

고급C프로그래밍 및 설계

1주차_월요일

Kwangwoon Univ.
Dept. of Computer engineering
Ki-Hoon Lee



Syllabus

● Professor

- 이기훈 (kihoonlee@kw.ac.kr), 940-8674, 비마관 508호

● Evaluation

- 출석 10%, 중간고사 30%, 기말고사 30%, 과제보고서 30%
- 지각 2회 = 결석 1회. (8회 이상 결석 시 F처리)
- 평가비율이 변경될 수 있음.

● Textbook

- 주교재: 열혈 C 프로그래밍, 윤성우, 오렌지미디어
- 부교재: C로 쓴 자료구조론 (2nd Edition), 이석호, 교보문고
- 부교재: Practical C Pointer, 김원선, 이한출판사

실습 운영 계획

- 실습시간에 나가는 모든 문제는 과제 (1주차 월요일 수업 제외)

- Softcopy

- 소스파일(.c)과 보고서만 압축해서 제출
- 압축파일 이름 : ‘고급C프_1주차_학번_이름.zip’
- 보고서에 고찰 및 문제 별 코드, 결과화면 포함할 것
- 주석은 최대한 작성할 것
- 문제에 있는 조건을 지키지 않을 시 감점
- Copy 발견 시, 해당 주차 과제 0점 처리

- 제출 방법

- U-Campus 로그인 -> 온라인 참여 학습 -> 과제 제출
- Hardcopy는 제출하지 않음



조건문

```
void main(){
    int num=5;

    if(num%2==0){
        printf("%d는 짝수입니다\n", num);
    }else{
        printf("%d는 홀수입니다\n", num);
    }
}
```

if 조건문



5는 홀수입니다



```
void main(){
    int num=5;

    printf("%d는 %s입니다\n", num, (num%2==0? "짝수": "홀수") );
}
```

삼항연산자 조건문

if ~

```
if(num>0)
    printf("%d는 양의정수\n", num);
```

else if ~

```
else if(num<0)
    printf("%d는 음의정수\n", num);
```

else ~

```
else
    printf("%d는 0\n", num);
```

equal: ==

and: &&

or: ||

if(0<num<10)

if(0<num && num<10)





반복문

특정 조건을 만족할 때까지 반복

-> 주로 while문 사용

```
void main(){  
    int num=0;  
  
    while(num != -1){  
        scanf("%d", &num);  
    }  
}
```



```
5  
4  
-1
```

원하는 횟수만큼 반복

-> 주로 for문 사용

```
void main(){  
    int num=0;  
    int i=0;  
  
    for(i=0; i<3; i++){  
        printf("%d\n", i);  
    }  
}
```



```
0  
1  
2
```

[for문 기본 문법]

for(i=0; i<반복횟수; i++)

ex) 10번 반복 for(i=0; i<10; i++)



함수 & 지역변수

```
void func1(){
    printf("func1 함수 호출\n");
}

int sum(int n){
    int i=0, sum=0;
    for(i=1; i<=n; i++){
        sum=sum+i;
    }
    return sum;
}

void main(){
    func1();
    printf("1부터 %d까지의 합: %d\n", 10, sum(10));
}
```



```
func1 함수 호출
1부터 10까지의 합: 55
```

```
void func(int num){
    num=5;
    printf("num: %d\n", num);
}

void main(){
    int num=3;

    printf("num: %d\n", num);
    func(num);
    printf("num: %d\n", num);
}
```



```
num: 3
num: 5
num: 3
```



문자열 출력

```
void main(){  
    char str[]="Hello World!";  
    int i=0;  
  
    while(str[i] != NULL)  
        printf("%c", str[i++]);  
  
    printf("\n");  
    puts(str);  
}
```



```
Hello World!  
Hello World!
```



2차원 배열

```
void main(){  
    int num[5][3];  
    int i=0, j=0;  
  
    for(i=0; i<5; i++){  
        for(j=0; j<3; j++){  
            num[i][j]=i+j;  
        }  
    }  
  
    for(i=0; i<5; i++){  
        for(j=0; j<3; j++){  
            printf("%d ", num[i][j]);  
        }  
        printf("\n");  
    }  
}
```



0	1	2
1	2	3
2	3	4
3	4	5
4	5	6



포인터

```
void main(){  
    int a=5;  
    int *pA = &a;  
  
    printf("a: %d\n", a);  
    printf("&a: %p\n", &a);  
  
    printf("pA: %p\n", pA);  
    printf("*pA: %d\n", *pA);  
    printf("&pA: %p\n", &pA);  
}
```



```
a: 5  
&a: 0030FEE8  
pA: 0030FEE8  
*pA: 5  
&pA: 0030FEDC
```

0030FEE8

a 5 *pA

0030FEDC

pA &a



실습 1

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 숫자2가 숫자1의 약수인지 판별

결과화면

```
숫자1 입력 : 9  
숫자2 입력 : 3  
3은<는> 9의 약수입니다
```



실습 2

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 입력한 수의 모든 약수 출력
2. for문 사용

결과화면

```
숫자 입력: 12
12의 약수: 1 2 3 4 6 12
```



실습 3

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 실습2에 약수들의 합을 출력하는 코드 추가

결과화면

```
숫자 입력: 12  
12의 약수: 1 2 3 4 6 12  
약수들의 합: 28
```



실습 4

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 실습3의 코드에서 약수는 출력하고, 약수들의 합은 리턴하는 코드를 함수로 작성
 2. 함수에서 약수는 출력, 약수들의 합은 return
- ※ 함수 형태: int 함수명(int 변수명) {약수 출력..., return 약수들의 합 }

결과화면

```
숫자 입력: 12
12의 약수: 1 2 3 4 6 12
약수들의 합: 28
```

main함수 예

```
printf("숫자 입력: ");
scanf("%d", &num);
```

```
printf("약수들의 합: %d\n", div_sum(num));
```

실습 5

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 실습4에 완전수 판별 코드 추가
 2. 약수 출력은 삭제
 3. 약수들의 합은 자신을 제외한 약수들의 합으로 변경
- ※ 완전수: 자신을 제외한 약수들의 합이 자신과 같은 수 (양의 정수)

결과화면

```
숫자 입력: 12
자신을 제외한 약수들의 합: 16
12은(는) 완전수가 아닙니다
```

```
숫자 입력: 28
자신을 제외한 약수들의 합: 28
28은(는) 완전수 입니다
```



실습 6

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 지금까지 작성한 코드를 참고하여 1부터 입력한 숫자 사이의 모든 완전수를 출력하는 코드 작성

결과화면

```
숫자 입력: 10000  
완전수(1 ~ 10000): 6 28 496 8128
```

실습 7

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 크기가 5인 int형 배열에 사용자의 입력값을 저장
2. 배열에 입력된 정수 중 가장 큰 정수, 가장 작은 정수를 출력

결과화면

```
5개의 정수 입력: 5 3 1 2 4  
Max: 5  
Min: 1
```




실습 8

□ 다음 결과화면과 같이 출력되도록 프로그램을 구현하시오.

[조건]

1. 실습7의 코드에서 Max, Min을 찾는 코드를 포인터를 이용한 코드로 변경
2. 포인터는 Max, Min값을 가리켜야 함

결과화면

```
5개의 정수 입력: 5 3 1 2 4
Max: 5
Min: 1
```

참고

int *pMax;

pMax = 가리키는 변수의 주소;

ex) pMax = # pMax = &arr[0];

가리키는 변수의 값: *pMax