2020年度　工学システム学類

プログラミング序論Ｂ　【レポート課題３】　※演習の課題は別にあります。

学籍番号:Student ID（202018021）　氏名:Name（Fu, Ziyu）

次のプログラムの動作を解説せよ。解答スペースは自由に変更して構わない。

Explain the behavior of the following program.

```
main ()
{
        int c;
        if ( (c=getchar()) != EOF) {
                main();
                printf (“%c”, c);
        }
        return 0;
        printf (“¥n”);
}
```

(Reponses on the next page.)

Regarding the returns of **getchar**, the C11 standard documentation says:
The **getchar** function returns the next character from the input stream pointed to by **stdin**. If the stream is at end-of-file, the end-of-file indicator for the stream is set and **getchar** returns **EOF**.

With this understanding in mind, it is easy to explain the behavior of the sample program. The **main** function takes one character from **stdin** and checks if it is **EOF**. If it isn't **EOF**, the program calls **main** again and checks the next character. This recursion is eventually broken by the detection of a **EOF** indicator.

In the last call of **main**, the end-of-file indicator doesn't fall under the **if** condition, hence successfully returning the last **main**. The program then goes back to the preceding recursion and prints the previous character in line and returning that **main** as well. The process keeps going back until the very first call of **main** returns successfully. The resulting output is the **stdin** input being printed out backwards.

Here is a sample output:

```
abcdefg
^D
gfedcba
```

Note:

1. **EOF** can be simulated in standard input by **Control+D** on a Unix / Linux based operating system (macOS), or **Control+Z** on Windows.

2. I don't exactly understand the point of the last **printf** after the **return** line in **main**. I don't believe that line will ever get executed.

Reference:
C11 standard (ISO/IEC 9899:2011): 7.21.7.6 The getchar function (p: 332)