Add a line at the end of file

```
/* add_at_end.c */
```

Add a line at the end of file

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

int main(void)
{









    return 0;
}
```

Add a line at the end of file

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

int main(void)
{

    /* open to append text */




    return 0;
}
```

Add a line at the end of file

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

int main(void)
{

    /* open to append text */
    fp = fopen("abc.txt", "a");



    return 0;
}
```

Add a line at the end of file

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

int main(void)
{
    FILE *fp;

    /* open to append text */
    fp = fopen("abc.txt", "a");



    return 0;
}
```

Add a line at the end of file

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

int main(void)
{
    FILE *fp;

    /* open to append text */
    fp = fopen("abc.txt", "a");

    fprintf(fp, "hello, world!\n");


    return 0;
}
```

Add a line at the end of file

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

int main(void)
{
    FILE *fp;

    /* open to append text */
    fp = fopen("abc.txt", "a");

    fprintf(fp, "hello, world!\n");

    fclose(fp);

    return 0;
}
```

Add a line at the end of file

```
$
```

Add a line at the end of file

```
$ cat abc.txt ↵
```

Add a line at the end of file

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

Add a line at the end of file

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

Add a line at the end of file

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

Add a line at the end of file

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

Add a line at the end of file

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

Add a line at the end of file

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

Add a line at the end of file

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cc add_at_end.c ↵
$ ./a.out ↵
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
hello, world!
$
```

Modes

mode	**Description**
`"r"`	It opens a file for reading. The file must exist.
`"w"`	It creates an empty file for writing. If a file with the same name already exists, its content is erased and the file is considered as a new empty file.
`"a"`	It appends to a file. Writing operations, append data at the end of the file. The file is created if it does not exist.

Add a line at the end of file – 2^{nd} version

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    fp = fopen("abc.txt", ?);




    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", ?);




    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");




    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");


    fseek(fp, 0L, SEEK_END);


    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2^{nd} version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");

    /* fseek :  */
    fseek(fp, 0L, SEEK_END);


    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2^{nd} version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");

    /* fseek :  from the end */
    fseek(fp, 0L, SEEK_END);


    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");

    /* fseek :  from the end go zero bytes */
    fseek(fp, 0L, SEEK_END);


    fclose(fp);

    return 0;
}
```

Add a line at the end of file – 2nd version

```
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open to read and write */
    fp = fopen("abc.txt", "r+");

    /* fseek :  from the end go zero bytes */
    fseek(fp, 0L, SEEK_END);

    fprintf(fp, "world!\n");

    fclose(fp);

    return 0;
}
```

mode	**Description**
`"r+"`	It opens a file for reading and writing. The file must exist.

"r+"

```
/* convert all uppercase letters to lowercase */
```

"r+"

```
/* convert all uppercase letters to lowercase */
#include <stdio.h>
                                                    "r+"

int main(void)
{









    return 0;
}
```

```
/* convert all uppercase letters to lowercase */
#include <stdio.h>
                                                    "r+"

int main(void)
{


    /* open file to read and write */









    return 0;
}
```

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>

int main(void)
{

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    return 0;
}
```

"r+"

```
/* convert all uppercase letters to lowercase */
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */

    fclose(fp);

    return 0;
}
```

"r+"

```
/* convert all uppercase letters to lowercase */
#include <stdio.h>
                                                    "r+"

int main(void)
{
    FILE *fp;

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
              fgetc(fp)




    fclose(fp);

    return 0;
}
```

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
          c = fgetc(fp)

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
          c = fgetc(fp)

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
                                                    "r+"

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {




    }

    fclose(fp);

    return 0;
}
```

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
                                                    "r+"

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
            tolower(c)



    }

    fclose(fp);

    return 0;
}
```

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
            tolower(c)



    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);



    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* overwrite the character read with new c */


    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* go back one position */


    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* go back one position */
        fseek(  ,    ,         );


    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* in the current file */
        fseek(fp,    ,         );


    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* from the current position */
        fseek(fp,    , SEEK_CUR);

    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* go back one character */
        fseek(fp, -1L, SEEK_CUR);

    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* go back one character */
        fseek(fp, -1L, SEEK_CUR);
                    /* put the character */
    }

    fclose(fp);

    return 0;
}
```

"r+"

```c
/* convert all uppercase letters to lowercase */
#include <stdio.h>
#include <ctype.h>

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

    /* open file to read and write */
    fp = fopen("abc.txt", "r+");

    /* read char at a time, convert to lowercase */
    while((c = fgetc(fp)) != EOF) {
        c = tolower(c);
        /* go back one character */
        fseek(fp, -1L, SEEK_CUR);
        fputc(c, fp);/* put the character */
    }

    fclose(fp);

    return 0;
}
```

