

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

5일차 과제 문한나

## 5일차 내용 정리

### - 배열

많은 변수가 필요할 때 이용

선언 방법 : 자료형 배열이름[변수의 길이] = { };

[ ]사이의 수를 index(인덱스)라고 하며 배열의 첫 번째 요소는 [0]에서 시작한다.

100개의 요소를 원하면 int number[100] 그러나 0부터 시작하므로 마지막은 [99]이다

### - 다중 배열

2차원 배열의 경우 행렬의 표현에 용이함

초기화 방법 : [2][3]의 경우 {{1, 2, 3}, {4, 5, 6}}과 같은 방식으로 함

위 경우는 [3]짜리가 2개 배열 형태로 있음을 의미함

- 포인터

주소를 저장할 수 있는 변수

Pointer의 크기는 HW가 몇 bit를 지원 하는냐에 따름

Pointer는 Memory의 어떤 주소값이든 접근할 수 있어야함

고로 Pointer의 크기는 4byte(32bit)

선언방법 : 자료형 \*변수명

- 다중 포인터

Pointer도 Stack에 할당되는 지역변수 즉, Pointer에 대한 주소값도 존재함

주소값은 Pointer에 대한 Pointer로 '\*\*' 형태

- 포인터 배열

포인터를 이용해 배열의 요소에 접근 가능 (배열의 주소값이 정해져 있기 때문)

## - Segmentation Fault

우리가 기계어를 보면서 살펴봤던 주소값들은 사실은 전부 가짜 주소라고 말했었다.

이 주소값은 엄밀하게 가상 메모리 주소에 해당하고 운영체제의 Paging 메커니즘을 통해서 실제 물리 메모리의 주소로 변환된다 (윈도우도 가상 메모리 개념을 베껴서 사용한다)

그렇다면 당연히 맥(유닉스)도 사용함을 알 수 있다.

가상 메모리는 리눅스의 경우 32 비트 버전과 64 비트 버전이 나뉜다.

32 비트 시스템은  $2^{32} = 4\text{GB}$  의 가상 메모리 공간을 가짐

여기서 1:3 으로 1 을 커널이 3 을 유저가 가져간다.

1 은 시스템(HW, CPU, SW 각종 주요 자원들)에 관련된 중요한 함수 루틴과 정보들을 관리

3 은 사용자들이 사용하는 정보들로 문제가 생겨도 그다지 치명적이지 않은 정보들로 구성됨

64 비트 시스템은 1:1 로  $2^{63}$  승에 해당하는 가상메모리를 각각 가진다.

문제는 변수를 초기화하지 않았을 경우 가지게 되는 쓰레기값이 0xCCCCC...CCC 로 구성됨이다.

그러므로 접근하면 안되는 메모리 영역에 접근하였기에 엄밀하게는 Page Fault(물리 메모리 할당되지 않음) 가

발생하게 되고 원래는 Interrupt 가 발생해서 Kernel 이 Page Handler(페이지 제어기)가 동작해서

가상 메모리에 대한 Paging 처리를 해주고 실제 물리 메모리를 할당해주는데

문제는 이것이 User 쪽에서 들어온 요청이므로 Kernel 쪽에서 강제로 기각해버리면서

Segmentation Fault 가 발생하는 것이다.

실제 Kernel 쪽에서 들어온 요청일 경우에는 위의 메커니즘에 따라서 물리 메모리를 할당해주게 된다.

# 문제풀기

문제 1. 배열에 문자열을 입력 받고, 각 배열 요소가 짝수인 경우만을 출력하는 함수를 작성하라.

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
//배열에 문자열을 입력 받고, 배열 요소가 짝수인 경우 출력하는 함수 작성
#include <stdio.h>

int main(void){

    int i;
    char str[] = "tired";

    for(i=0;i<6;i++){
        if(str[i] % 2 == 0)
            printf("%c",str[i]);
    }
    return 0;
}
```

<결과>

```
mhn@mhn-900X3L:~/my_proj/c/5_h$ ./a.out
trdmhn@mhn-900X3L:~/my_proj/c/5_h$
```

문제 2. 아래와 같은 숫자들이 배열에 들어 있다고 가정한다.  
3, 77, 10, 7, 4, 9, 1, 8, 21, 33  
이 요소들을 배열에 거꾸로 집어넣어보자.

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main(){

    int i;
    int arr[] = {3,77,10,7,4,9,1,8,21,33};

    for(i=9;i>=0;i--){
        printf("%d\n",arr[i]);
    }

    return 0;
}
```

