Homework4

김민주

01

스키장에서 스키 장비를 임대하는데 37500원이 든다. 또 3일 이상 이용할 경우 20%를 할인 해준다.
일주일간 이용할 경우 임대 요금은 얼마일까 ? (연산 과정은 모두 함수로 돌린다)

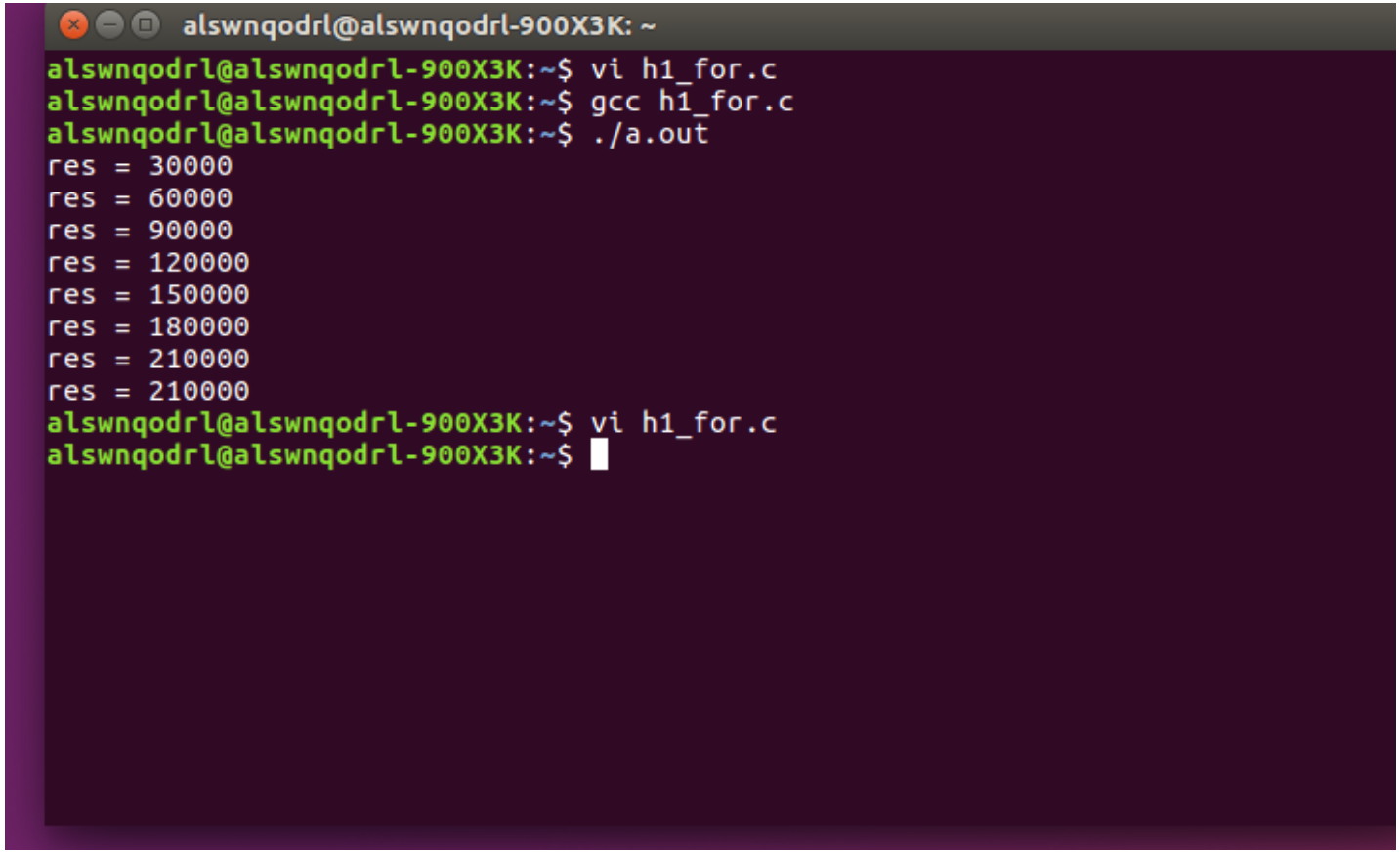
alswnqodrl@alswnqodrl-900X3K: ~

```c
#include <stdio.h>

// input: first - day, second - 37500
int borrow_equip(int day, double money)
{
    int i = 0, res = 0;
    double rate = 1.0;
    double tmp = 0;

    if(day >= 3)
    {
        rate = 0.8;
        tmp = money * rate;
    }

    for(i = 0; i < 7; i++)
    {
        res += tmp;
        printf("res = %d\n", res);
    }

    return res;
}
```

