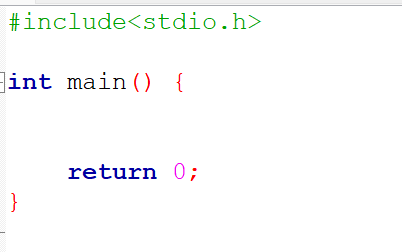
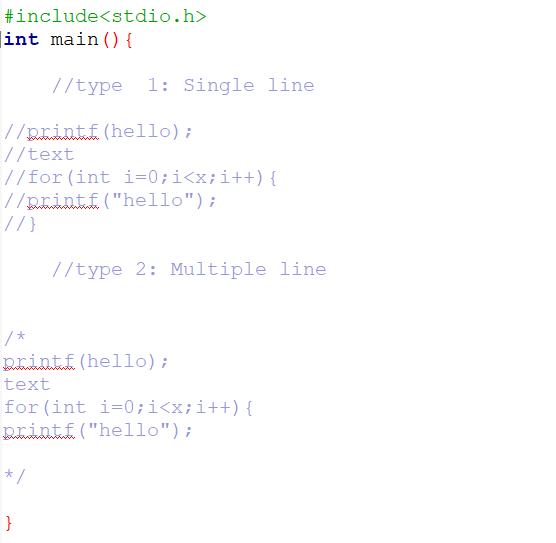
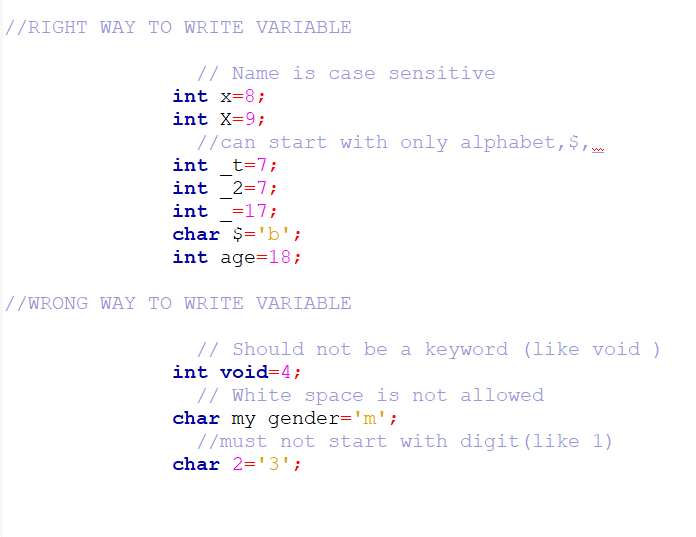
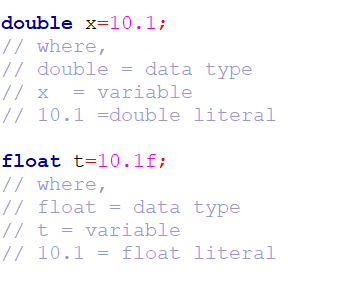
Format Comments

