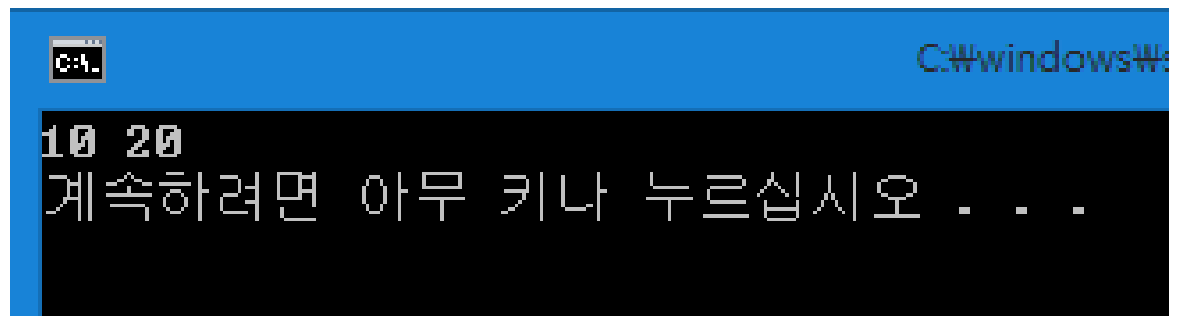


프로그래밍 언어 실습 06

8주차

포인터

- 예제1)
- 포인터 변수 `ptr1`이 `num2`를 가리키게 하고, 포인터 변수 `ptr2`가 `num1`을 가리키게 하라.
 - Int형 변수 `num1`과 `num2`를 선언과 동시에 10과 20으로 초기화한다.
 - Int형 포인터 변수 `ptr1`과 `ptr2`를 선언하여 각각 `num1`과 `num2`를 가리킨다.
 - 포인터 변수 `ptr1`과 `ptr2`를 이용하여 `num1`을 10증가시키고, `num2`를 10감소시켜라.
- 출력화면예시)



```
C:\windows#  
10 20  
계속하려면 아무 키나 누르십시오 . . .
```

포인터

- 예제1)

```
#include <stdio.h>

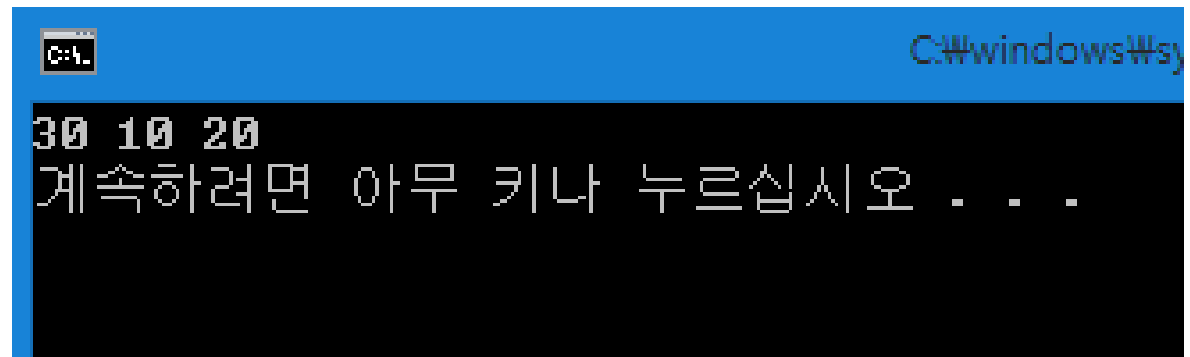
int main( )
{
    int num1 = 10, num2 = 20;
    int *ptr1 = &num1;
    int *ptr2 = &num2;
    int *temp;
    (*ptr1)+=10;
    (*ptr2)-=10;

    temp = ptr1;
    ptr1 = ptr2;
    ptr2 = temp;

    printf( "%d %d #n", *ptr1, *ptr2);
    return 0;
}
```