"r+"

"a"

```
/* to file abc.txt, add the contents of def.txt */
                                                    "a"
```

```
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{



    /* open def.txt to read */









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp2;




    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp2;


    /* open file abc.txt to add at the end */


    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp2;


    /* open file abc.txt to append */


    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;


    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;


    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2,  */









    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2,  */
    while ((c = fgetc(fp2)) != EOF)




    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)




    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);



    return 0;
}
```

```c
/* to file abc.txt, add the contents of def.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);

    fclose(fp1);
    fclose(fp2);

    return 0;
}
```

"a"

```
$
```

"a"

```
$ cat abc.txt ↵
```

"a"

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

"a"

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

"a"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cat def.txt ↵
hello, world!
$
```

"a"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cat def.txt ↵
hello, world!
$ ./a.out ↵
$
```

"a"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cat def.txt ↵
hello, world!
$ ./a.out ↵
$ cat abc.txt ↵
```

"a"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cat def.txt ↵
hello, world!
$ ./a.out ↵
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
hello, world!
$
```

"a"

```
$ ./a.out ↵
$
```

"a"

```
$ ./a.out ↵
$ cat abc.txt ↵
```

"a"

```
$ ./a.out ↵
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
hello, world!
hello, world!
$
```

```c
/* append def.txt to abc.txt */

#include <stdio.h>                                  "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);

    fclose(fp1);
    fclose(fp2);

    return 0;
}
```

```
/* append def.txt to abc.txt; print abc.txt */

#include <stdio.h>                                    "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);

    fclose(fp1);
    fclose(fp2);

    return 0;
}
```

```c
/* append def.txt to abc.txt; print abc.txt */

#include <stdio.h>                                    "a"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);

    fclose(fp1);
    fclose(fp2);

    return 0;
}
```

```c
/* append def.txt to abc.txt; print abc.txt */

#include <stdio.h>                                  "a+"

int main(void)
{
    FILE *fp1, *fp2;
    int c;

    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a+");

    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");

    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);

    fclose(fp1);
    fclose(fp2);

    return 0;
}
```

Modes

mode	**Description**
`"r+"`	It opens a file for reading and writing. The file must exist.
`"a+"`	It opens a file for reading and appending.

```c
/* append def.txt to abc.txt; print abc.txt */
#include <stdio.h>
int main(void)
{
    FILE *fp1, *fp2;
    int c;
    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a+");
    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");
    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

"a+"

```c
/* append def.txt to abc.txt; print abc.txt */
#include <stdio.h>                                    "a+"
int main(void)
{
    FILE *fp1, *fp2;
    int c;
    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a+");
    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");
    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);
    rewind(fp1);
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

```c
/* append def.txt to abc.txt; print abc.txt */
#include <stdio.h>
int main(void)
{
    FILE *fp1, *fp2;
    int c;
    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a+");
    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");
    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);
    rewind(fp1);
    while((c = fgetc(fp1)) != EOF)
        putchar(c);
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

"a+"

```c
/* append def.txt to abc.txt; print abc.txt */
#include <stdio.h>
int main(void)
{
    FILE *fp1, *fp2;
    int c;
    /* open file abc.txt to append */
    fp1 = fopen("abc.txt", "a+");
    /* open def.txt to read */
    fp2 = fopen("def.txt", "r");
    /* read characters from fp2, put it in fp1 */
    while ((c = fgetc(fp2)) != EOF)
        fputc(c, fp1);
    rewind(fp1); fseek(fp1, 0L, SEEK_SET);
    while((c = fgetc(fp1)) != EOF)
        putchar(c);
    fclose(fp1);
    fclose(fp2);
    return 0;
}
```

"a+"

Modes

mode	**Description**
`"r+"`	It opens a file for reading and writing. The file must exist.
`"a+"`	It opens a file for reading and appending.

Modes

mode	**Description**
`"r+"`	It opens a file for reading and writing. The file must exist.
`"a+"`	It opens a file for reading and appending.
`"w+"`	It opens a file for reading and writing.

Modes

mode	**Description**
`"r+"`	It opens a file for reading and writing. The file must exist.
`"a+"`	It opens a file for reading and appending.
`"w+"`	It opens a file for reading and writing. It overwrites the old content.

"w+"

```c
#include <stdio.h>

int main(void)
{

    return 0;
}
```

"w+"

```
#include <stdio.h>

int main(void)
{


    /* open the file */

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open the file */
    fp = fopen("abc.txt", "w+");