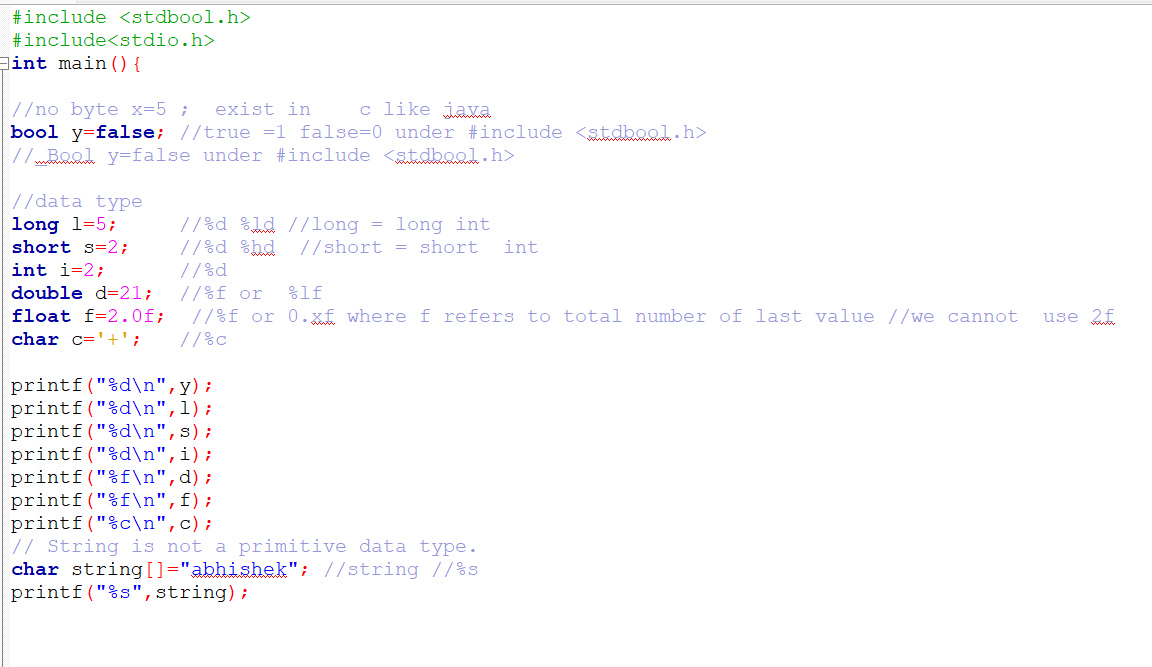
Data Types and variable



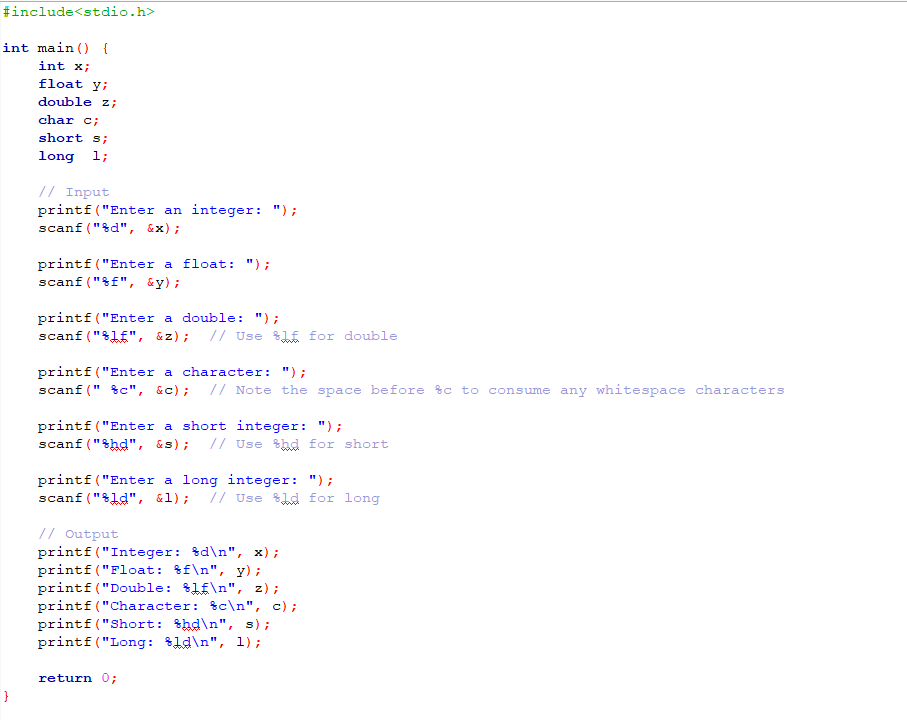


 initializing multiple similar datatype variable

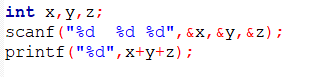
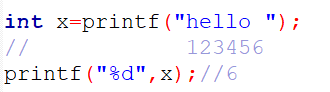
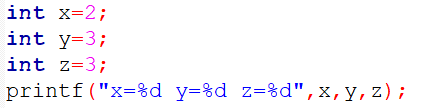
Primitive Data Type



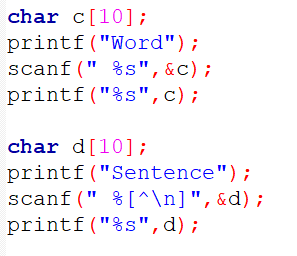
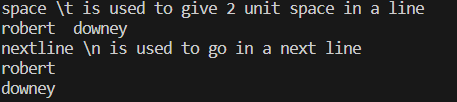
Scanner Class



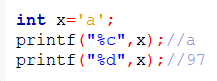
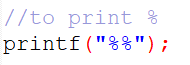
Multiple scan in one time Multiple printing in one time concept count in printf

Scanning word vs sentence

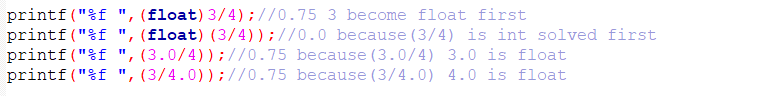
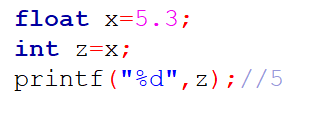
Wrong format to initialize to print % in sentence ascii value

