

cat

\$

cat

```
$ cat
```

cat

```
$ cat
```

```
# print contents of a file
```

cat

```
$ cat abc.txt ↵ # print contents of a file
```

cat

```
$ cat abc.txt ↵ # print contents of a file
The quick
Brown fox jumps
oVER
the
lazy DOG.
$
```

cat

```
$ ./a.out abc.txt ↵ # print contents of a file
The quick
Brown fox jumps
oVER
the
lazy DOG.
$
```

cat

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* open file to read */  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* fopen() */  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* fopen(file name) */  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* fopen(file name, mode) */  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

cat

```

/* mycat.c */

#include <stdio.h>

int main(void)
{

    /* fopen(file name, mode) */
    fopen(

    )

    return 0;

}

```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* fopen(file name, mode) */  
    fopen("abc.txt"      )  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* fopen(file name, mode) */  
    fopen("abc.txt", "r")  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* return type? */  
    fopen("abc.txt", "r")  
  
    return 0;  
}
```


cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
  
    /* return type - file pointer */  
    fopen("abc.txt", "r")  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fopen("abc.txt", "r")  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
    /* read characters till end-of-file */  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
    /* read characters till end-of-file(EOF) */  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
        fgetc( )  
  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
  
int main(void)  
{  
    FILE *fp;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
    fgetc(fp)  
  
    return 0;  
}
```

cat

```
/* mycat.c */
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int c;

    /* return type - file pointer */
    fp = fopen("abc.txt", "r");

    /* read characters till end-of-file(EOF) */
        c = fgetc(fp)

    return 0;
}
```


cat

```
/* mycat.c */  
#include <stdio.h>  
  
int main(void)  
{  
    FILE *fp;  
    int c;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
    while( c = fgetc(fp)          )  
  
        return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
  
int main(void)  
{  
    FILE *fp;  
    int c;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
    while((c = fgetc(fp)) != EOF)  
  
        return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
    int c;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
  
    return 0;  
}
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
    int c;  
  
    /* return type - file pointer */  
    fp = fopen("abc.txt", "r");  
  
    /* read characters till end-of-file(EOF) */  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
  
    /* close file */  
  
    return 0;  
}
```

cat

```
/* mycat.c */
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int c;

    /* return type - file pointer */
    fp = fopen("abc.txt", "r");

    /* read characters till end-of-file(EOF) */
    while((c = fgetc(fp)) != EOF)
        putchar(c);

    /* close file */
    fclose( );

    return 0;
}
```

cat

```
/* mycat.c */
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int c;

    /* return type - file pointer */
    fp = fopen("abc.txt", "r");

    /* read characters till end-of-file(EOF) */
    while((c = fgetc(fp)) != EOF)
        putchar(c);

    /* close file */
    fclose(fp);

    return 0;
}
```

cat

\$

cat

```
$ cc mycat.c ↵
```


cat

```
$ cc mycat.c ↵  
$
```

cat

```
$ cc mycat.c ↵
```

```
$ ./a.out ↵
```

cat

```
$ cc mycat.c ↵
```

```
$ ./a.out ↵
```

```
The quick
```

```
Brown fox jumps
```

```
oVER
```

```
the
```

```
lazy DOG.
```

```
$
```

cat

```
$ cc mycat.c ↵  
$ ./a.out abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$
```

cat

```
$ cc mycat.c ↵  
$ ./a.out data.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$
```

cat

```
/* mycat.c */  
#include <stdio.h>  
int main(void)  
{  
    FILE *fp;  
    int c;  
    fp = fopen("abc.txt", "r");  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
    fclose(fp);  
    return 0;  
}
```

cat

```
/* mycat.c with command line arguments */  
#include <stdio.h>  
  
int main(void)  
{  
    FILE *fp;  
    int c;  
  
    fp = fopen("abc.txt", "r");  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
  
    fclose(fp);  
  
    return 0;  
}
```

cat

```
/* mycat.c with commmand line arguments */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int c;  
  
    fp = fopen("abc.txt", "r");  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
  
    fclose(fp);  
  
    return 0;  
}
```


cat

```
/* mycat.c with commmand line arguments */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int c;  
  
    fp = fopen(argv[1], "r");  
    while((c = fgetc(fp)) != EOF)  
        putchar(c);  
  
    fclose(fp);  
  
    return 0;  
}
```

cat

\$

cat

```
$ cc mycat.c ↵
```

cat

```
$ cc mycat.c ↵  
$
```

cat

```
$ cc mycat.c ↵  
$ ./a.out abc.txt ↵
```

cat

```
$ cc mycat.c ↵  
$ ./a.out abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$
```

cat

```
$ cc mycat.c ↵  
$ ./a.out abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$ ./a.out mycat.c ↵
```

WC

\$


```
$ cat abc.txt ↵
```

```
$ cat abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$
```

```
$ cat abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$ wc
```

```
$ cat abc.txt ↵
```

```
The quick
```

```
Brown fox jumps
```

```
oVER
```

```
the
```

```
lazy DOG.
```

```
$ wc
```

```
# word count
```

```
$ cat abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$ wc abc.txt ↵ # word count
```

```
$ cat abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$ wc abc.txt ↵ # word count  
    5    9   45 abc.txt  
$
```

```
$ cat abc.txt ↵  
The quick  
Brown fox jumps  
oVER  
the  
lazy DOG.  
$ wc abc.txt ↵ # word count  
    5    9   45 abc.txt  # words lines chars  
$
```

