



EDDI

Electronic Design  
Development Institute

---

# 에디로봇아카데미 임베디드 마스터 Lv# 과정

제 #기

2022. 01. 08

정성훈

# CONTENTS

## A. 수업내용 복습 (github 파일 업로드 및 예제코드)

## B. 숙제

1.  $x$  &  $\sim(2^{\text{의 } y} \text{승} - 1)$ 의 의미에 대해 서술하시오.
2. 변수의 정확한 정의에 대해 기술하시오.
3. 16진수 체계에 대해 최대한 상세히 기술하시오.
4. xor를 이용한 대소문자 변환 방식에 대해 정리해봅시다.
5. Memory 구조(Stack, Heap, Data, Text)에 대해서 기술하시오.
6. gdb를 사용하여 디버깅 하기 위해 필요한 컴파일 옵션에 대해 기술하시오.
7. 아무 변수나 선언하고 이 변수의 주소값과 실제 메모리의 값이 일치하는지 확인해보시오(디버깅)
8. 두 개의 변수를 입력받아 대소 비교를 하시오.
9. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... 피보나치 수열을 표현할 수 있도록 프로그래밍 해보시오. (학습한 내용 - 변수, 제어문만 가지고 작업해봅시다)
10. 주사위 게임을 만들어봅시다.

총 2개의 주사위가 있고 하나는 컴퓨터, 나머지는 사용자의 주사위입니다. (높은 값을 도출하면 승리했다고 출력하도록 만듭니다.)

## A. github 파일 업로드

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git remote -v
origin  https://github.com/jung011/EmbeddedMasterLv1.git (fetch)
origin  https://github.com/jung011/EmbeddedMasterLv1.git (push)
upsteam https://github.com/EDDI-RobotAcademy/EmbeddedMasterLv1.git (fetch)
upsteam https://github.com/EDDI-RobotAcademy/EmbeddedMasterLv1.git (push)
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$
```

Show Applications

## A. github 파일 업로드

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  ../c/

nothing added to commit but untracked files present (use "git add" to track)
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$
```

## A. github 파일 업로드

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git add ../
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   ../c/1/first.c
    new file:   ../c/1/flst.c
    new file:   ../c/2/bit_align.c
    new file:   ../c/2/bit_oper.c
    new file:   ../c/2/for.c
    new file:   ../c/2/if.c
    new file:   ../c/2/major.c
    new file:   ../c/2/switch.c
    new file:   ../c/2/uppor_trans.c
    new file:   ../c/2/while.c

jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$
```

## A. github 파일 업로드

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git commit -am "[SHJ] 임베디드 마
스터 레벨 1 2회차 코드작업 샘플"
[main 5ef97cd] [SHJ] 임베디드 마스터 레벨 1 2회차 코드작업 샘플
10 files changed, 170 insertions(+)
create mode 100644 "3\352\270\260\SHJ/c/1/first.c"
create mode 100644 "3\352\270\260\SHJ/c/1/fist.c"
create mode 100644 "3\352\270\260\SHJ/c/2/bit_align.c"
create mode 100644 "3\352\270\260\SHJ/c/2/bit_oper.c"
create mode 100644 "3\352\270\260\SHJ/c/2/for.c"
create mode 100644 "3\352\270\260\SHJ/c/2/lf.c"
create mode 100644 "3\352\270\260\SHJ/c/2/major.c"
create mode 100644 "3\352\270\260\SHJ/c/2/switch.c"
create mode 100644 "3\352\270\260\SHJ/c/2/uppor_trans.c"
create mode 100644 "3\352\270\260\SHJ/c/2/while.c"
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$
```

## A. github 파일 업로드

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$ git pull origin main
From https://github.com/jung011/EmbeddedMasterLv1
* branch      main      -> FETCH_HEAD
Already up to date.
jung@ubuntu:~/EmbeddedMasterLv1/3기/SHJ/test$
```

## A. github 파일 업로드

```
Jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
Jung@ubuntu:~/EmbeddedMasterLv1/371/SHJ/test$ git merge upstream/main
Already up to date.
Jung@ubuntu:~/EmbeddedMasterLv1/371/SHJ/test$
```



## A. github 파일 업로드

```
Jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
Jung@ubuntu:~/EmbeddedMasterLv1/371/SHJ/test$ git push origin main
Username for 'https://github.com': jung011
Password for 'https://jung011@github.com':
```

Show Applications

## A. 예제코드 - for문

```
Jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
#include<stdio.h>
#define MAX 10

int main(void)
{
    int i;
    for (i=0; i<MAX; i++)
    {
        printf("%d\n",i);
    }
    return 0;
}

-- INSERT --
```

## A. 예제코드 - while문

```
jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
#include<stdio.h>
#include<stdbool.h>

int main(void)
{
    int cnt = 0;
    bool loop = true;

    while(loop)
    {
        printf("cnt=%d\n", ++cnt);
        if(cnt == 7)
        {
            loop = false;
        }
    }
    return 0;
}