Data Type Conversion /widening/implicit conversion

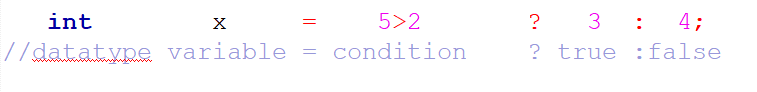
short🡪int🡪 long🡪 float🡪double

Type Casting / Narrowing /Explicit Conversion



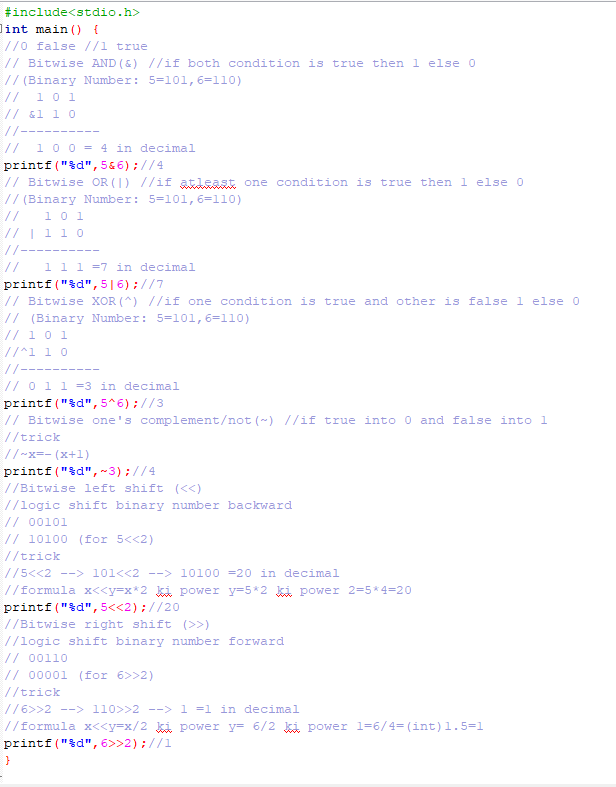
Operators

1. Arithmetic
2. Binary (mathematical) operator: + - \* / %
3. Unary operator: ++x x++ --x x—
4. Ternary operator: ?:



1. Relational ( == != > >= < <= )
2. Logical ( && || ! )
3. Assignment ( = += -= \*= %= /=)
4. Bitwise (
5. &(bitwise and)
6. |(bitwise or)
7. <<(shift left)
8. >>(shift right)
9. ~(one’s complement)
10. ^(bitwise exclusive or) )

Bitwise Operator in detail



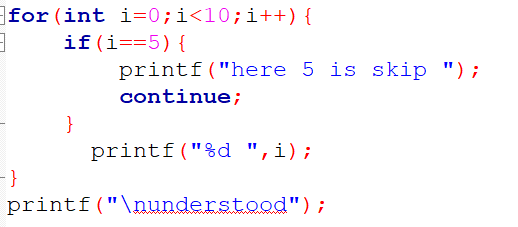
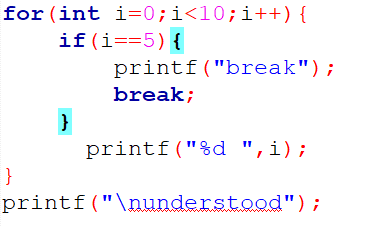






Break and Continue Statement

Break(to exit loop) Continue(to skip specific condition iteration)



