

3일 차

사물 인터넷(IoT/ICT)환경에서의 임베디드 응용SW 개발자 양성과정

은태영

1번.

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)

1번.

```
tewill@tewill-B85M-D3H: ~/my_proj
#include <stdio.h>

int discount(int day)
{
    int result;

    if(day <= 0){
        printf("잘못된 날짜입니다.\n");
    } else if(day < 3){
        result = day * 37500;
    } else {
        result = day * 4 * 37500 / 5;
    }

    return result;
}

int main(void)
{
    int result;
    result = discount(3);

    printf("임대 요금 : %d\n",result);

    return 0;
}

~
```

17,0-1 All

1 번.

```
tewill@tewill-B85M-D3H: ~/my_proj
tewill@tewill-B85M-D3H:~/my_proj$ ./debug
임대 요금 : 90000
tewill@tewill-B85M-D3H:~/my_proj$
```

3번.

1 ~ 1000사이에 3의 배수의 합을 구하시오.

3 번.

```
tewill@tewill-B85M-D3H: ~/my_proj
#include <stdio.h>

int addmultiple(int min,int max)
{
    int result = 0;

    while(min < max)
    {
        if(!(min%3)){
            result += min;
        }
        min++;
    }

    return result;
}

int main(void)
{
    int result;

    result = addmultiple(1, 1000);
    printf("1~1000 사이의 3 배수의 합 : %d\n", result);
    return 0;
}
~
~
"que03.c" 25L, 300C 1,1 All
```

3번.

```
tewill@tewill-B85M-D3H: ~/my_proj
tewill@tewill-B85M-D3H:~/my_proj$ ./debug
1~1000 사이의 3 배수의 합 : 166833
tewill@tewill-B85M-D3H:~/my_proj$
```

4번.

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

4번.

```
tewill@tewill-B85M-D3H: ~/my_proj
#include <stdio.h>

int cal(int min, int max)
{
    int result = 0;

    while(min < max)
    {
        if(1 == min%4 && 1 == min%6){
            result += min;
        }
        min++;
    }

    return result;
}

int main(void)
{
    int result;

    result = cal(1, 1000);

    printf("결과 값 : %d\n", result);

    return 0;
}
"que04.c" 27L, 282C
```

4번.

```
tewill@tewill-B85M-D3H: ~/my_proj
tewill@tewill-B85M-D3H:~/my_proj$ ./debug
결과 값 : 41916
tewill@tewill-B85M-D3H:~/my_proj$
```

5번.

7의 배수로 이루어진 값들이 나열되어 있다고 가정한다.
함수의 인자(input)로 항의 갯수를 받아서 마지막 항의 값을
구하는 프로그램을 작성하라.

5번.

```
tewill@tewill-B85M-D3H: ~/my_proj
#include <stdio.h>

int multiple(int max)
{
    int result;

    result = 7 * max;

    return result;
}

int main(void)
{
    int result;
    result = multiple(10);
    printf("7의 10번째 배수 : %d\n", result);
    return 0;
}
~
~
~
~
~
~
~
"que05.c" 19L, 214C
```

5번.

```
tewill@tewill-B85M-D3H: ~/my_proj
tewill@tewill-B85M-D3H:~/my_proj$ ./debug
7의 10번째 배수 : 70
tewill@tewill-B85M-D3H:~/my_proj$
```

7번.

C로 함수를 만들 때, Stack이란 구조가 생성된다.

이 구조가 어떻게 동작하는지 Assembly Language를 해석하며 기술해보시오.

esp, ebp, eip등의 Register에 어떤 값이 어떻게 들어가는지 등등 메모리에 어떤 값들이 들어가는지 등을 자세히 기술하시오.

7번.

```
int mult2(int num){  
    return num * 2;  
}
```

```
int main(void){  
    int i, sum = 0, result;  
    for(i = 0; i < 5; i++)  
        sum += i;  
    result = mult2(sum);  
    return 0;  
}
```

7번.

```
tewill@tewill-B85M-D3H: ~/my_proj
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/tewill/my_proj/debug

Breakpoint 2, main () at que07.c:9
9      {
(gdb) disas
Dump of assembler code for function main:
=> 0x0000000004004e4 <+0>:      push    %rbp
    0x0000000004004e5 <+1>:      mov     %rsp,%rbp
    0x0000000004004e8 <+4>:      sub     $0x10,%rsp
    0x0000000004004ec <+8>:      movl    $0x0,-0x8(%rbp)
    0x0000000004004f3 <+15>:     movl    $0x0,-0xc(%rbp)
    0x0000000004004fa <+22>:     jmp     0x400506 <main+34>
    0x0000000004004fc <+24>:     mov     -0xc(%rbp),%eax
    0x0000000004004ff <+27>:     add     %eax,-0x8(%rbp)
    0x000000000400502 <+30>:     addl    $0x1,-0xc(%rbp)
    0x000000000400506 <+34>:     cmpl    $0x4,-0xc(%rbp)
    0x00000000040050a <+38>:     jle     0x4004fc <main+24>
    0x00000000040050c <+40>:     mov     -0x8(%rbp),%eax
    0x00000000040050f <+43>:     mov     %eax,%edi
    0x000000000400511 <+45>:     callq   0x4004d6 <mult2>
    0x000000000400516 <+50>:     mov     %eax,-0x4(%rbp)
    0x000000000400519 <+53>:     mov     $0x0,%eax
    0x00000000040051e <+58>:     leaveq
    0x00000000040051f <+59>:     retq
End of assembler dump.
(gdb) █
```


10번.

구구단을 만들어보시오.

10번.

```
tewill@tewill-B85M-D3H: ~/my_proj
#include <stdio.h>

void multiplication(int num1, int num2)
{
    int min1 = 1, min2 = 1;
    while(min1 < num1)
    {
        min1++;
        while(min2 <= num2)
        {
            printf("%d * %d = %d\n", min1, min2, min1 * min2);
            min2++;
        }
        min2 = 1;
        printf("\n");
    }
}

int main()
{
    multiplication(9,9);

    return 0;
}
"que10.c" 27L, 302C                                     1,1      All
```

10번.

```
tewill@tewill-B85M-D3H: ~/my_proj
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72

9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81

tewill@tewill-B85M-D3H:~/my_proj$
```

12번.

리눅스의 Debugging 하는 방법에 대해 기술해 보시오.

12번.