**# and ## Operators in C**

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 03 Dec, 2019

**Stringizing operator (#)**

This operator causes the corresponding actual argument to be enclosed in double quotation marks. The # operator, which is generally called the*stringize* operator, turns the argument it precedes into a quoted string. For more on pre-processor directives – refer [this](https://www.geeksforgeeks.org/cc-preprocessors/)  
**Examples :**

1. The following preprocessor turns the line printf(mkstr(geeksforgeeks)); into printf(“geeksforgeeks”);

|  |
| --- |
| // CPP program to illustrate (#) operator  #include <stdio.h>  #define mkstr(s) #s  int main(void)  {      printf(mkstr(geeksforgeeks));      return 0;  } |

1. Output:
2. geeksforgeeks
3. In this program, value of a is replaced by macro.

|  |
| --- |
| // CPP program to illustrate (#) operator  #include <iostream>  using namespace std;    #define a 8.3297    int main()  {      cout << "Value of a is " << a << endl;        return 0;  } |

1. Output:
2. Value of a is 8.3297
3. This program finds out maximum out of two numbers using macro

|  |
| --- |
| // CPP program to illustrate (#) operator  #include <iostream>  using namespace std;    #define MAX(i, j) (((i) > (j)) ? i : j)    int main()  {      int a, b;        a = 250;      b = 25;        cout << "The maximum is " << MAX(a, b) << endl;        return 0;  } |

1. Output:
3. The maximum is 250

**Token-pasting operator (##)**

Allows tokens used as actual arguments to be concatenated to form other tokens. It is often useful to merge two tokens into one while expanding macros. This is called token pasting or token concatenation. The ‘##’ pre-processing operator performs token pasting. When a macro is expanded, the two tokens on either side of each ‘##’ operator are combined into a single token, which then replaces the ‘##’ and the two original tokens in the macro expansion.  
**Examples :**

1. The preprocessor transforms printf(“%d”, concat(x, y)); into printf(“%d”, xy);

|  |
| --- |
| // CPP program to illustrate (##) operator  #include <stdio.h>  #define concat(a, b) a##b  int main(void)  {      int xy = 30;      printf("%d", concat(x, y));      return 0;  } |

1. Output:
2. 30

**Application:** The ## provides a way to concatenate actual arguments during macro expansion. If a parameter in the replacement text is adjacent to a ##, the parameter is replaced by the actual argument, the ## and surrounding white space are removed, and the result is re-scanned.