

```
C task2.c 3 x
C task2.c > main()
1 // 1. 이진법으로 무한한 수를 처리할 수 있다?
2 // 2. 컴퓨터로 무한한 수를 처리할 수 있다?
3 // 3. 두 양수의 덧셈 결과는 항상 양수이다?
4 // 4. 부동 소수점 연산으로 정확한 소수 연산을 할 수 있다?
5 #include <stdio.h>
6 int main() {
7     printf("12345678901234567890 is %d\n", 12345678901234567890);
8     printf("12345678901234567890 is %s\n", "12345678901234567890");
9     printf("1234567890 + 1234567890 = %d\n", 1234567890 + 1234567890);
10    printf("12345678901234567890.0 = %lf\n", 12345678901234567890.0);
11    printf("1.1 = %.20lf\n", 1.1);
12
13    return 0;
14 }
```

문제 3 출력 디버그 콘솔 터미널 포트

```
• → Tasks ll
total 288
-rw-r--r--@ 1 kogandhi05 staff 103K 3 15 02:24 20254482 고 형 인 과제 01.pdf
-rwxr-xr-x@ 1 kogandhi05 staff 33K 3 21 01:41 task2
-rw-r--r--@ 1 kogandhi05 staff 619B 3 21 01:40 task2.c
drwxr-xr-x@ 3 kogandhi05 staff 96B 3 21 01:41 task2.dSYM
• → Tasks ./task2
12345678901234567890 is -350287150
12345678901234567890 is 12345678901234567890
1234567890 + 1234567890 = -1825831516
12345678901234567890.0 = 12345678901234567168.000000
1.1 = 1.100000000000000008882
○ → Tasks
```

1.1을 %.20f로 소수점 아래 20자리까지 1.100000000000000008882으로 출력했는데 이는 컴퓨터가 사용하는 부동 소수점 연산의 한계라고 할 수 있다. 이진법으로 소수점 아래 자리수를 나타내기 때문에 메모리 공간의 크기 또는 int 자료형의 크기로 인해 나타낼 수 있는 소수점 아래 표현의 한계로 인해 이러한 문제가 발생한다. 0.125, 0.25, 0.5 등 범위 내에서 이진법으로 표현 가능한 소수만 오차 없이 정확한 소수 연산을 할 수 있다.