2,0-1 Top

01

스키장에서 스키 장비를 임대하는데 37500원이 든다. 또 3일 이상 이용할 경우 20%를 할인 해준다.
일주일간 이용할 경우 임대 요금은 얼마일까 ? (연산 과정은 모두 함수로 돌린다)



02

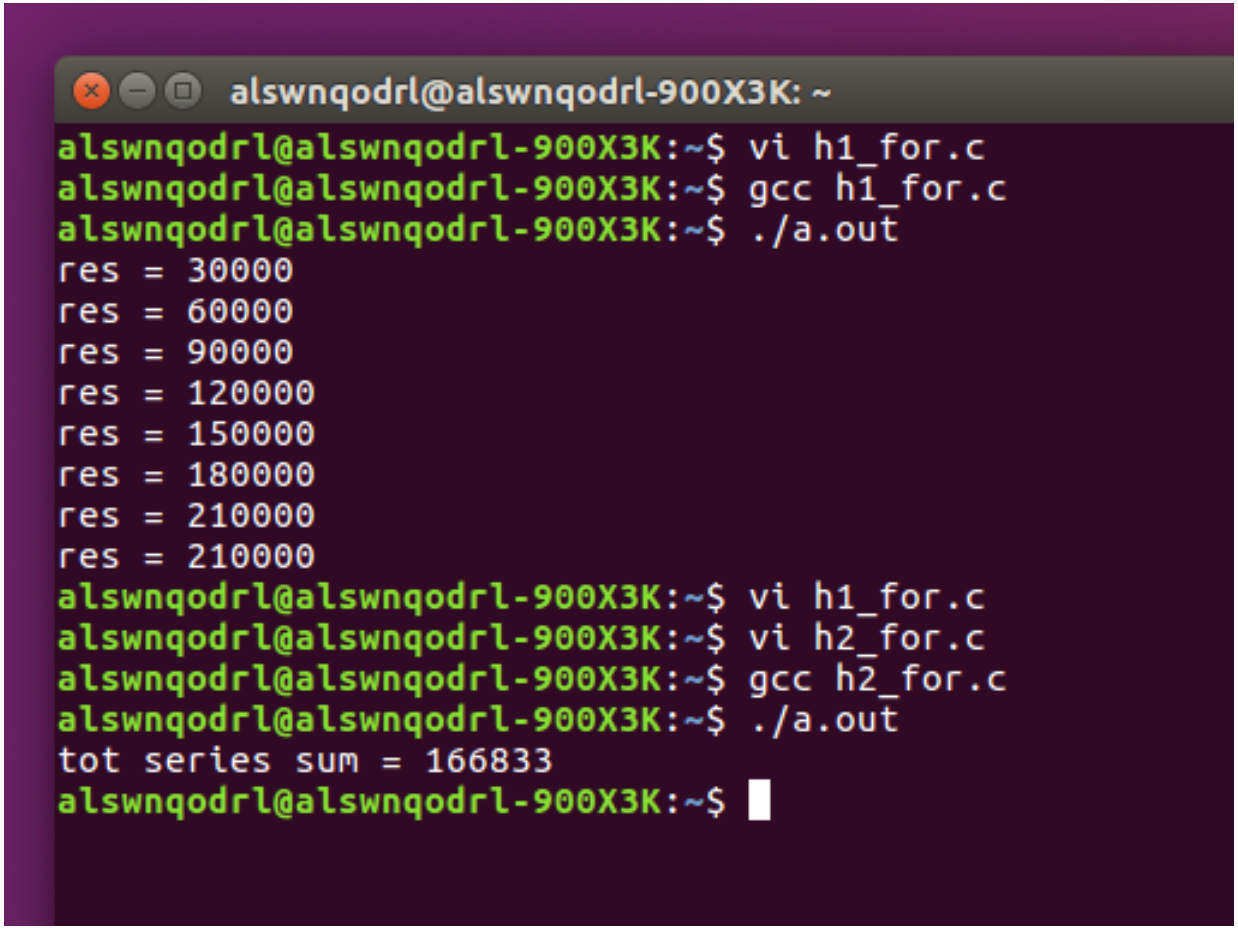
1 ~ 1000사이에 3의 배수의 합을 구하시오.

