

# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 은태영

Zero\_bird@naver.com

# 목차

- 5 일 차 복습
- 문제 은행\_79
- 문제 은행\_104
- 삼각함수
- 문제 은행\_교재\_279

# 문제 은행\_79\_1

배열의 문자열을 입력 받고,  
각 배열 요소가 짝수인 경우만을  
출력하는 함수를 작성하라.

의문 1. 포인터를 통하여 배열을 제어할 경우,  
Main에서 배열의 길이를 input 하지 않을 경우,  
배열의 길이를 알 수 없나?

# 문제 은행\_79\_1

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
#include <stdio.h>

void pri(char *p, int size)
{
    int i;
    for(i = 0; i < size; i++)
    {
        if(!(i%2))
            printf("%c", p[i]);
    }
    printf("\n");
}

int main(void)
{
    char hello[] = "hello world";
    int hello_size = sizeof(hello)/sizeof(char);
    pri(hello, hello_size);
    return 0;
}
~
~
~
```

단순하게 살기 :: [C/C++]배열의 길이를 알아보자..

11,1-8 All

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ ./debug
hlowrd
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```

# 문제 은행\_79\_3

아래와 같은 숫자들이 배열에 들어 있다고 가정한다.

3, 77, 10, 7, 4, 9, 1, 8, 21, 33

이 요소들을 배열에 거꾸로 집어넣어보자.

# 문제 은행\_79\_3

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
#include <stdio.h>

void reverse(int *num, int num_size)
{
    int temp = 0;
    int max = num_size/2, i;

    for(i = 0; i < max; i++)
    {
        temp = num[i];
        num[i] = num[num_size - i - 1];
        num[num_size - i - 1] = temp;
    }

    for(i = 0; i < num_size; i++)
        printf("%d\n", num[i]);
}

int main(void)
{
    int num[] = {3, 77, 10, 7, 4, 9, 1, 8, 21, 33};

    int num_size = sizeof(num)/sizeof(int);

    reverse(num, num_size);

    return 0;
}
```

1,1 All

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ ./debug
33
21
8
1
9
4
7
10
77
3
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```

# 문제 은행\_79\_4

위의 숫자 3, 77, 10, 7, 4, 9, 1, 8, 21, 33 에서  
홀수 번째 요소의 합과 짝수 번째 요소의 합을 곱하시오.

# 문제 은행\_79\_4

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
#include <stdio.h>

int add_mul(int *num, int size)
{
    int result1, result2, i;
    result1 = result2 = 0;

    for(i = 0; i < size; i++)
    {
        if(i%2)
            result1 += num[i];
        else
            result2 += num[i];
    }

    return result1 * result2;
}

int main(void)
{
    int num[] = {3, 77, 10, 7, 4, 9, 1, 8, 21, 33};

    int num_size = sizeof(num)/sizeof(int);

    printf("result : %d\n", add_mul(num, num_size));
    return 0;
}
~
"que1_4.c" 27L, 404C 1,1 All
```

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ ./debug
result : 5226
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```



# 문제 은행\_79\_6

행렬의 곱셈, 덧셈, 나눗셈, 뺄셈에 대해 조사하시오.  
숫자를 예로 들어서 계산도 해보시오.

# 문제 은행\_79\_6

행렬의 덧셈과 뺄셈의 경우 같은 차원이 아니면 계산되지 않는다.

Ex)  $[2][1]$  와  $[1][2]$ 는 덧셈과 뺄셈이 불가능 하다.

같은 차원일 경우, 같은 위치에 있는 숫자끼리 계산을 한다.

$$\begin{array}{ccccccc} \text{Ex)} & 1 & , & 2 & + & 5 & , & 6 & = & 6 & , & 8 \\ & 3 & , & 4 & & 7 & , & 8 & & 10 & , & 12 \end{array}$$

# 문제 은행\_79\_6

행렬의 나눗셈은 존재하지 않으며, 곱셈의 경우  
앞 행렬의 열과 뒤 행렬의 행의 개수가 일치해야  
계산이 가능하다. 결과값의 형태를 곱하는 두 행렬,  
[열의 길이][행의 길이] 로 만든다.

계산하는 방식은 다음과 같다.

$$\begin{array}{lcl} \text{Ex)} & 2, 3 & * \quad 6, 7 = (2*6 + 3*8), (2*7 + 3*9) = 36, 41 \\ & 4, 5 & \quad 8, 9 \quad (4*6 + 5*8), (4*7 + 5*9) \quad \quad 64, 73 \end{array}$$

# 문제 은행\_104\_2

정수 2004016을 변수에 저장하고 이것을 char형 포인터로 받는다. 그리고, 정수형은 총 4byte 로 구성되므로 총 4개의 byte 를 볼 수 있을것이다.  
각 byte 에 숫자가 어떻게 배치되었는지 확인해보자.

문제 은행\_104\_2

[illegible]

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ ./debug
[1] byte : 48
[2] byte : -108
[3] byte : 30
[4] byte : 0
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```

# 문제 은행\_104\_4

우리는 예제에서 주소값을 교환하여 값을 변경하는 것을 해보았다. 그렇다면 변수 3개를 놓고, 이것에 대해서 무한 Loop 를 돌리면서 저글링을 해보자.

# 문제 은행\_104\_4

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
#include <stdio.h>

int main(void)
{
    int num1 = 10, num2 = 20, num3 = 30;
    int temp = 0;

    while(1)
    {
        temp = num1;
        num1 = num2;
        num2 = num3;
        num3 = temp;

        printf("%d, %d, %d\n", num1, num2, num3);

    }
    return 0;
}
~
~
~
~
~
~
~
~
~
~
"que2_4.c" 19L, 228C
```

