 400

array[0] contains value 7.

array[1] contains value 4.

array[2] contains value 3.

array[3] contains value 8.

array[4] contains value 400.

array has 5 elements.

Index starts at 0, thus the last element in the array is array[4]

Elements can be used like any other variable

Examples:

/\*get the third value from array and store it to x\*/

int x = array[2];

/\* change the value of an element \*/

array[3] = 7;

/\* must have single quotation if it is a character \*/

char c[20];

c[17] = ‘z’;

CONDITIONAL STATEMENTS

Go through MODULE\_4\_1.pdf

* The PDFs belong to the CpE Department. They were used during our CpE315 Class.

== equal

>= greater than or equal to

<= lesser than or equal to

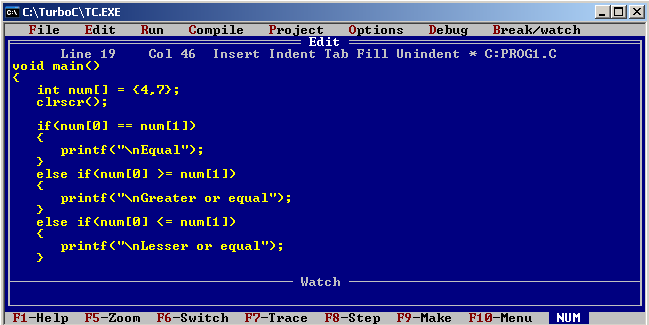
> greater than

< less than

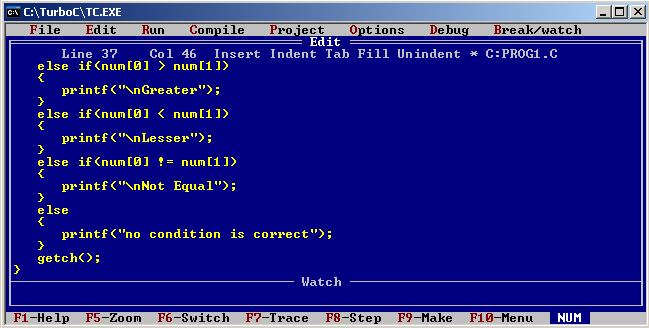
! not

!= not equal to

Consider the following program:



Continued…



Watch the difference if you change “else if” with just an “if”.

Note the number of outputs.

An “if” starts the possible series of comparison with another “if”, “else if” or “else”

An “if” can exist alone, but not an “else if” and an “else” which requires an “if”.

Multiple “if” can be executed. They are executed simultaneously by the naked eye but in reality, the first if executed first.

SWITCH statement

Discrete values are compared. Therefore when int, case 2:, case 3:, case4:, and when char, case ‘a’:, case’Z’:, etc..

Examples:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int key;

key = getche();

switch(key)

{

case 13:

/\* to do code here \*/

break;

default:

break;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

The key word break behaves in a way that the program instantaneously goes out of a loop or a switch.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char key;

key = getche();

switch(key)

{

case ‘a’:

case ‘A’:

/\* to do code here \*/

break;

default:

break;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Notice that case ‘a’ does not have a break statement. It means that it will continue to read down until it finds a break.

What this code do is that whenever a user input ‘a’ or ‘A’, something will happen depending on the content of the /\*to do code here\*/

Switch is usually used in menus. The switch is inside a do-while loop in this case.

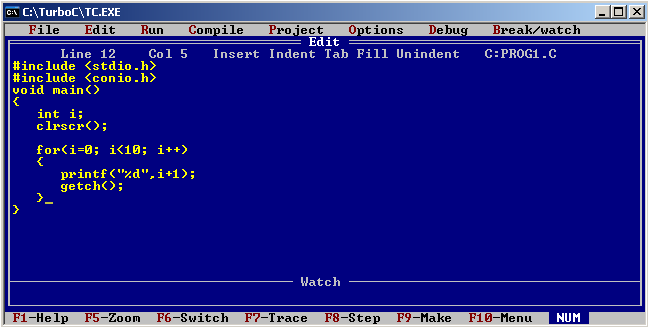
LOOP:

Discussions are on the MODULE\_4\_2.pdf

* The PDFs belong to the CpE Department. They were used during our CpE315 Class.

Examples:

“for” loop



Sequence:

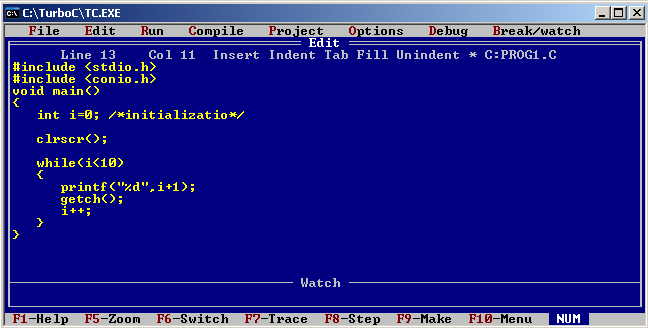
1 – initialization (i=0)

2 – condition (i<10)

3 – body (printf……. getch();)

4 – update (i++), then goes back to 2

“while” loop

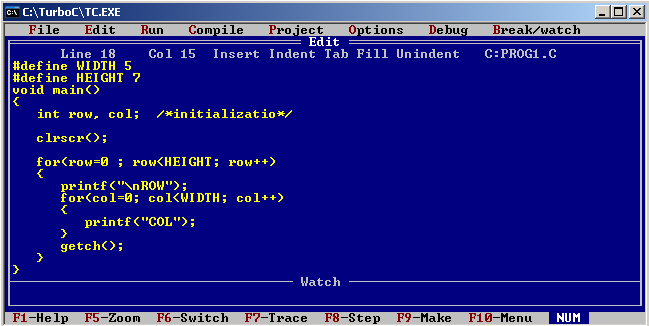


Update is inside the block.

A block of code happens when a code is within ‘{‘ and ‘}’

Do-while is just another version of this only, the conditional statement is after the block.

Nested loop:



This is a Nested For Loop

INFINITE LOOP

An infinite loop happens when a function conditional statement is always true.

for(;;);

while(1);

to terminate this during run time, CTRL + BREAK then ESC