alswnqodrl@alswnqodrl-900X3K: ~

```
#include <stdio.h>
// synthesis: first - start, second - end, three - times
int syn(int start, int end, int times)
{
    int res = 0, i = start;

    for(i = start;i  < end + 1;i++)
    {
        if(!(i % 3))
        {
            res += i;
        }

    }

    return res;
}

int main(void)
{
    printf("tot series sum = %d\n", syn(1, 1000, 3));
    return 0;
}
"h2_for.c" 23L, 372C
```

02

1 ~ 1000사이에 3의 배수의 합을 구하시오.



03

1 ~ 1000사이에 4나 6으로 나눠도 나머지가 1인 수의 합을 출력하라.

alswnqodrl@alswnqodrl-900X3K: ~

```c
#include <stdio.h>

int syn(int start, int end, int t1, int t2)
{
    int res = 0, i = start;

    for(i = start; i < end + 1;i++)
    {
        if(((i % 4) == 1) || ((i % 6) == 1))
        {
            res += i;
        }
    }

    return res;
}

int main(void)
{
    printf("tot series sum = %d\n", syn(1, 1000, 4, 6));
    return 0;
}
~
"h5_for.c" 22L, 340C
```

03

1 ~ 1000사이에 4나 6으로 나눠도 나머지가 1인 수의 합을 출력하라.

```
alswnqodrl@alswnqodrl-900X3K: ~
res = 150000
res = 180000
res = 210000
res = 210000
alswnqodrl@alswnqodrl-900X3K:~$ vi h1_for.c
alswnqodrl@alswnqodrl-900X3K:~$ vi h2_for.c
alswnqodrl@alswnqodrl-900X3K:~$ gcc h2_for.c
alswnqodrl@alswnqodrl-900X3K:~$ ./a.out
tot series sum = 166833
alswnqodrl@alswnqodrl-900X3K:~$ vi h2_for.c
alswnqodrl@alswnqodrl-900X3K:~$ vi h3_for.c
alswnqodrl@alswnqodrl-900X3K:~$ gcc h3_for.c
alswnqodrl@alswnqodrl-900X3K:~$ ./a.out
tot series sum = 166167
alswnqodrl@alswnqodrl-900X3K:~$ vi h4_for.c
alswnqodrl@alswnqodrl-900X3K:~$ gcc h4_for.c
alswnqodrl@alswnqodrl-900X3K:~$ ./a.out
7       14      21      28      35      42      49
alswnqodrl@alswnqodrl-900X3K:~$ vi h4_for.c
alswnqodrl@alswnqodrl-900X3K:~$ vi h5_for.c
alswnqodrl@alswnqodrl-900X3K:~$ gcc h5_for.c
alswnqodrl@alswnqodrl-900X3K:~$ ./a.out
tot series sum = 166167
alswnqodrl@alswnqodrl-900X3K:~$
```

04

7의 배수로 이루어진 값들이 나열되어 있다고 가정한다.

함수의 인자(input)로 항의 갯수를 받아서 마지막 항의 값을 구하는 프로그램을 작성하라.

