ungetc() in C/C++

- Difficulty Level : Easy
- Last Updated : 25 Oct, 2018

The **ungetc()** function takes a single character and shoves it back onto an input stream. It is the opposite of the getc() function, which reads a single character from an input stream. Also, ungetc() is an input function, not an output function.

Syntax:

```
int ungetc(int char, FILE *stream)
```

Parameters:

- **char**: specifies the int promotion of the character to be put back. The value is internally converted to an unsigned char when put back.
- **stream**: specifies the pointer to a FILE object that identifies an input stream.

Return Value: The function returns two kind of values.

- On success, the ungetc() function returns the character ch.
- On failure, EOF is returned without changing the stream.

Important points about the function:

1. The ungetc() function pushes the byte specified by char (converted to an unsigned char) back onto the input stream pointed to by stream.
2. The pushed-back bytes is returned by subsequent reads on that stream in the reverse order of their pushing.
3. A successful intervening call (with the stream pointed to by stream) to a file-positioning function (fseek(), fsetpos(), or rewind()) discards any pushed-back bytes for the stream.
4. The external storage corresponding to the stream shall be unchanged.
5. A successful call to ungetc() clears the end-of-file indicator for the stream.
6. The value of the file-position indicator for the stream after reading or discarding all pushed-back bytes shall be the same as it was before the bytes were pushed back.
7. The file-position indicator is decremented by each successful call to ungetc(), if its value was 0 before a call, its value is unspecified after the call.

Below programs illustrate the above function.

Program 1:

```
#include <stdio.h>

int main()
{
    FILE* f;
    int char;
    char buffer[256];

    // read a file
    f = fopen("use1.txt", "r");

    // when no data
    if (f == NULL) {
        printf("Error in opening file");
        return (-1);
    }

    // read lines till end
    while (!feof(f)) {
        // get line
        char = getc(f);
        // replace ! with +
        if (char == '!') {
            ungetc('+', f);
        }
        // if not
        else {
            ungetc(c, f);
        }
        fgets(buffer, 255, f);
        fputs(buffer, stdout);
    }
    return 0;
}
```

Let us assume, we have a text file use1.txt, which contains the following data. This file will be used as an input for our example program, then the input and output are shown below:

```
Input: !c standard library
       !library function stdio.h-ungetc()
Output: +c standard library
        +library function stdio.h-ungetc()
```

Program 2:

```
// C program for taking input till we
// get 1 at the input
#include <stdio.h>
int main()
{
    int ch;

    // reads characters from the stdin and show
    // them on stdout until encounters '1'
    while ((ch = getchar()) != '1')
        putchar(ch);

    // ungetc() returns '1' previously
    // read back to stdin
    ungetc(ch, stdin);

    // getchar() attempts to read
    // next character from stdin
    // and reads character '1' returned
    // back to the stdin by ungetc()
    ch = getchar();

    // putchar() displays character
    putchar(ch);
    return 0;
}
```