포인터와 함수

- 예제2)
- 세 변수에 저장된 값을 서로 바꾸는 함수를 정의하여라.
 - 예) `swap3(&num1, &num2, &num3)`의 형태로 함수가 출력되어야 함
 - 함수호출의 결과로 `num1`에 저장된 값은 `num2`에 저장
 - `Num2`에 저장된 값은 `num3`에 저장
 - `Num3`에 저장된 값은 `num1`에 저장
- 출력화면예시)



```
C:\windows#sy
30 10 20
계속하려면 아무 키나 누르십시오 . . .
```

포인터와 함수

- 예제2)

```
#include <stdio.h>

void swap3(int *ptr1, int *ptr2, int *ptr3)
{
    int temp = *ptr3;
    *ptr3 = *ptr2;
    *ptr2 = *ptr1;
    *ptr1 = temp;
}

int main()
{
    int num1 = 10, num2 = 20, num3 = 30;
    swap3(&num1, &num2, &num3);

    printf("%d %d %d \n", num1, num2, num3);
    return 0;
}
```

다차원 배열

- 예제3)
- 가로 길이가 9, 세로 길이가 3인 int형 2차원 배열을 선언하여 구구단 중 2단, 3단, 4단을 저장하여라.
- 출력화면예시)

```
C:\#windows#s  
2 4 6 8 10 12 14 16 18  
3 6 9 12 15 18 21 24 27  
4 8 12 16 20 24 28 32 36  
계속하려면 아무 키나 누르십시오 . . .
```

다차원 배열

- 예제3)

```
#include <stdio.h>

int main( )
{
    int arr[3][9];
    int i, j;

    for(i=0; i<3; i++)
        for(j=0; j<9; j++)
            arr[i][j] = (i+2) * (j+1);

    for(i=0; i<3; i++)
    {
        for(j=0; j<9; j++)
            printf( "%4d", arr[i][j] );
        printf( "\n" );
    }
    return 0;
}
```

Call-by-Value / Call-by-Reference

- 예제4)
- 변수 `num`에 저장된 값의 제곱을 계산하는 함수를 정의하고, 이를 호출하는 `main` 함수를 작성하여라.
 - Call-by-Value 기반의 `squareByValue` 함수
 - Call-by-Reference 기반의 `squareByReference` 함수
 - `squareByValue` 함수는 인자로 전달된 값의 제곱을 반환해야 함
 - `squareByReference` 함수는 정수가 저장되어 있는 변수의 주소 값을 인자로 받아서 해당 변수에 저장된 값의 제곱근을 그 변수에 다시 저장한다.
- 출력화면 예시)

```
100
100
계속하려면 아무 키나 누르십시오 . . .
```


Call-by-Value / Call-by-Reference

- 예제4)

```
#include <stdio.h>

int squareByValue(int num)
{
    return num*num;
}

void squareByReference(int*ptr)
{
    int num = *ptr;
    *ptr = num*num;
}

int main()
{
    int num = 10;
    printf("%d #n", squareByValue(num));
    squareByReference(&num);
    printf("%d #n", num);
    return 0;
}
```

malloc 함수

- 예제5)
- 그림과 같은 결과를 출력하도록 소스 코드를 작성하여라.
 - 입력 받은 n만큼 malloc 함수를 사용하여 동적 할당
 - 값을 입력 받아 저장할 때 배열 사용
 - Ex) scanf("%d", &arry[k]);
- 출력화면예시)

```
몇개의 숫자 데이터를 입력하실건가요? : 5
차례대로 입력하세요.
3 4 8 1 9
입력된 데이터는 3 4 8 1 9 입니다
```

malloc 함수

- 예제5)

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

void main()
{
    int x, num; int *ary;
    printf("몇개의 숫자 데이터를 입력하실건가요? : ");
    scanf("%d", &num);
    ary = (int *)malloc(sizeof(int)*num); //입력받은 n만큼 동적할당
    printf("차례대로 입력하세요.\#n");
    for (x = 0; x < num; x++)
    {
        scanf("%d", &ary[x]);
    }
    printf("입력된 데이터는 ");
    for (x = 0; x < num; x++)
    {
        printf("%d ", ary[x]);
    }
    printf("입니다.\#n");
}
```

Q & A