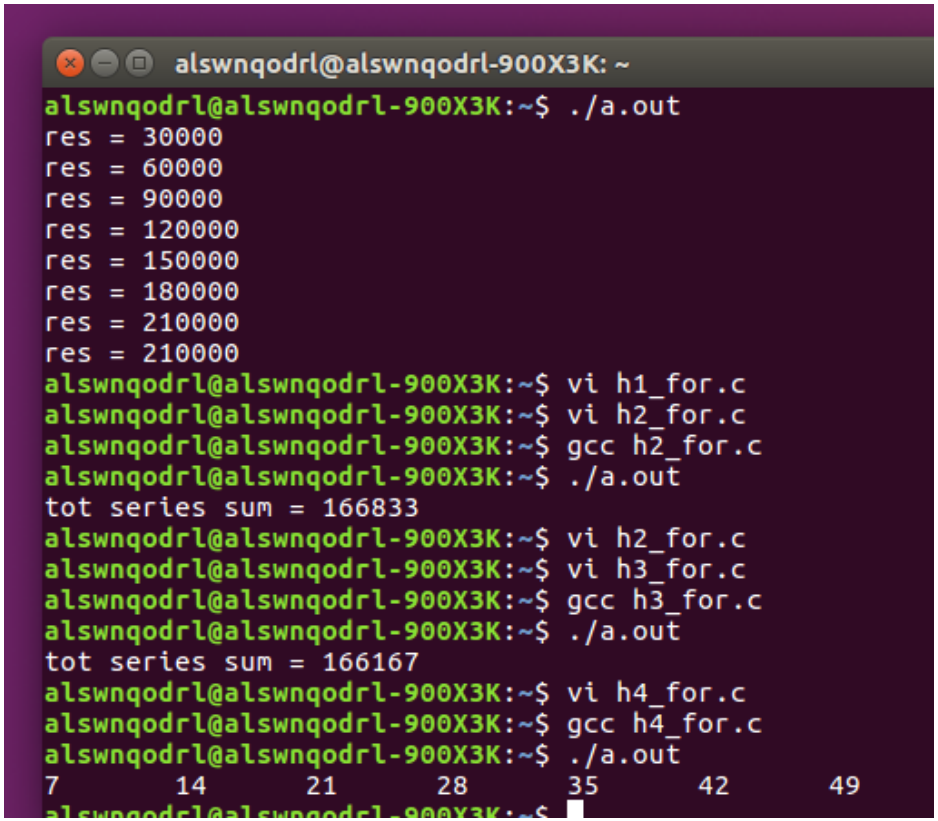
alswnqodrl@alswnqodrl-900X3K: ~

```c
#include <stdio.h>

void print_seven_series(int num)
{
    int i;

    for(i = 0;i < num; i++ )
    {
        if(i < num - 1)
        {
            printf("%d\t", (i + 1) * 7);
        }
        else
        {
            printf("%d\n", (i + 1) * 7);
        }
    }
}

int main(void)
{
    print_seven_series(7);
    return 0;
"h4_for.c" 24L, 330C
```