<결과>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
mhn@mhn-900X3L:~/my_proj/c/5_h$ vi test2.c
mhn@mhn-900X3L:~/my_proj/c/5_h$ ./a.out
33
21
8
1
9
4
7
10
77
3
mhn@mhn-900X3L:~/my_proj/c/5_h$
```

문제 3. 위의 숫자 3, 77, 10, 7, 4, 9, 1, 8, 21, 33에서

홀수 번째 요소의 합과 짝수 번째 요소의 합을 곱하시오.

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main(){
    int i,even,odd;
    int arr[10] = {3,77,10,7,4,9,1,8,21,33};

    for(i=0;i<=9;i++){
        if((arr[i] % 2) != 0 ){
            odd += arr[i];
        }
    }

    printf("홀수 더한 값:%d\n",odd);

    for(i=0;i<=9;i++){
        if((arr[i] % 2) == 0){
            even += arr[i];
        }
    }

    printf("짝수 더한 값:%d\n",even);

    printf("두 수를 곱한 값:%d\n",even * odd);

    return 0;
}
```

```
홀수 더한 값:151
짝수 더한 값:4196115
두 수를 곱한 값:633613365
mhn@mhn-900X3L:~/my_proj/c/5_h$
```

다시 풀어보겠습니다

문제 4. 행렬의 곱셈, 덧셈, 나눗셈, 뺄셈에 대해 조사하시오.

숫자를 예로 들어서 계산도 해보시오.

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

void add(int arr1[2][2],int arr2[2][2]){

    int res[2][2];
    int i,j;

    for(i=0;i<=1;i++){
        for(i=0;i<=1;j++){
            res[i][j]=arr1[i][j]+arr2[i][j];
        }
    }
    for(i=0;i<=1;i++){
        for(j=0;j<=1;j++){
            printf("%d",res[i][j]);
        }
        printf("\n");
    }
}

void sub(int arr1[2][2],int arr2[2][2]){

    int res[2][2];
    int i,j;

    for(i=0;i<=1;i++){
        for(i=0;i<=1;j++){
            res[i][j]=arr1[i][j]-arr2[i][j];
        }
    }
    for(i=0;i<=1;i++){
        for(j=0;j<=1;j++){
            printf("%d",res[i][j]);
        }
        printf("\n");
    }
}

-- INSERT --
```

```
mhn@mhn-900X3L: ~/my_proj/c/5_h

    }
    printf("\n");
}

void mul(int arr1[2][2],int arr2[2][2]){

    int res[2][2];
    int i,j;

    for(i=0;i<=1;i++){
        for(i=0;i<=1;j++){
            res[i][j]=arr1[i][j]*arr2[i][j];
        }
    }
    for(i=0;i<=1;i++){
        for(j=0;j<=1;j++){
            printf("%d",res[i][j]);
        }
        printf("\n");
    }
}

int main(){

    int arr1[2][2]={2,3},{4,5};
    int arr2[2][2]={3,7},{5,6};

    add(arr1,arr2);
    sub(arr1,arr2);
    mul(arr1,arr2);

    return 0;
}

-- INSERT --
```

다시 풀어보겠습니다

```
mhn@mhn-900X3L:~/my_proj/c/5_h$ ./a.out
Segmentation fault (core dumped)
```



**문제 5.** 정수 2004016을 변수에 저장하고 이것을 char형 포인터로 받는다.  
그리고 정수형은 총 4byte로 구성되므로 총 4개의 byte를 볼 수 있을것이다.

각 byte에 숫자가 어떻게 배치되었는지 확인해보자.

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main(){
    int num[] = "2004016";
    char *p = &num;
    printf("%c\n",p[0]);

    return 0;
}
~
~
~
~
~
~
~
~
~
```

**문제 6.** 우리는 예제에서 주소값을 교환하여 값을 변경하는 것을 해보았다.  
그렇다면 변수 3개를 놓고, 이것에 대해서 무한 Loop를 돌면서 저글링을 해보자!

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main(){

    int num1 = 3,num2 = 4,num3 = 5;
    int *temp = NULL;
    int *num1_p = &num1;
    int *num2_p = &num2;
    int *num3_p = &num3;
    int **num_p_p = &num1_p;
    int **num2_p_p = &num2_p;

    printf("num1_p = %d\n",*num1_p);
    printf("num2_p = %d\n",*num2_p);
    printf("num3_p = %d\n",*num3_p);