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
10, 20, 30
20, 30, 10
30, 10, 20
^C
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```

# 삼각함수

밑변과 높이를 이용하여 넓이를 구하여 보자.



# 삼각함수

밑변과 다른 변이 이루는 각도를 이용하여 넓이를 구하여 보자.

# 문제 은행\_교재\_279\_1

총 7개의 통장을 만들어서 100만원 단위로 최대 500만원까지 입금하였다. 이자율이 연 4% 라고 할 때, 3년 후 각각의 총액을 구하시오.

의문1. 7개의 통장에 100만원 단위의 임의의 값이 들어가는 것인가?

# 문제 은행\_교재\_279\_3

2 by 2 행렬의 곱셈을 계산할 수 있는 프로그램을 만드시오.

# 문제 은행\_교재\_279\_3

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
#include <stdio.h>

void mul(int *num1, int *num2, int *result)
{
    int i, j, k;
    for(i = 0; i < 2; i++)
    {
        for(j = 0; j < 2; j++)
        {
            for(k = 0; k < 2; k++)
            {
                result[i][j] += num1[i][k] * num2[k][j];
            }
        }
    }
}

int main(void)
{
    int i, j;
    int num1[2][2] = {{1, 3}, {5, 4}};
    int num2[2][2] = {{2, 3}, {7, 2}};
    int result[2][2] = {0};

    mul(num1, num2, result);

    for(i = 0; i < 2; i++)
    {
        for(j = 0; j < 2; j++)
        {
            printf("%d\t", result[i][j]);
        }
        printf("\n");
    }

    return 0;
}
"que3_3.c" 39L, 512C      1,1      All
```

```
tewill@tewill-B85M-D3H: ~/my_proj/lesson005
que3_3.c:3:6: note: expected 'int *' but argument is of type 'int (*)[2]'
void mul(int *num1, int *num2, int *result)
^
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ vi que3_3.c
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ ^C
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ vi que3_3.c
tewill@tewill-B85M-D3H:~/my_proj/lesson005$ gcc -g -O0 -o debug que3_3.c
que3_3.c: In function 'mul':
que3_3.c:13:14: error: subscripted value is neither array nor pointer nor vector
    result[i][j] += num1[i][k] * num2[k][j];
               ^
que3_3.c:13:28: error: subscripted value is neither array nor pointer nor vector
    result[i][j] += num1[i][k] * num2[k][j];
                           ^
que3_3.c:13:41: error: subscripted value is neither array nor pointer nor vector
    result[i][j] += num1[i][k] * num2[k][j];
                               ^
que3_3.c: In function 'main':
que3_3.c:27:6: warning: passing argument 1 of 'mul' from incompatible pointer type [-Wincompatible-pointer-types]
    mul(num1, num2, result);
    ^
que3_3.c:3:6: note: expected 'int *' but argument is of type 'int (*)[2]'
void mul(int *num1, int *num2, int *result)
^
que3_3.c:27:12: warning: passing argument 2 of 'mul' from incompatible pointer type [-Wincompatible-pointer-types]
    mul(num1, num2, result);
    ^
que3_3.c:3:6: note: expected 'int *' but argument is of type 'int (*)[2]'
void mul(int *num1, int *num2, int *result)
^
que3_3.c:27:18: warning: passing argument 3 of 'mul' from incompatible pointer type [-Wincompatible-pointer-types]
    mul(num1, num2, result);
    ^
que3_3.c:3:6: note: expected 'int *' but argument is of type 'int (*)[2]'
void mul(int *num1, int *num2, int *result)
^
tewill@tewill-B85M-D3H:~/my_proj/lesson005$
```