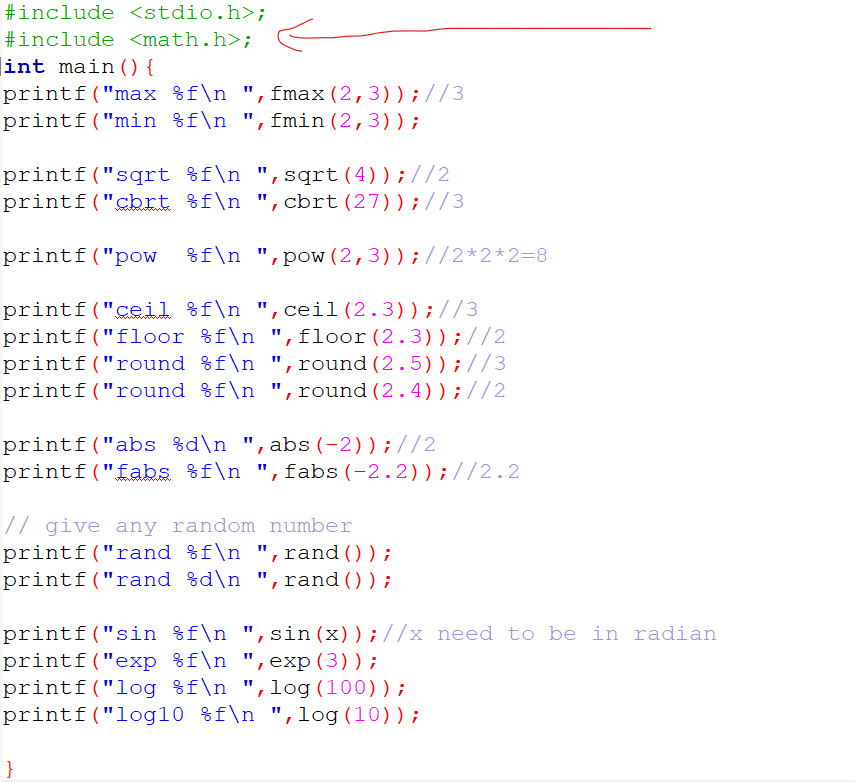
***Output: break***



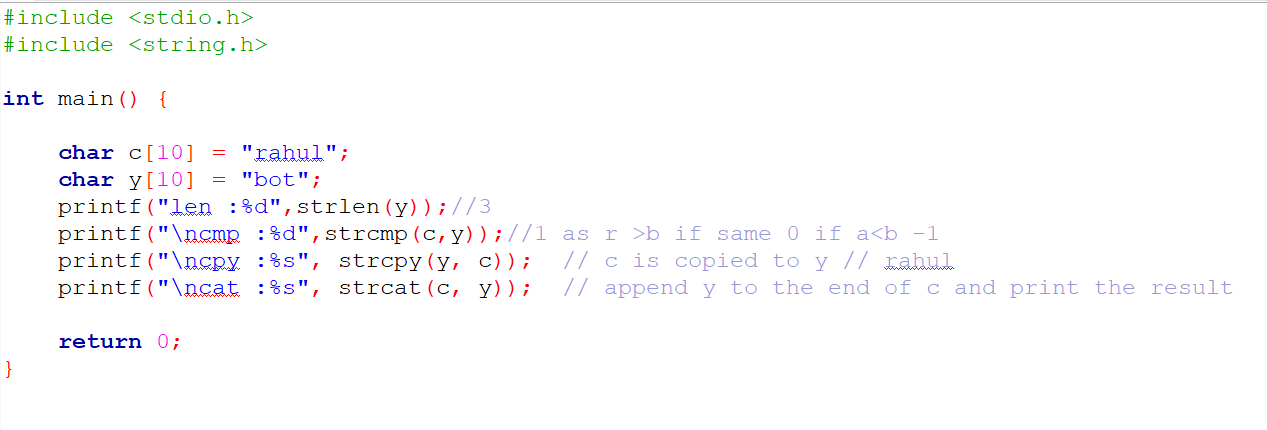
***Output: continue***

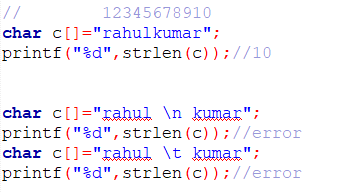
******

Math function



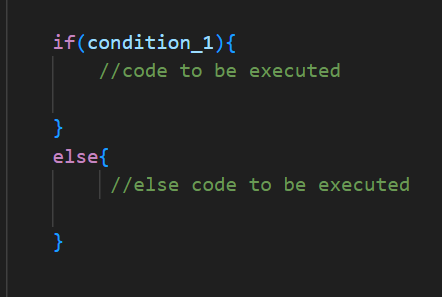
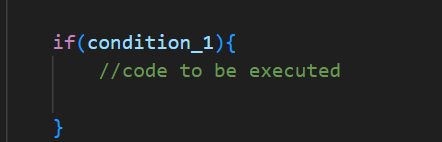
String Function



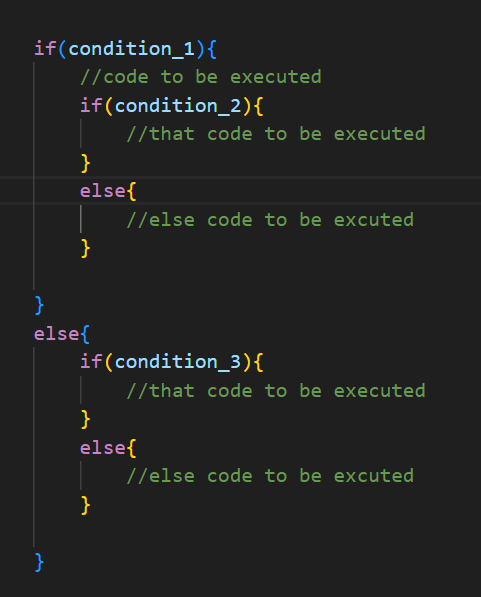
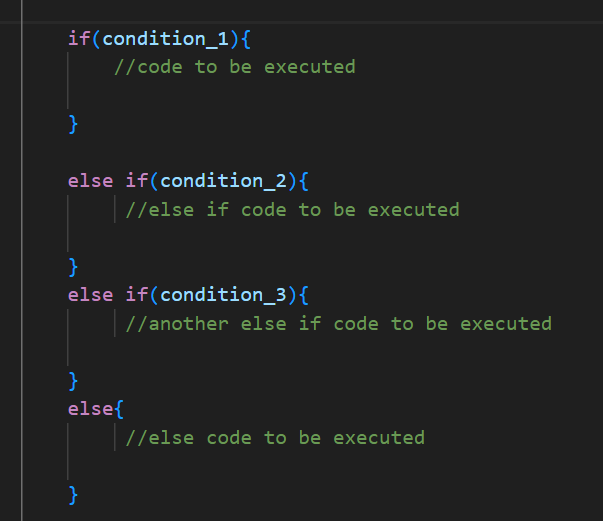


