

<LAB 3>

전처리와 입출력

제출일 : 2022/09/23

이름 : 고려욱

학번 : 201810528

1. Lab 3-9)

1) 실습 문제

다음 소스를 실행해 보고 어떻게 그런 결과가 나왔는지 자세히 설명하시오

```
#include <stdio.h>
void main(void) {

    char c;
    unsigned char cc;

    c = -1;
    cc = -1;
    printf("c = %d\n", c);
    printf("cc = %d\n\n\n\n", cc);

    short d; unsigned short dd;

    d = -1;
    dd = -1;
    printf("d = %d\n\n", d);
    printf("dd = %d\n", dd);

    return 0;
}
```

2) 배경 지식

[1] printf()함수

기본적인 문자열을 출력하는 함수의 일종이다. Printf("<출력내용>",<출력목록>);와 같이 사용한다. 출력내용 안에는 출력과 형식지정자 등이 사용된다. 출력 목록에는 출력내용 안에 있는 출력지정자에 해당하는 변수명 또는 연산식이 들어가 출력한다.

[2] 형식 지정자

형식지정자 또는 형식문자는 원하는 옵션의 형식을 출력하기 위해 사용하는 문자이다. 문자는 %c, 문자열은 %s, 정수형은 %d, 팔진수는 %o, 16진수는 %x, 실수형은 %f, 지수형은 %e로 사용한다.

[3] 오버플로와 언더플로

자료형에는 각 자료형에 따라 저장할 수 있는 범주가 존재한다. 저장할 수 있는 범주 외의 값이 변수에 저장되면 오버플로와 언더플로가 발생한다. 오버플로가 자료형 범주의 최대값에서 벗어난 값이 저장되면 범주의 최소값이 저장된다. 반대로 언더플

로는 범주의 최소값에서 벗어난 값이 저장되면 범주의 최대값이 저장된다.

3) 소스 코드

```
#include <stdio.h>

void main(void) {
    char c;
    unsigned char cc;

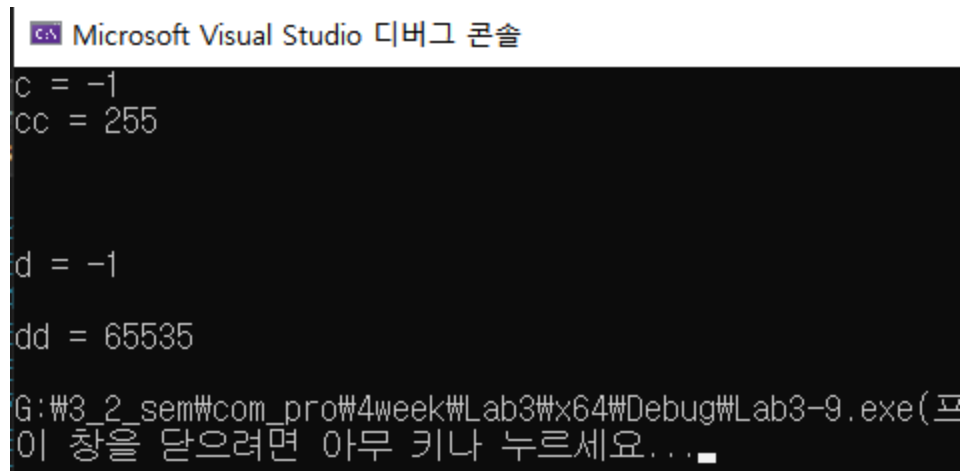
    c = -1;
    cc = -1;
    printf("c = %d\n", c);
    printf("cc = %d\n\n\n\n", cc);

    short d; unsigned short dd;

    d = -1;
    dd = -1;
    printf("d = %d\n\n", d);
    printf("dd = %d\n", dd);

    return 0;
}
```