    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */

    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;

    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */
    fprintf(fp, "Hello, world!\n");

    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;


    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */
    fprintf(fp, "Hello, world!\n");

    /* go to the beginning */




    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;


    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */
    fprintf(fp, "Hello, world!\n");

    /* go to the beginning */
    rewind(fp);




    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

int main(void)
{
    FILE *fp;


    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */
    fprintf(fp, "Hello, world!\n");

    /* go to the beginning */
    rewind(fp);

    /* read and print */



    fclose(fp);

    return 0;
}
```

"w+"

```c
#include <stdio.h>

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

    /* open the file */
    fp = fopen("abc.txt", "w+");

    /* write to the file */
    fprintf(fp, "Hello, world!\n");

    /* go to the beginning */
    rewind(fp);

    /* read and print */
    while((c = fgetc(fp)) != EOF)
        putchar(c);

    fclose(fp);

    return 0;
}
```

"w+"

"w+"

```
$
```

"w+"

```
$ cat abc.txt ↵
```

"\w+"

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

"w+"

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

"w+"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cc test.c ↵
$ ./a.out ↵
Hello, world!
$
```

"w+"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cc test.c ↵
$ ./a.out ↵
Hello, world!
$ cat abc.txt
```

"w+"

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ cc test.c ↵
$ ./a.out ↵
Hello, world!
$ cat abc.txt ↵
Hello, world!
$
```

`fseek` *constants*

constant	**Description**
`SEEK_SET`	Beginning of file
`SEEK_CUR`	Current position of the file pointer
`SEEK_END`	End of file

End of file

End of file

```
#include <stdio.h>

int main(void)
{
    FILE *fp;


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









    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

int main(void)
{
    FILE *fp;


    fp = fopen("abc.txt", "r");
    while(1) {




    }
    fclose(fp);
    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);




    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
            feof(fp)



    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
            feof(fp)  /* true if end-of-file */



    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
        if (feof(fp)) /* true if end-of-file */



    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
        if (feof(fp)) /* true if end-of-file */
            break;


    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
        if (feof(fp)) /* true if end-of-file */
            break;
        putchar(c);
    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
        if (feof(fp)) /* true if end-of-file */
            break;
        putchar(c);
    }

    fclose(fp);

    return 0;
}
```

End of file

```
#include <stdio.h>

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

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

    while(1) {
        c = fgetc(fp);
        if (feof(fp)) /* true if end-of-file */
            break;
        putchar(c);
    }

    fclose(fp);

    return 0;
}
```

Print file backwards

Print file backwards

```
#include <stdio.h>
int main(void)
{









    return 0;
}
```

Print file backwards

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

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;


    fp = fopen("abc.txt", "r");    /* open the file */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;


    fp = fopen("abc.txt", "r");    /* open the file */
                                   /* go to the end */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;


    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;


    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
                                   /* get current position */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;



    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
                           /* till beginning of file */









    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */









    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */
                                  /*go back one char*/




    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/




    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */

    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
                                  /* read character */




    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);            /* read character */



    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
                                  /* print character */


    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
        putchar(c);               /* print character */


    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
        putchar(c);               /* print character */
                                     /* fp has moved */

    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);          /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
        putchar(c);               /* print character */
        fseek(fp, -1L, SEEK_CUR);   /* fp has moved */

    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
        putchar(c);               /* print character */
        fseek(fp, -1L, SEEK_CUR);   /* fp has moved */
                              /* read current position */
    }
    return 0;
}
```

Print file backwards

```
#include <stdio.h>
int main(void)
{
    FILE *fp;
    long pos;           /* file size could be large */
    int c;
    fp = fopen("abc.txt", "r");    /* open the file */
    fseek(fp, 0L, SEEK_END);       /* go to the end */
    pos = ftell(fp);         /* get current position */
    while (pos > 0) {     /* till beginning of file */
        fseek(fp, -1L, SEEK_CUR); /*go back one char*/
        c = fgetc(fp);             /* read character */
        putchar(c);               /* print character */
        fseek(fp, -1L, SEEK_CUR);   /* fp has moved */
        pos = ftell(fp);   /* read current position */
    }
    return 0;
}
```

Print file backwards

```
$
```

Print file backwards

```
$ cat abc.txt ↵
```

Print file backwards

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

Print file backwards

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

Print file backwards

```
$ cat abc.txt ↵
The quick
Brown fox jumps
oVER
the
lazy DOG.
$ ./a.out ↵
.GOD yzal
eht
REVo
spmuj xof nworB
kciuq ehT
$
```