Conditional Branching/Selectional Control/Decision Making

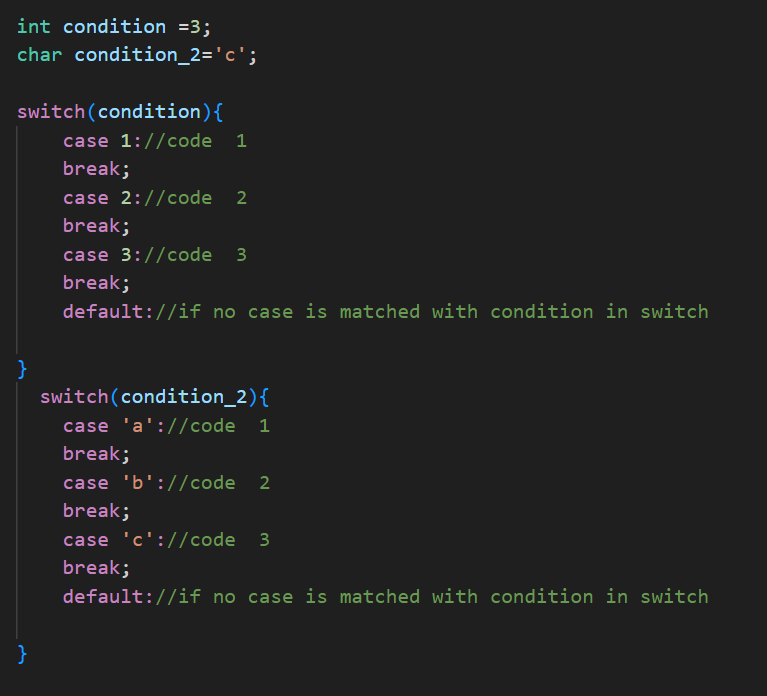
If Statement if else Statement



Else if Statement Nested if Statement

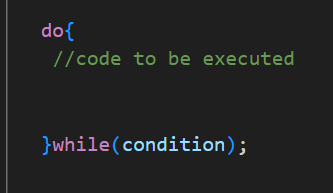


Switch statement

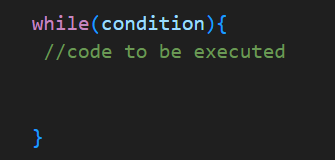
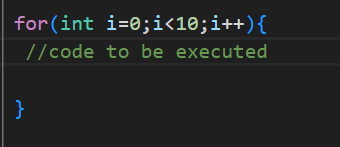


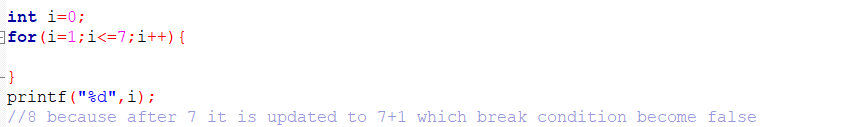
Loop Statement

1)Exit Controlled Loop/post tested loop (do while loop)



2)Entry Controlled Loop/pre tested loop (for loop, while loop)



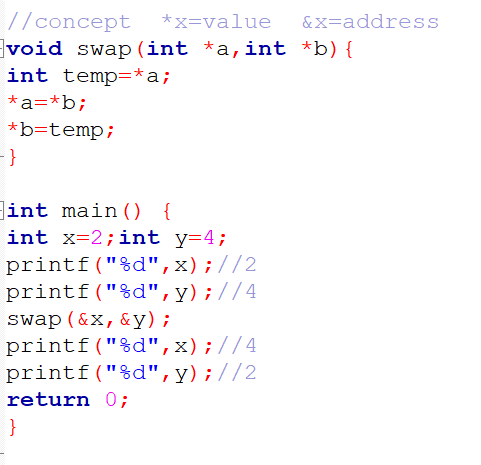
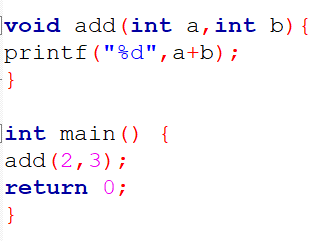


Function

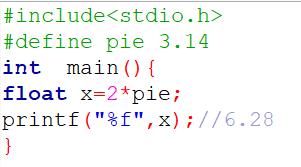
* function overloading (same name different parameter or same name but parameter datatype different) is not supported in c

Here value is passed in function by call by value and call by reference.