    temp = *num_p_p;
    *num_p_p = num2_p;
    *num2_p_p = num3_p;
    num3_p = temp;

    for(;;){
        printf("*num1_p = %d\n", *num1_p);
        printf("*num2_p = %d\n", *num2_p);
        printf("*num3_p = %d\n", *num3_p);
    }

    return 0;
}
```

## <결과>

[illegible]

## 문제 7. 삼각형의 넓이 구하는 문제

case 1) 밑변, 높이 case 2) 밑변, 밑변과 다른 변이 이루는 각도  
2 가지 경우로 모두 구현해본다. (삼각함수 복습합시다 ^^)

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main(){
    int length = 3;
    int heigh = 4;

    printf("%d", (3*4)/2);

    return 0;
}
~
~
~
~
~
~
~
~
~
~
```

<결과>

```
mhn@mhn-900X3L:~/my_proj/c/5_h$ ./a.out
6mhn@mhn-900X3L:~/my_proj/c/5_h$
```

두번째 케이스는 구현을 못했습니다.

문제 8. 총 7개의 통장을 만들어서 100만원 단위로 최대 500만원까지 입금하였다. 이자율이 연 4%라고 할 때, 3년 후 각각의 총액을 구하시오.

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(){

    int bankbook[7];
    int r,i,rate=4,y=3;

    //srand(time(NULL));
    //r = rand() - 난수 범위 지정??

    for(i=0;i<7;i++){
        bankbook[i] += r;
    }

    for(i=0;i<=y;i++){
        bankbook[i]=bankbook[i]*((100+r)/100);
        printf("%d",bankbook[i]);
    }

    return 0;
}
```

난수의 범위를 지정하는 방법을 모르겠습니다

## 문제 9. 2 by 2 행렬의 곱셈을 계산할 수 있는 프로그램을 만드시오

<코드>

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
#include <stdio.h>

int main()
{
    int a[2][2];
    int b[2][2];
    int res[2][2];
    int i,j;

    for(i=0; i<=1; i++)
    {
        for(j=0; j<=1; j++)
        {
            printf("A%d%d 행에 넣을 숫자를 입력하세요.",i+1,j+1);
            scanf("%d",&a[i][j]);
        }
    }

    printf("\n");

    for(i=0; i<=1; i++)
    {
        for(j=0; j<=1; j++)
        {
            printf("B%d%d 행에 넣을 숫자를 입력하세요.",i+1,j+1);
            scanf("%d",&b[i][j]);
        }
    }

    printf("\n");
    printf("행렬 A\n");

    for(i=0;i<=1;i++)
    {
        for(j=0;j<=1;j++)
```

-- INSERT --

23,16-3

```
mhn@mhn-900X3L: ~/my_proj/c/5_h
    }

    printf("\n");
    printf("행렬 A\n");

    for(i=0;i<=1;i++)
    {
        for(j=0;j<=1;j++)
        {
            printf("%d", a[i][j]);
        }

        printf("\n");
    }

    printf("행렬 B\n");

    for(i=0;i<=1;i++)
    {
        for(j=0;j<=1;j++)
        {
            printf("%d", b[i][j]);
        }

        printf("\n");
    }

    for(i=0;i<=1;i++)
    {
        for(j=0;j<=1;j++)
        {
            res[i][j] = a[i][j]*b[i][j];
        }
        printf("\n");
    }

    printf("\n");

    for(i=0;i<=1;i++)
    {
        for(j=0;j<=1;j++)
```

-- INSERT --

```

printf("행렬B\n");

for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        printf("%d", b[i][j]);
    }

    printf("\n");
}

for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        res[i][j] = a[i][j]*b[i][j];
    }
    printf("\n");
}

printf("\n행렬A X 행렬B\n");

for(i=0;i<=1;i++)
{
    for(j=0;j<=1;j++)
    {
        printf("%d", res[i][j]);
    }

    printf("\n");
}

return 0;

```

```

}
-- INSERT --

```

<결과>

```

mhn@mhn-900X3L: ~/my_proj/c/5_h
mhn@mhn-900X3L:~/my_proj/c/5_h$ ./a.out
A11 행에 35를 숫자를 입력하세요 .3
A12 행에 79를 숫자를 입력하세요 .5
A21 행에 35를 숫자를 입력하세요 .7
A22 행에 18를 숫자를 입력하세요 .9

B11 행에 35를 숫자를 입력하세요 .3
B12 행에 79를 숫자를 입력하세요 .5
B21 행에 35를 숫자를 입력하세요 .1
B22 행에 18를 숫자를 입력하세요 .8

행렬A
35
79
행렬B
35
18

행렬A X 행렬B
925
772
mhn@mhn-900X3L:~/my_proj/c/5_h$