04

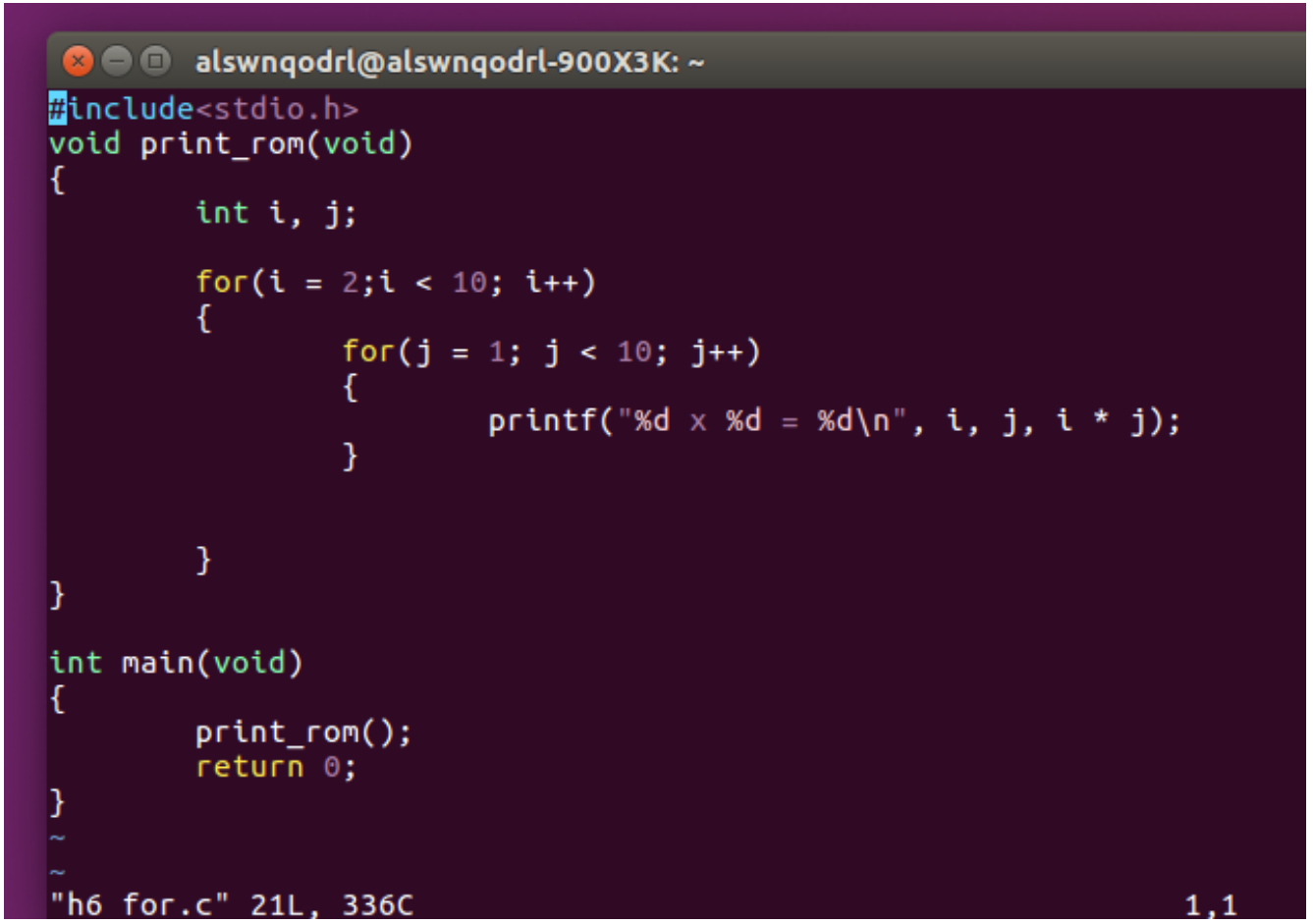
7의 배수로 이루어진 값들이 나열되어 있다고 가정한다.

함수의 인자(input)로 항의 갯수를 받아서 마지막 항의 값을 구하는 프로그램을 작성하라.