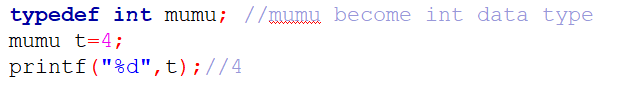
# Call by value Call by reference



#Define



typedef



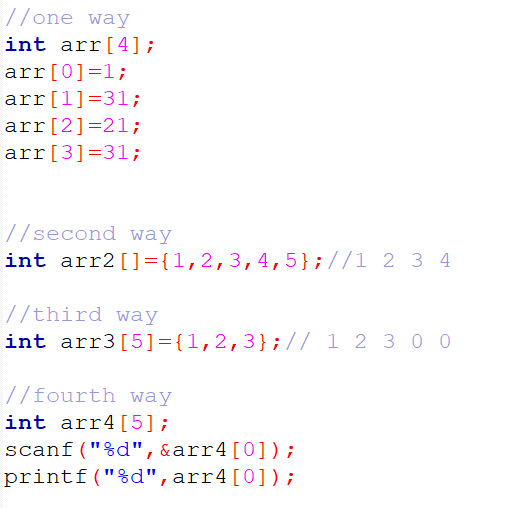
Array

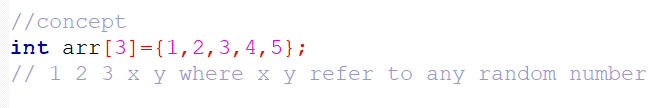
*For understanding only*

1. *Int dim[row]*
2. *Int dim2[row][column]*
3. *Int dim3[depth][row][column] (generally not used but can be)*

