

# 사물인터넷(IoT/ICT)환경의 임베디드 개발자 양성과정

과제 3 일차 이우석

# [C programing] - (문제 1, 3, 4, 5, 7, 10, 12 번)

## 1 번문제 과정

```
#include <stdio.h>

int q1()
{
    int fee = 37500;
    int day=0;
    int result=0;

    printf("빌리는 기간 : ");
    scanf("%d",&day);

    if(day>=3)
    {
        result =(day*fee)*0.8;
    }

    else
    {
        result=day*fee;
    }

    return result;
}

int main(void)
{
    printf("임대 요금 : %d\n", q1());

    return 0;
}
```

## 1 번문제 결과

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
빌리는 기간 : 3
임대 요금 : 90000
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
빌리는 기간 : 2
임대 요금 : 75000
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
빌리는 기간 : 1
임대 요금 : 37500
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
```

### 3번문제 과정

```
#include <stdio.h>

int q3()
{
    int num=1;

    int result=0;

    while(num<=1000)
    {
        if(num%3==0)
        {
            result=result+num;
        }
        num++;
    }
    return result;
}

int main(void)
{
    printf("1 ~ 1000 사이의 3의 배수의 합 : %d\n", q3());

    return 0;
}
```

### 3번문제 결과

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
1 ~ 1000 사이의 3의 배수의 합 : 166833
```

## 4번문제 과정

```
#include <stdio.h>

int q4()
{
    int num=1;
    int result=0;
    while(num<=1000)
    {
        if((num%4==1)&&(num%6==1))
        {
            result=result+num;
        }
        num++;
    }

    return result;
}

int main(void)
{
    printf("1~1000 사이에 4나 6으로 나뉘도 나머지가 1인 수의 합 : %d\n", q4());

    return 0;
}
```

## 4번문제 결과값

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
1~1000 사이에 4나 6으로 나뉘도 나머지가 1인 수의 합 : 41916
```

## 5번문제 과정

```
#include <stdio.h>

int q5(int input)
{
    int result=0;
    result=7*input;

    printf("마지막 항의 값 : %d\n", result);

    return 0;
}

int main(void)
{
    int input=0;
    printf("항의 개수를 입력하시오 : ");
    scanf("%d",&input);

    q5(input);

    return 0;
}
```

## 5번문제 결과값

```
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
항의 개수를 입력하시오 3
마지막 항의 값 : 21
wooseok91@wooseok91-MacBookAir:~/folder$ vi q5.c
wooseok91@wooseok91-MacBookAir:~/folder$ gcc q5.c
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
항의 개수를 입력하시오 : 5
마지막 항의 값 : 35
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
항의 개수를 입력하시오 : 4
마지막 항의 값 : 28
```

## 7번문제

Stack 은 보통 PUSH 와 POP 등으로 동작된다. 또한 sp, bp, ip 등의 Register 에는 주소값이 들어간다.  
ex)

```
(gdb) b *0x0000000000400535
Breakpoint 2 at 0x400535: file func1.c, line 11.
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/wooseok91/my_proj/Homework/sanghoonlee/debug

Breakpoint 2, main () at func1.c:11
11      {
(gdb) disas
Dump of assembler code for function main:
=> 0x0000000000400535 <+0>:      push    %rbp
0x0000000000400536 <+1>:      mov     %rsp,%rbp
0x0000000000400539 <+4>:      sub     $0x10,%rsp
0x000000000040053d <+8>:      movl    $0x3,-0x8(%rbp)
0x0000000000400544 <+15>:     mov     -0x8(%rbp),%eax
0x0000000000400547 <+18>:     mov     %eax,%edi
0x0000000000400549 <+20>:     callq   0x400526 <myfunc>
0x000000000040054e <+25>:     mov     %eax,-0x4(%rbp)
0x0000000000400551 <+28>:     mov     -0x4(%rbp),%eax
0x0000000000400554 <+31>:     mov     %eax,%esi
0x0000000000400556 <+33>:     mov     $0x4005f4,%edi
0x000000000040055b <+38>:     mov     $0x0,%eax
0x0000000000400560 <+43>:     callq   0x400400 <printf@plt>
0x0000000000400565 <+48>:     mov     $0x0,%eax
0x000000000040056a <+53>:     leaveq  %eax
0x000000000040056b <+54>:     retq
End of assembler dump.
(gdb) p/x $rsp
$1 = 0x7fffffffdfcf8
(gdb) p/x $rbp
$2 = 0x400570
```

push: stack 에 데이터를 저장

mov: a(왼쪽)를 b(오른쪽)로 데이터를 복사 해준다.

Sub: 주소값을 뺀다 (stack 에 쌓인다)

callp: push + jump = 점프 주소로 간다

(복귀주소를 가지면서)

retp: pop rip = 리턴주소로 간다

그리고 메모리에는 주소값에 할당된 데이터들이  
들어간다.

## 10 번문제 과정

```
#include <stdio.h>

int main(void)
{
    int dan=0, num=1;
    printf("단 수 : ");
    scanf("%d", &dan);

    while(num<10)
    {
        printf("%dX%d=%d \n", dan, num, dan*num);
        num++;
    }

    return 0;
}
```

## 10 번문제 결과값

```
단 수 : 2
2X1=2
2X2=4
2X3=6
2X4=8
2X5=10
2X6=12
2X7=14
2X8=16
2X9=18
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
단 수 : 3
3X1=3
3X2=6
3X3=9
3X4=12
3X5=15
3X6=18
3X7=21
3X8=24
3X9=27
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
단 수 : 4
4X1=4
4X2=8
4X3=12
4X4=16
4X5=20
4X6=24
4X7=28
4X8=32
4X9=36
wooseok91@wooseok91-MacBookAir:~/folder$ ./a.out
단 수 : 5
5X1=5
5X2=10
5X3=15
5X4=20
5X5=25
5X6=30
5X7=35
5X8=40
5X9=45
```

## 12 번문제 기술

[Visual Studio 에서 Debugging 하는 방법]

: 1. 디버거 시작 - 2. 중단점을 설정하고 디버거 시작 - 3. 응용 프로그램을 시작 - 4. 단계 명령을 사용하여 디버거에서 코드탐색 - 5.속성을 한단계씩 호출스택을 검사 - 6. 프로시저 나가기 - 7. 커서실행 - 8. 실행흐름 변경 - 9. 데이터 팁을 사용하여 변수검사 시작 - 10. 자동 및 지역창을 사용하여 변수검사 시작 - 11. 조사식 설정 - 12. 예외 검사

\*Break point 잡는 방법 :

ex) b main or b \*0x0000000000400535(주소값)

- 역 할 -

조사식 :

메모리 :

레지스터:

무엇을 알고자 할때 유용한지 기술 :