4) 실행결과



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```
c = -1
cc = 255

d = -1
dd = 65535

G:\#3_2_sem\com_pro\4week\Lab3\64\Debug\Lab3-9.exe(프
이 창을 닫으려면 아무 키나 누르세요...)
```

5) 결과분석

먼저, c와 cc는 문자형 변수 char과 unsigned char로 선언되었다. 변수 c의 경우 signed char로 -127~127범주안에 해당하는 값을 저장할 수 있어 c에 저장된 -1값이 출력 지정자 %d에 의해 정수형 표현인 -1로 출력된다. 이후 \n에 의해 한줄이 내려간 곳에서 다음 출력이 일어난다.

변수cc의 경우 unsigned char로 선언되어 양수값만 저장할 수 있는 변수이다.

Unsigned char는 0~255의 값이 저장될 수 있는데 현재 -1이 저장되어 Underflow가 일어나게 된다. 이에 의해 cc에는 최대값인 255가 저장되게 되고 printf함수 statement에 의해 정수형으로 255로 출력된다. 이후 \n이 4개가 존재하여 4줄이 내려간 곳에 다음 출력이 진행된다.

변수 d와 dd는 short, unsigned short로 선언되어 short형태의 값이 정수형으로 저장된다. 변수 d는 short로 선언되어 signed형태로 -32768~32768 범주에 해당하는 양수, 음수 모두 저장될 수 있다. 따라서, d에는 -1이 그대로 저장되며 printf함수 statement에 의해 정수형으로 -1이 출력된다. 이후 \n이 2개가 존재하여 2줄이 내려간 곳에 다음 출력이 진행된다.

마지막으로 dd는 unsigned short로 선언되어 0~65535 범주안의 양수값만 저장이 가능하다. 따라서 dd에 -1이 저장되지 않고 언더플로에 의해 범주의 최대값인 65535이 저장된다. 이후 printf statement에 의해 정수형으로 65535가 출력되며 이후 \n에 의해 한줄이 내려간 곳에 커서가 위치하고 프로그램이 종료된다.

2. Lab 3-10)

1) 실습 문제

- 표준입력으로 원의 반지름 값인 실수 하나를 입력 받아 다음조건을 만족하는 프로그램을 작성하시오
- 매크로 상수로 원주율 3.141592 정의
- 원 반지름 출력 , 원 면적 출력 : πr^2 , 원 둘레 출력 : $2\pi r$

2) 배경 지식

[1] 매크로 상수

매크로 상수는 #define <이름> <원하는 형태>로 사용한다. 전처리기 지시자 #define은 main함수를 찾아 실행하기 전에 #define에 해당하는 변수, 함수 등을 미리 지정한 것으로 변환해 준다. 주로 대문자 이름으로 정의한다.

[2] scanf()함수

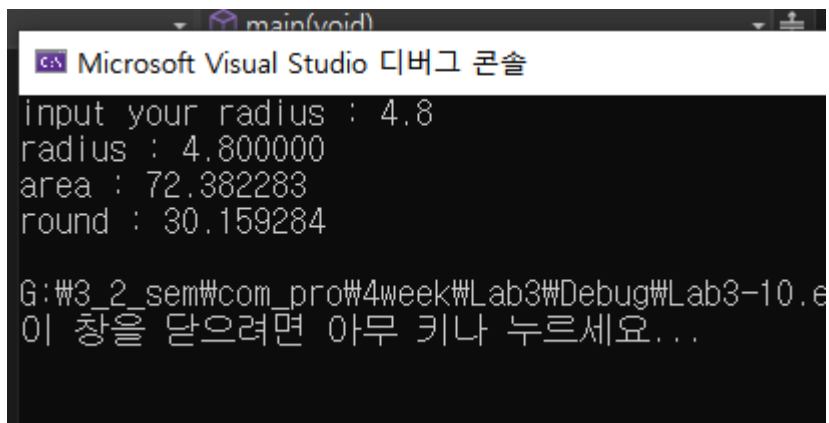
대표적인 입력함수이다. Scanf("<형식지정자>", 저장할 변수의 주소)로 사용하며 형식지정자를 사용하여 입력을 진행한다. 형식지정자에 입력 및 저장을 원하는 형식지정자를 사용하며 해당 입력을 저장하는 변수를 사용한다. 이때 변수의 주소를 뒤에 입력해야하므로 변수앞에 주소연산자 &를 붙여 사용해야 한다.

3) 소스 코드

```
#pragma warning(disable:4996)
#include <stdio.h>
#define PI 3.141592

int main(void)
{
    float r;
    printf("input your radius : ");
    scanf("%f", &r);
    printf("radius : %f\n", r);
    printf("area : %f\n", r * r * PI);
    printf("round : %f\n", 2 * PI * r);
    return 0;
}
```

4) 실행결과



```
main(void)
Microsoft Visual Studio 디버그 콘솔
input your radius : 4.8
radius : 4.800000
area : 72.382283
round : 30.159284