```

# 이중 포인터 예제 그림 그리기

```
mhn@mhn-900X3L: ~/my_proj/c/5_s
#include <stdio.h>

int main(){

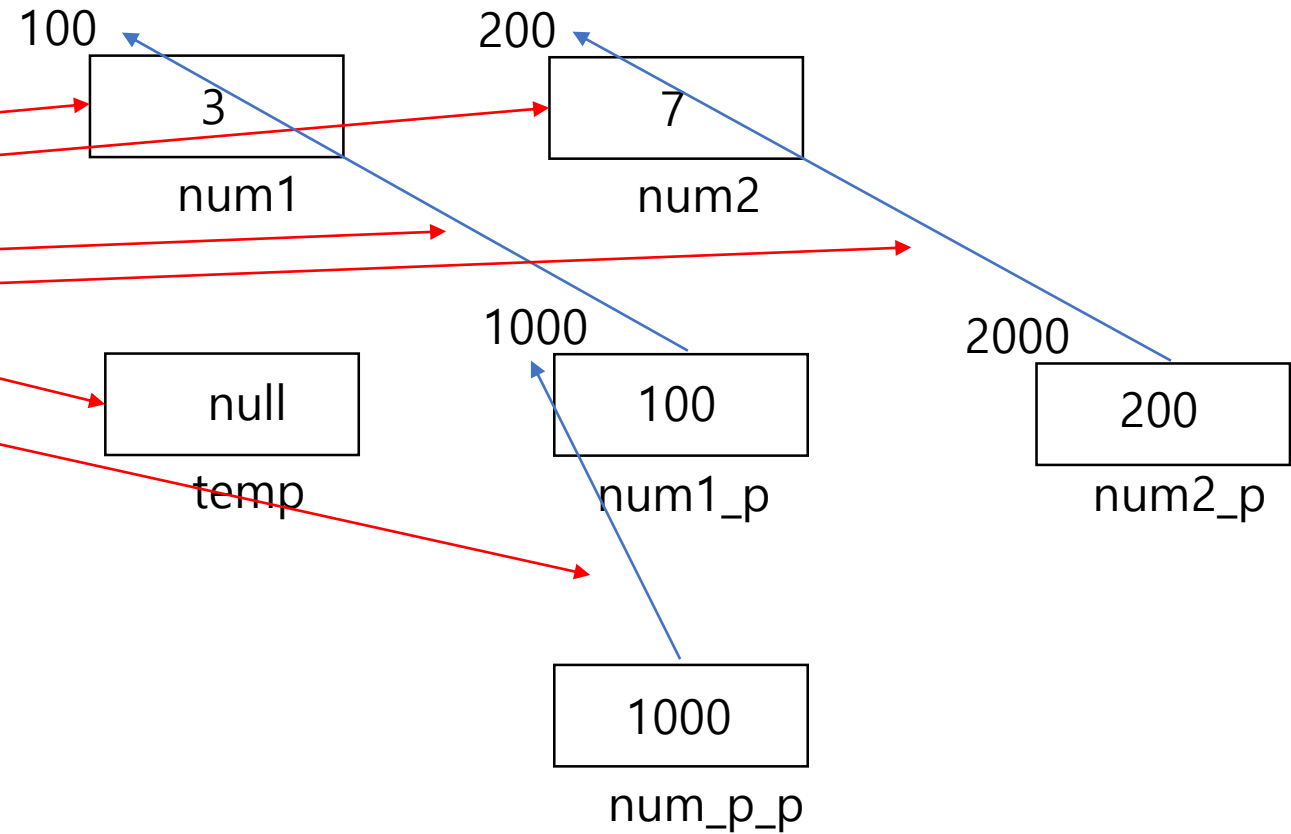
    int num1 = 3,num2 = 7;
    int *temp = NULL;
    int *num1_p = &num1;
    int *num2_p = &num2;
    int **num_p_p = &num1_p;

    printf("num1_p = %d\n",*num1_p);
    printf("num2_p = %d\n",*num2_p);

    temp = *num_p_p;
    *num_p_p = num2_p;
    num2_p = temp;

    printf("*num1_p = %d\n", *num1_p);
    printf("*num2_p = %d\n", *num2_p);

    return 0;
}
"test20.c" 23L, 358C
```



```