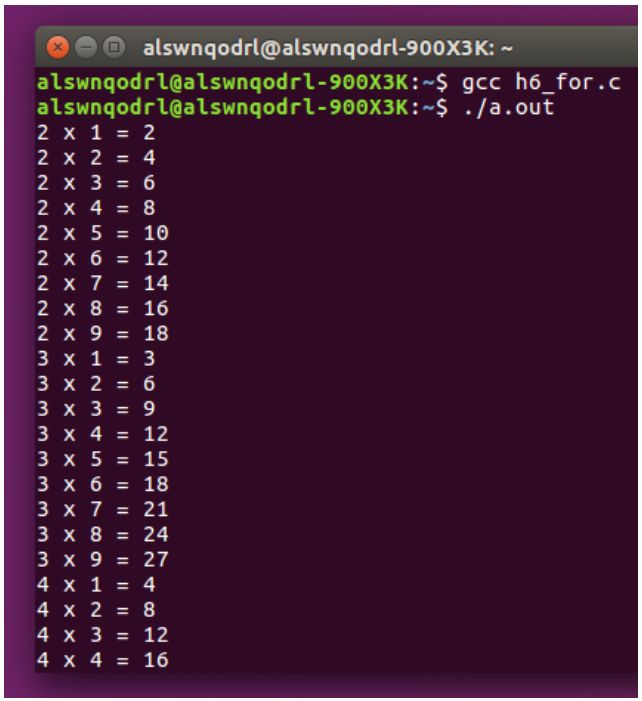
05

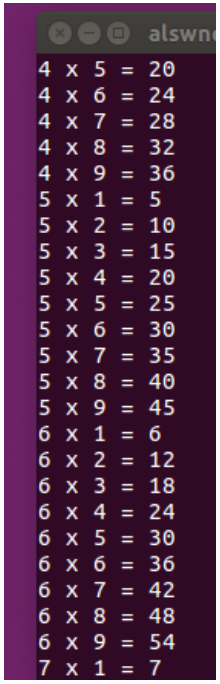
구구단을 만들어보시오.

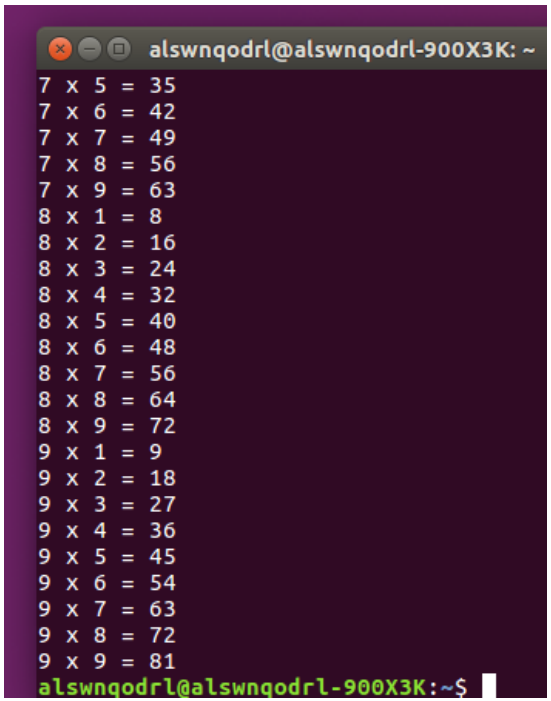


05

구구단을 만들어보시오.







06

오늘 배운 내용 복습

: goto와 이점과 파이프라인

Add (**fetch**)	Add (**decode**)	Add (**excute**)				
	mov (**fetch**)	mov (**decode**)	mov (**excute**)			
		call (**fetch**)	call (**decode**)	call (**excute**)		
			mov (**fetch**)	mov (**decode**)	mov (**excute**)	
				mov (**fetch**)	mov (**decode**)	mov (**excute**)

파이프 라인 3

Call이나 jmp를 cpu instruction이라고 하고 레벨에서 분기 명령어라고 한다.
이들은 cpu 파이프라인에 매우 치명적인다.
이 때 goto를 사용하게 되면 버리게 되는 clock을 줄일 수 있다는 장점이 있다. 3Clock x 분기회수 것을 goto를 사용하여 jmp한 번으로 줄이는 것이다.

Call 실행하면
Call 하단의 clock(이 경우 3)이
모두 날아갔다가 다시 시행되어야 하므로 cpu에 치명적임

07

오늘 배운 내용 복습:

fib함수 동작분석

```
alswnqodrl@alswnqodrl-Z20NH-AS51B1U: ~/my_proj/Homework/mi
#include <stdio.h>
int fib(int num)
{
        if(num==1||num==2)
        return 1;
        else
        return fib(num-1)+fib(num-2);
}

int main(void)
{
int result, final_val;
        printf("피보나치수열의 항의 개수를 입력하시오:");
        scanf("%d", &final_val);
        result=fib(final_val);
        printf("%d번째 항의 수는 = %d\n", final_val, result);
        return 0;
}
```

```
Reading symbols from a.out...done.
(gdb) b main
Breakpoint 1 at 0x400642: file fib.c, line 11.
(gdb) r
Starting program: /home/alswnqodrl/my_proj/Homework/minjukim/a.out

Breakpoint 1, main () at fib.c:11
11      {
(gdb) n
13              printf("피보나치수열의 항의 개수를 입력하시오:");
(gdb)
14              scanf("%d", &final_val);
(gdb)
피보나치수열의 항의 개수를 입력하시오:6
15              result=fib(final_val);
(gdb) bt
#0  main () at fib.c:15
(gdb) s
fib (num=6) at fib.c:4
4               if(num==1||num==2)
(gdb)
7               return fib(num-1)+fib(num-2);
(gdb) bt
#0  fib (num=6) at fib.c:7
#1  0x0000000000400680 in main () at fib.c:15
(gdb) s
fib (num=5) at fib.c:4
4               if(num==1||num==2)
(gdb)
7               return fib(num-1)+fib(num-2);
(gdb) bt
#0  fib (num=5) at fib.c:7
#1  0x0000000000400622 in fib (num=6) at fib.c:7
#2  0x0000000000400680 in main () at fib.c:15
(gdb) s
fib (num=4) at fib.c:4
4               if(num==1||num==2)
(gdb)
7               return fib(num-1)+fib(num-2);
(gdb) bt
#0  fib (num=4) at fib.c:7
#1  0x0000000000400622 in fib (num=5) at fib.c:7
#2  0x0000000000400622 in fib (num=6) at fib.c:7
#3  0x0000000000400680 in main () at fib.c:15
```

```
(gdb) s
fib (num=3) at fib.c:4
4               if(num==1||num==2)
(gdb)
7               return fib(num-1)+fib(num-2);
(gdb) bt
#0  fib (num=3) at fib.c:7
#1  0x0000000000400622 in fib (num=4) at fib.c:7
#2  0x0000000000400622 in fib (num=5) at fib.c:7
#3  0x0000000000400622 in fib (num=6) at fib.c:7
#4  0x0000000000400680 in main () at fib.c:15
(gdb) s
fib (num=2) at fib.c:4
4               if(num==1||num==2)
(gdb)
5               return 1;
(gdb) bt
#0  fib (num=2) at fib.c:5
#1  0x0000000000400622 in fib (num=3) at fib.c:7
#2  0x0000000000400622 in fib (num=4) at fib.c:7
#3  0x0000000000400622 in fib (num=5) at fib.c:7
#4  0x0000000000400622 in fib (num=6) at fib.c:7
#5  0x0000000000400680 in main () at fib.c:15
(gdb) s
8       }
(gdb)
fib (num=1) at fib.c:4
4               if(num==1||num==2)
(gdb) bt
#0  fib (num=1) at fib.c:4
#1  0x0000000000400631 in fib (num=3) at fib.c:7
#2  0x0000000000400622 in fib (num=4) at fib.c:7
#3  0x0000000000400622 in fib (num=5) at fib.c:7
#4  0x0000000000400622 in fib (num=6) at fib.c:7
#5  0x0000000000400680 in main () at fib.c:15
(gdb) s
5               return 1;
(gdb) finish
Run till exit from #0  fib (num=1) at fib.c:5
0x0000000000400631 in fib (num=3) at fib.c:7
7               return fib(num-1)+fib(num-2);
Value returned is $1 = 1
(gdb)
```

07

오늘 배운 내용 복습:

fib함수 동작분석

Main Res 6	Fin 6
Fib 6 Num num-1 (8)	
Fib 5 Num num-1 (5)	Fib 4 num num-2 (3)
Fib 4 num num-1 (3)	Fib 3 num num-2 (2)
Fib 3 num num-1 (2)	Fib 2 num num-2 (1)
Fib 2 num num-1 (1)	Fib 1 num num-2 (1)