-- INSERT --
```

## A. 예제코드 - 난수

```
Jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
#include<stdbool.h>

int main(void)
{
    int num;
    bool loop;

    srand(time(NULL));

    while(loop)
    {
        num=rand() % 100 +1;
    }
    return 0;
}

-- INSERT --
```

7,2 All

## A. 예제 코드 - switch

```
jung@ubuntu: ~/EmbeddedMasterLv1/371/SH1/test

#include<stdio.h>
#include<time.h>
#include<stdlib.h>
#include<stdbool.h>
#define ONE 1
#define TWO 2

int main(void)
{
    int num;
    bool loop=true;

    srand(time(NULL));

    while(loop)
    {
        num=rand() % 100 +1;

        switch(num %3)
        {
            case ZERO;
            printf("ZERO\tnum=%d\n", num);
            break;

            case ONE;
            printf("ONE\tnum=%d\n", num*3);
            break;

            case TWO;
            printf("게임이 종료되었습니다.\n");
            loop = false;
            break;
        }
    }
    return 0;
}

-- INSERT --
```

## A. 예제코드 - 대소문자

```
jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
#include<stdio.h>

int main(void)
{
    char c = '\0';
    printf("문자를 입력하세요:");
    scanf("%c", &c);

    if(c>96 && c<123)
    {
        printf("소문자를 입력하셨습니다!\n");
        printf("대문자는 %c\n", c - 0x20);
    }
    if (c>64 && c<91)
    {
        printf("대문자를 입력하셨습니다!\n");
        printf("소문자는 %c\n", c + 0x20);
    }
    return 0;
}

-- INSERT --
```

18,41-51 All

## A. 예제코드 - xor

jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test

```
#include<stdio.h>

int main(void)
{
    char c = '\0';

    printf("문자를 입력하세요: ");
    scanf("%c", &c);

    if((c>96 && c<123) || (c>64 && c<91));
    {
        printf("xor는 뭐든 가능하다!\n");
        printf("입력값 = %c, 변환값 = %c\n", c, c^0x20);
    }

    return 0;
}
```

## A. 예제코드 - 홀수, 짝수

```
Jung@ubuntu: ~/EmbeddedMasterLv1/371/SHJ/test
#include<stdio.h>

int main(void)
{
    int nun = 3;

    if(nun %2)
    {
        printf("홀수\n");
    }
    else
    {
        printf("짝수\n");
    }

    return 0;
}

-- INSERT --
```



## A. 예제코드 - 논리 연산자

```
Jung@ubuntu: ~/EmbeddedMasterLv1/3기/SHJ/test
#include<stdio.h>

int main(void)
{
    int num1 = 5, num2 = 10;

    printf("%d and %d = %d\n", num1, num2, num1 & num2);
    printf("%d or %d = %d\n", num1, num2, num1 | num2);
    printf("%d xor %d = %d\n", num1, num2, num1 ^ num2);
    printf("not %d = %d\n", num1, ~num1);

    return 0;
}

-- INSERT --
```

13,39-46 All



## B. 숙제

1.  $x \& \sim(2 \text{의 } y \text{승} - 1)$ 의 의미에 대해 서술하시오.

2. 변수의 정확한 정의에 대해 기술하시오.

: 특정한 데이터 타입을 가지고 있으며 데이터를 저장할 수 있는 메모리 공간

3. 16진수 체계에 대해 최대한 상세히 기술하시오.

: 16진법으로 이루어진 수의 체계

: 16진수는 HEXA로 읽고 0x로 씀

: 2진수는 많은 메모리 용량을 차지하므로 16진수로 표기함

4. xor를 이용한 대소문자 변환 방식에 대해 정리해봅시다.

5. Memory 구조(Stack, Heap, Data, Text)에 대해서 기술하시오.

Stack : 지역변수

Heap : 동적할당

Data : 전역변수

Text : 기계어

6. gdb를 사용하여 디버깅 하기 위해 필요한 컴파일 옵션에 대해 기술하시오.

: `gcc -g first.c`

7. 아무 변수나 선언하고 이 변수의 주소값과 실제 메모리의 값이 일치하는지 확인해보시오(디버깅)

8. 두 개의 변수를 입력받아 대소 비교를 하시오.

## B. 숙제 - 주사위 게임

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(void) {
    int die, reply;

    srand(time(NULL));
    die = rand() % 6 + 1;

    printf("/n 주사위를 던졌습니다.");
    scanf("%d", &reply);

    if (die < reply) {
        printf("%d 승리하였습니다. /n", die);
    }
    else{
        printf("%d 패배하였습니다. /n", die);
    }
    return 0;
}
```

## B. 속제 - 피보나치 수열

```
#include<stdio.h>

int main(void) {
    int num;
    int start = 0;
    int mid = 0;
    int end = 1;

    printf("input the integer number :");
    scanf("%d", &num);

    if (start<num) {
        for(int i=0; i<num; i++){
            if(start<num){
                printf("%d", start);

                mid = start + end;
                start = end;
                end = mid;
            }
        }
    }
    return 0;
}
```