

Step 1

```
1 #include <stdio.h>
2
3 int factorial(int n)
4 {
5     int result = 1;
6     while(n-->0)
7         result *= n;
8     return result;
9 }
10
11 int main()
12 {
13     int n = 5;
14     int val;
15     printf("** %d factorial **\n",n);
16     val = factorial(n);
17     printf("%d\n",val);
18     return 0;
19 }
```

```
mjswindells@ubuntu:~/A$ ls
fact.c
```

나오는 코드 작성.

Step 2

```
mjswindells@ubuntu:~/A$ gcc -g -o fact fact.c
mjswindells@ubuntu:~/A$ ./fact
** 5 factorial **
0
```

gcc -g -o fact fact.c : GDB를 사용할 때 line이 보이게 하기 위해 -g 옵션을 추가해준다.

-g : 디버깅정보 삽입으로 심볼 문자열, 주소 컴파일에 사용된 소스파일 등의 정보를 포함

Step 3

```
mjswindells@ubuntu:~/A$ gdb fact
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License: GPLv3+: GNU GPL version 3 or later <http://
```

`gdb fact` : gdb를 실행시키고 바로 `fact`를 읽는다.

```
(gdb) b factorial
Breakpoint 1 at 0x1149: file fact.c, line 4.
(gdb) r
Starting program: /home/mjswindells/A/fact
** 5 factorial **

Breakpoint 1, factorial (n=21845) at fact.c:4
4      {
(gdb) n
5          int result = 1;
(gdb) display n
1: n = 5
(gdb) display result
2: result = 21845
(gdb) n
7          while(n--){
1: n = 5
2: result = 1
(gdb) n
8              result *= n;
1: n = 4
2: result = 1
(gdb) n
7          while(n--){
1: n = 4
2: result = 4
```

`b factorial` : 브레이킹 포인트를 `factorial` 함수로 설정하였다.

`r` : 시작

`n` : 함수 내부로 들어가지 않고 수행 후 다음 행으로 이동 (현재 행 수행 후 정지)

`display n(display result)` : 넘어갈 때마다 화면에 변수 정보를 출력해준다.

첫번째 잘못된 부분:

위 결과에서 보면 `factorial` 함수를 실행하는 과정의 `while(n--)` 에서 `n=5 result=1` 로 잘 설정되어진 것처럼 보인다.

하지만 바로 그 다음 라인에서 `n=4`로 설정이 된다. 즉 후위연산자로 인해 `while`에서는 `n`을 5로 인식하지만 `while` 내부에서는 `n`을 4로 인식한다. 그렇기에 `result`는 처음에 5가 되는 것이 아니라 4가 되는 것을 볼 수 있다.

=>

```

(gdb) n
7           while(n--){
1: n = 1
2: result = 24
(gdb) n
8           result *= n;
1: n = 0
2: result = 24
(gdb) n
7           while(n--){
1: n = 0
2: result = 0
(gdb) n
11          return result;
1: n = -1
2: result = 0
(gdb) n
12      }
1: n = -1
2: result = 0

```

두번째 잘못된 부분:

이런식에 작동으로 인해 while(n--) 에서 n을 1로 인식할 경우 내부에선 0으로 인식하게 된다

이렇게 되면 결국 return해야하는 result의 값은 0으로 설정이 된다.

따라서 원하는 값 120이 아닌 0이 return된다.

개선방안

```

7      while(n){
8          result *= n;
9          n--;
10     }

```

```

mjswindells@ubuntu:~/A$ gcc -o fact fact.c
mjswindells@ubuntu:~/A$ ./fact
** 5 factorial **
120

```

While 문을 위와 같이 바꾸게 되면 같은 n으로 인식을 하게 되어 원하는 값인 120을 얻을 수 있다.