WC

```
/* mywc.c - count characters */
```


WC

```
/* mywc.c - count characters */
#include <stdio.h>

int main(int argc, char *argv[])
{

    return 0;
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
int main(int argc, char *argv[])  
{  
  
    /* open file */  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
    /* read character by character */  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    fgetc(fp)  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    while(fgetc(fp) != EOF)  
  
        return 0;  
}
```

```
/* mywc.c - count characters */
#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *fp;

    /* open file */
    fp = fopen(argv[1], "r");

    /* read character by character */
    while(fgetc(fp) != EOF)
        nc++;

    return 0;
}
```



```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int nc = 0;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    while(fgetc(fp) != EOF)  
        nc++;  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int nc = 0;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    while(fgetc(fp) != EOF)  
        nc++;  
  
    printf("%d\n", nc);  
  
    return 0;  
}
```

```
/* mywc.c - count characters */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int nc = 0;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    while(fgetc(fp) != EOF)  
        nc++;  
  
    printf("%d\n", nc);  
  
    fclose(fp);  
  
    return 0;  
}
```

```
/* mywc.c - count lines */
```

WC

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    fp = fopen(argv[1], "r");
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    fp = fopen(argv[1], "r");
```

```
    /* read character by character */
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *fp;

    fp = fopen(argv[1], "r");

    /* read character by character */
    while ((c = getc(fp)) != EOF)

    return 0;
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c;
```

```
    fp = fopen(argv[1], "r");
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        return 0;
```

```
}
```



```
/* mywc.c - count lines */
#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *fp;
    int c;
    fp = fopen(argv[1], "r");

    /* read character by character */
    while ((c = getc(fp)) != EOF)
        if (c == '\n')

    return 0;
}
```

```
/* mywc.c - count lines */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    int c;  
    fp = fopen(argv[1], "r");  
  
    /* read character by character */  
    while ((c = getc(fp)) != EOF)  
        if (c == '\\n')  
            nl++;  
  
    return 0;  
}
```

```
/* mywc.c - count lines */
#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *fp;
    int c, nl = 0;
    fp = fopen(argv[1], "r");

    /* read character by character */
    while ((c = getc(fp)) != EOF)
        if (c == '\n')
            nl++;

    return 0;
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen(argv[1], "r");
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen(argv[1], "r");
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])  
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen("abc.txt", "r");
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen("abc.txt", "r");
```

```
    /* what if the file does not exist? */
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen("abc.txt", "r");
```

```
    /* what if the file does not exist? */
```

```
    if (fp == NULL) {
```

```
    }
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```



```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen("abc.txt", "r");
```

```
    /* what if the file does not exist? */
```

```
    if (fp == NULL) {
```

```
        printf("Cannot open file.\n");
```

```
    }
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
```

```
#include <stdio.h>
```

WC

```
int main(int argc, char *argv[])
```

```
{
```

```
    FILE *fp;
```

```
    int c, nl = 0;
```

```
    fp = fopen("abc.txt", "r");
```

```
    /* what if the file does not exist? */
```

```
    if (fp == NULL) {
```

```
        printf("Cannot open file.\n");
```

```
        exit(1);
```

```
    }
```

```
    /* read character by character */
```

```
    while ((c = getc(fp)) != EOF)
```

```
        if (c == '\n')
```

```
            nl++;
```

```
    printf("%d\n", nl);
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count lines */
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    FILE *fp;
    int c, nl = 0;
    fp = fopen("abc.txt", "r");
    /* what if the file does not exist? */
    if (fp == NULL) {
        printf("Cannot open file.\n");
        exit(1);
    }
    /* read character by character */
    while ((c = getc(fp)) != EOF)
        if (c == '\n')
            nl++;
    printf("%d\n", nl);
    return 0;
}
```

```
/* mywc.c - count characters - version 2 */
```

WC

```
/* mywc.c - count characters - version 2 */
```

WC

```
#include <stdio.h>
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    return 0;
```

```
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
    /* go to the end of the file */  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* go to the end of the file */  
    fseek(fp, 0L, SEEK_END);  
  
    return 0;  
}
```



```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : */  
    fseek(fp, 0L, SEEK_END);  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end */  
    fseek(fp, 0L, SEEK_END);  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, OL, SEEK_END);  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
  
  
  
  
  
  
  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
    ftell(fp)  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
    nc = ftell(fp);  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    long nc;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
    nc = ftell(fp);  
  
  
    return 0;  
}
```

```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    long nc;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
    nc = ftell(fp);  
  
    printf("%d\n", nc);  
  
    return 0;  
}
```



```
/* mywc.c - count characters - version 2 */  
#include <stdio.h>  
  
int main(int argc, char *argv[])  
{  
    FILE *fp;  
    long nc;  
  
    /* open file */  
    fp = fopen(argv[1], "r");  
  
    /* fseek : from the end go zero bytes */  
    fseek(fp, 0L, SEEK_END);  
  
    /* current position */  
    nc = ftell(fp);  
  
    printf("%d\n", nc);  
  
    fclose(fp);  
  
    return 0;  
}
```