mhn@mhn-900X3L: ~/my_proj/c/5_s
#include <stdio.h>

int main(){

    int num1 = 3,num2 = 7;
    int *temp = NULL;
    int *num1_p = &num1;
    int *num2_p = &num2;
    int **num_p_p = &num1_p;

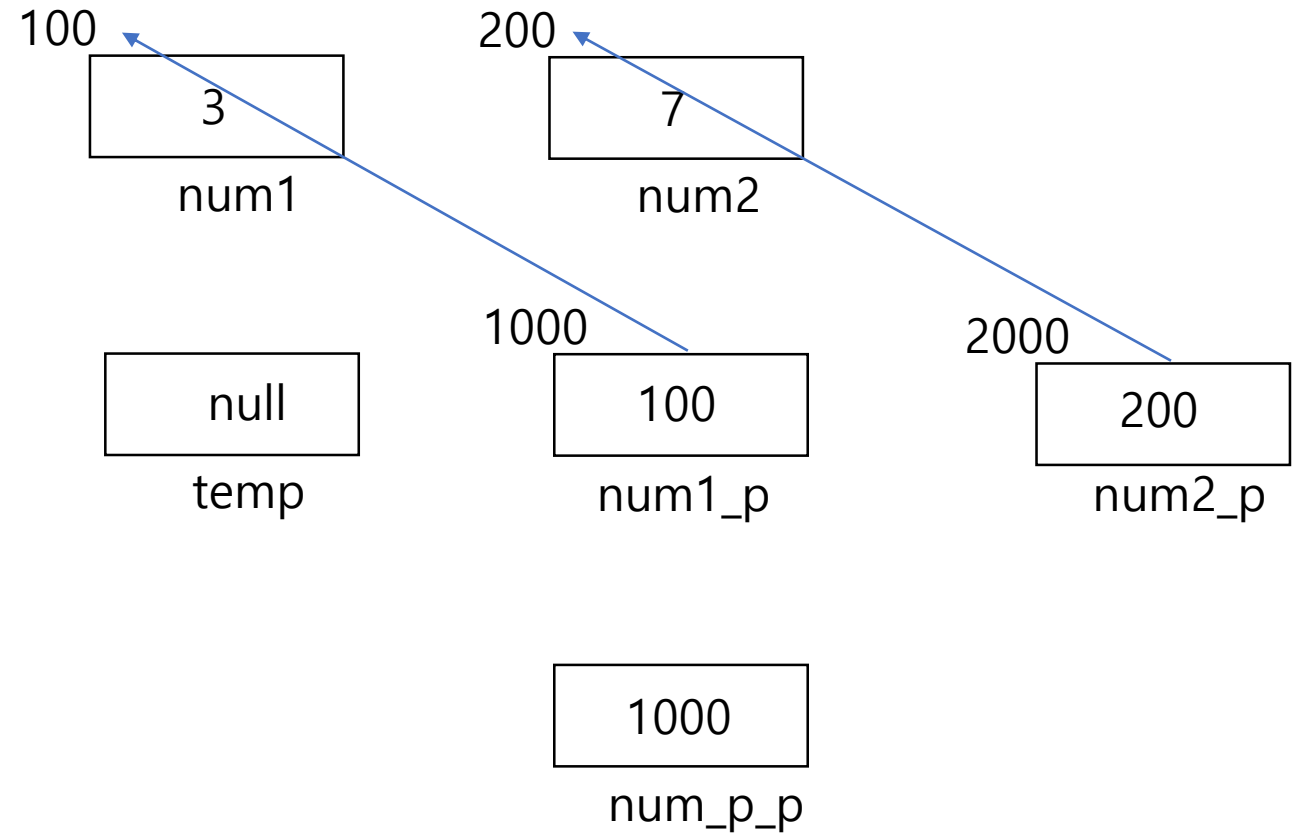
    printf("num1_p = %d\n",*num1_p);
    printf("num2_p = %d\n",*num2_p);

    temp = *num_p_p;
    *num_p_p = num2_p;
    num2_p = temp;

    printf("*num1_p = %d\n", *num1_p);
    printf("*num2_p = %d\n", *num2_p);

    return 0;
}
"test20.c" 23L, 358C
```

```
num1_p = 3
num2_p = 7
```