* *Row=sizeof(dim)/sizeof(dim[0])*
* *Column= sizeof(dim[0])/sizeof(dim[0][0])*

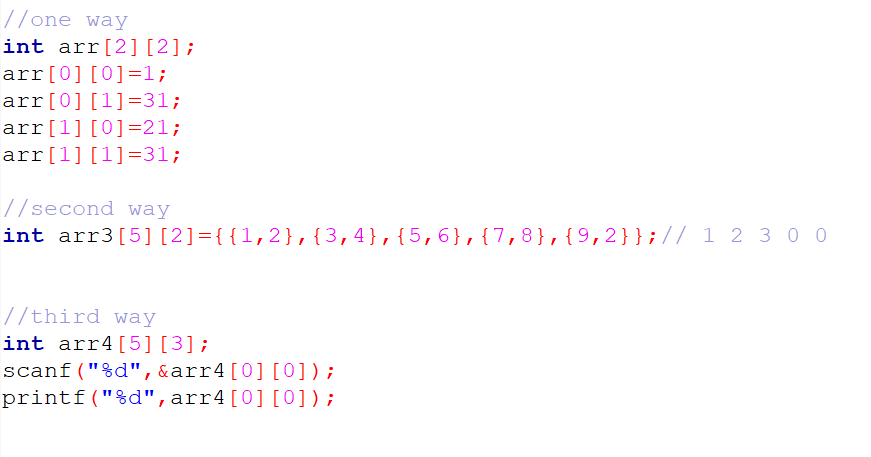
One dimensional Array





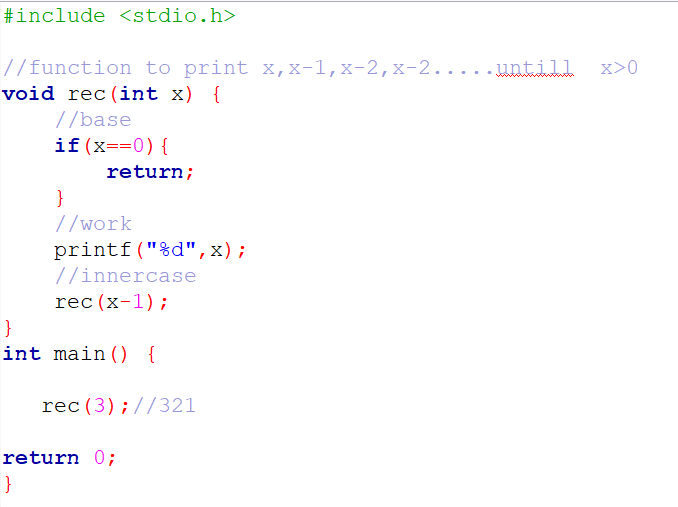
Multidimensional Array

2d and 3d array and many more



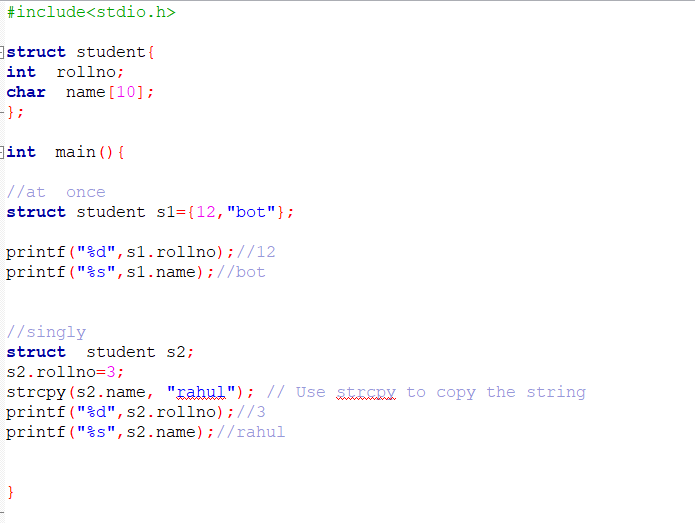
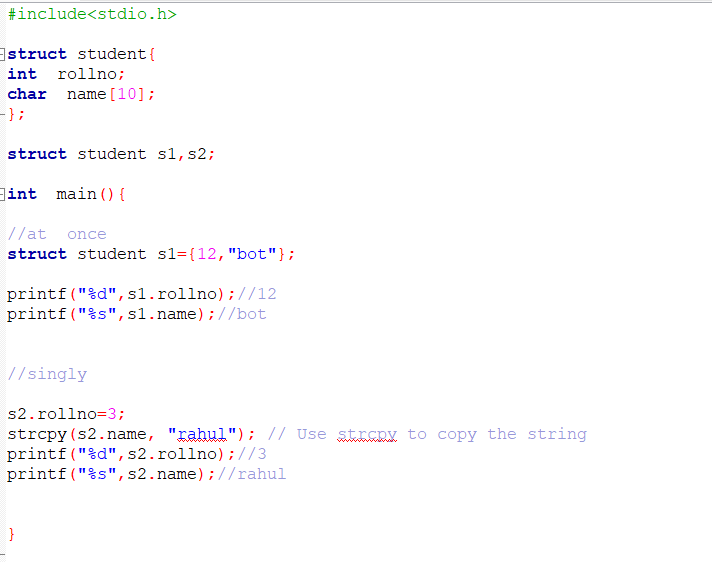
Recursion

1. Base case
2. Work
3. Inner case



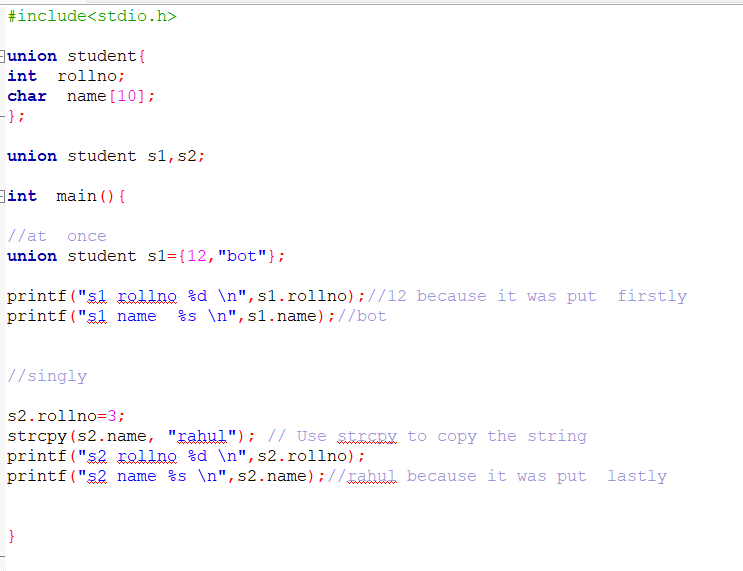
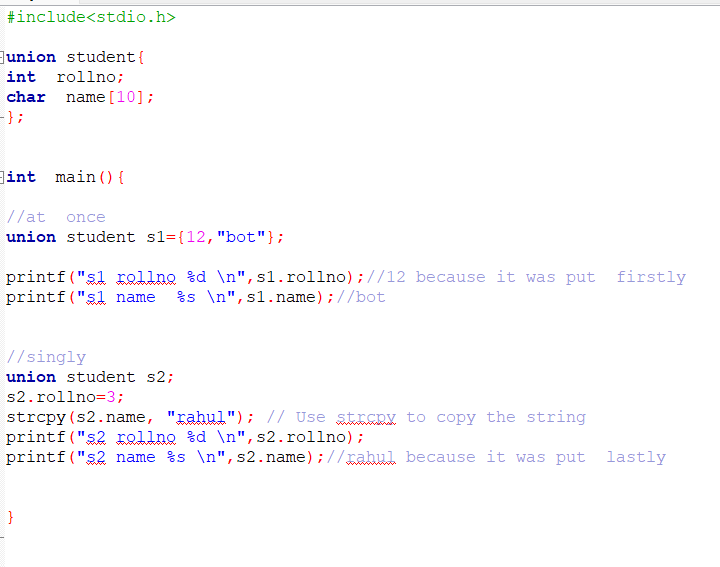
Structure