```
mhn@mhn-900X3L: ~/my_proj/c/5_s
#include <stdio.h>

int main(){

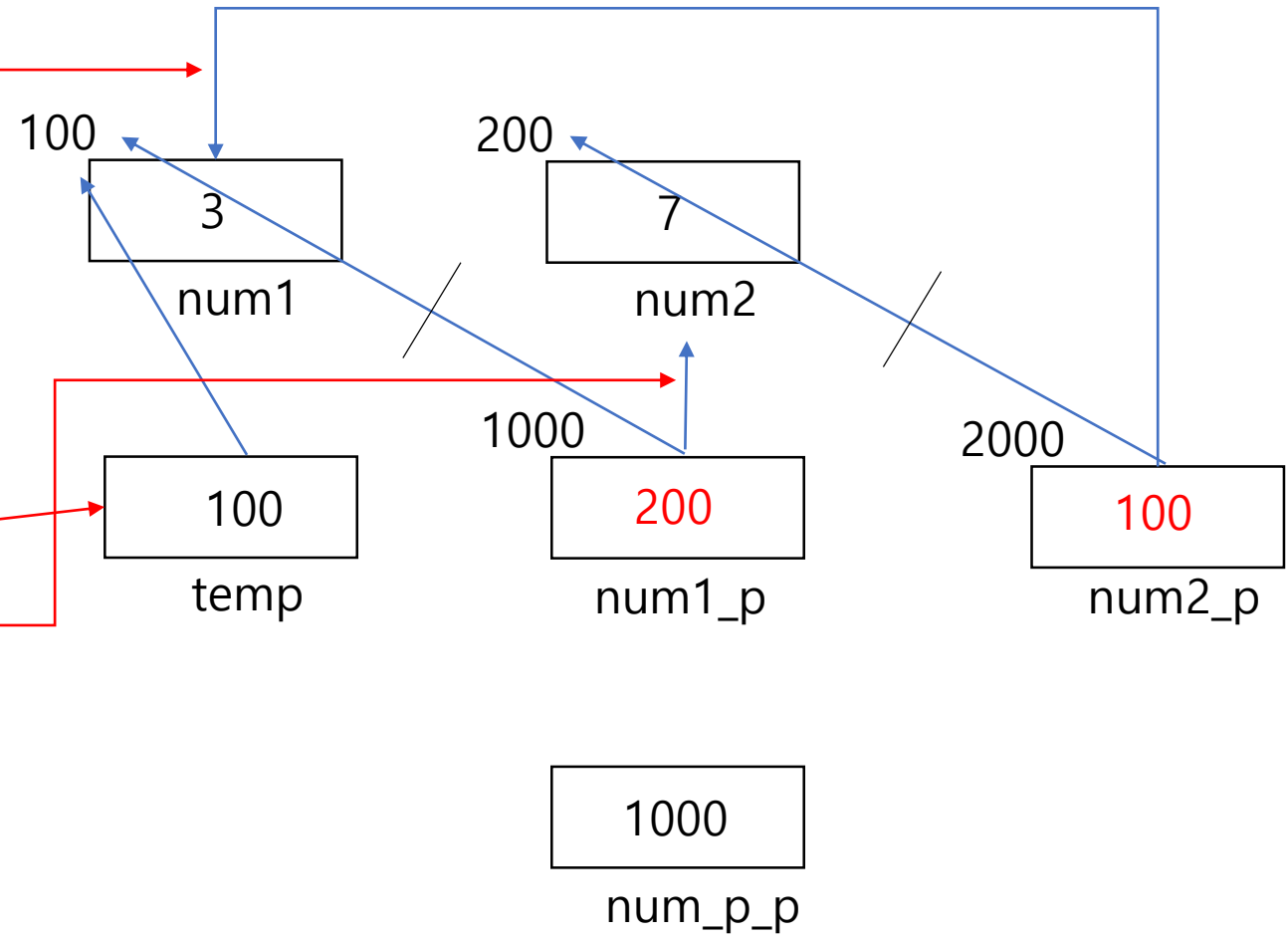
    int num1 = 3,num2 = 7;
    int *temp = NULL;
    int *num1_p = &num1;
    int *num2_p = &num2;
    int **num_p_p = &num1_p;

    printf("num1_p = %d\n",*num1_p);
    printf("num2_p = %d\n",*num2_p);

    temp = *num_p_p;
    *num_p_p = num2_p;
    num2_p = temp;

    printf("*num1_p = %d\n", *num1_p);
    printf("*num2_p = %d\n", *num2_p);

    return 0;
}
"test20.c" 23L, 358C
```



```
*num1_p = 7
*num2_p = 3
```