# In main Globally

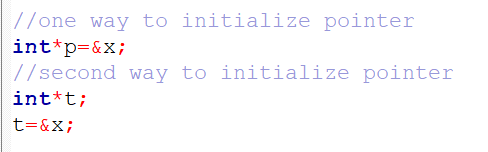
Union

# In main Globally

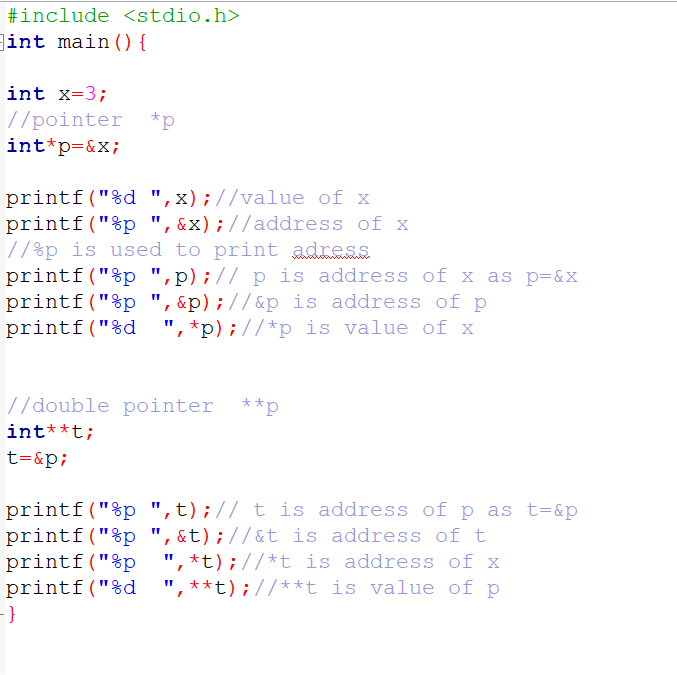


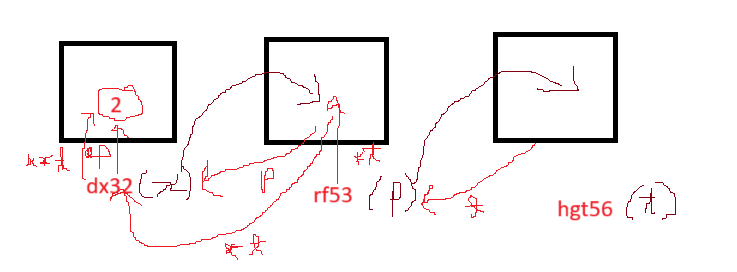
Pointer

# Declaration



# Type of pointer : Pointer and Double Pointer



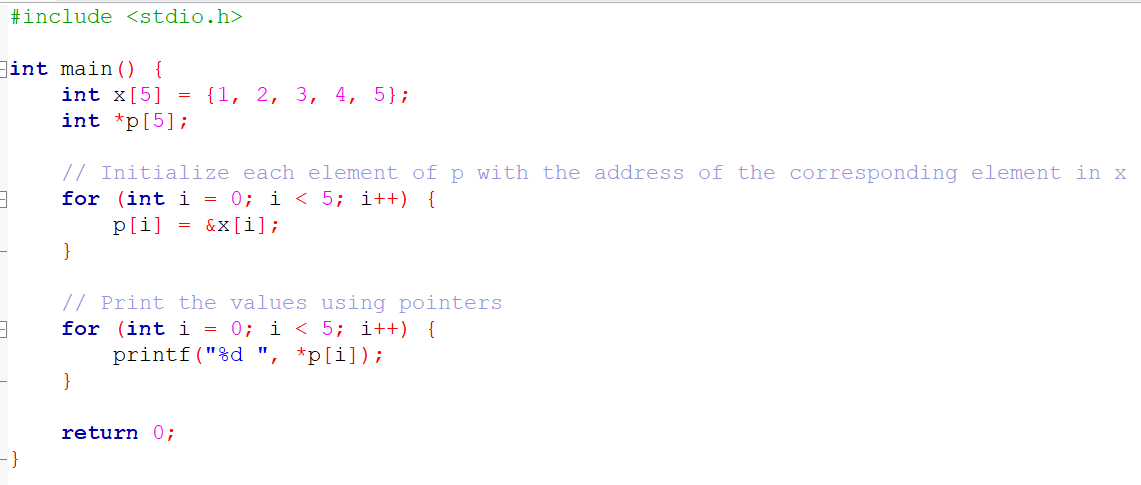


\*p mean value of &x(address x)

\*t mean value of &p(address p)

\*\*t mean value of &x(address x)

Pointer to array



Pending topic

1. Pointer to function
2. Pointer to structure
3. Memory allocation
4. linked structure

5. file handling