G:\#3_2_sem\com_pro\#4week\Lab3\Debug\Lab3-10.e
이 창을 닫으려면 아무 키나 누르세요...
```

5) 검토의견

- visual studio 상에서 scanf사용시 보안상의 문제로 오류 발생
- pragma warning(disable:4996)을 사용하여 해당 오류 무시하는 코드로 컴파일 진행

3. Lab 3-11)

1) 실습 문제

표준입력으로 4자리 정수를 하나 입력 받아 다음 조건을 만족하는 프로그램을 작성하시오

- 만일 입력이 6527이라면 다음과 같은 입출력이 되도록
- 모두 전체폭은 10

```
4자리 정수 입력: 6527
      6527
0000006527
+000006527
6527
      14577
0000014577
      197f
0x0000197f
```

2) 배경 지식

[1] 정수의 형식지정자

기본적으로 정수를 출력하기 위해서는 10진수는 %d, 8진수는 %o, 16진수는 %x를 사용한다. 이때 추가적인 옵션으로 출력시의 출력폭을 지정해 줄 수 있다. 형식지정자에서 %와 알파벳 사이에 원하는 길이의 출력폭을 넣어 출력을 진행할 수 있다. 또한 출출력폭 앞에 +/-0/#을 사용하여 각각 +기호 출력/ 좌측정렬/ 남아있는 폭을 0으로 채움/ 8진수,16진수의 경우 0또는 0x를 채움 의 옵션을 붙여 출력할 수 있다.

3) 소스 코드

```
#pragma warning(disable:4996)
#include <stdio.h>

void main(void)
{
    int num;
    printf("4자리 정수 입력 : ");
    scanf("%d", &num);
```

```

    printf("%10d\n", num);
    printf("%010d\n", num);
    printf("%+010d\n", num);
    printf("%-10d\n", num);
    printf("%10o\n", num);
    printf("%010o\n", num);
    printf("%10x\n", num);
    printf("%#010x\n", num);
}

```

4) 실행결과

```

4자리 정수 입력 : 6527
6527
0000006527
+0000006527
6527
14577
0000014577
197f
0x0000197f

G:\#3_2_sem\com_pro\4week\Lab3\64\Debug\Lab3-11.exe
이 창을 닫으려면 아무 키나 누르세요...

```

5) 검토의견

- 옵션지정자들의 순서는 상관없다. 예로, "%+010d\n"과 "%0+10d\n"는 같은 결과로 출력된다.

4. Lab 3-12)

1) 실습 문제

- (1) 소스 putchar.c를 실행해 보고 결과를 설명하시오.

```

1 // file: putchar.c
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     char a = '\0', b;
8
9     puts("문자 하나 입력:");
10    a = getchar();
11    putchar(a); putchar('\n');
12
13    return 0;
14 }
15

```

(2) 소스 putchar1.c를 실행해 보고 어떻게 그런 결과가 나왔는지 설명하시오.

```

1 // file: putchar1.c
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     char a = '\0', b;
8     puts("문자 하나 입력:");
9     a = getchar();
10    putchar(a); putchar('\n');
11
12    puts("문자 두개 입력:");
13    a = getchar(); b = getchar();
14    putchar(a); putchar(a);
15    putchar(b); putchar('\n');
16    return 0;
17 }
18

```

2) 배경 지식

[1] putchar()&getchar() 함수

Putchar()와 getchar()는 문자의 입출력 함수이다. getchar()함수는 문자 하나를 입력받는 함수이며 putchar()함수는 반대로 문자 하나를 출력하기 위한 함수이다. A = getchar()으로 사용하며 괄호안에 인자 없이 함수를 호출하여 사용한다. 반대로 putchar('A')식으로 putchar는 지정된 변수의 값을 가져와 문자로 출력한다.

3) 소스 코드

[1]

```

#include <stdio.h>

int main(void)
{
    char a = '\0', b;

    puts("문자하나 입력 : ");
    a = getchar();
    putchar(a);
    putchar('\n');

    return 0;
}

```

[2]

```

#include <stdio.h>

int main(void)

```



```

{
    char a = 'w0', b;

    puts("문자하나 입력 :");
    a = getchar();
    putchar(a);
    putchar('Wn');

    puts("문자 두개 입력 :");
    a = getchar();
    b = getchar();
    putchar(a);
    putchar(a);
    putchar(b);
    putchar('Wn');
    return 0;
}

```

4) 실행결과

[1]

```

Microsoft Visual Studio 디버그 콘솔
문자하나 입력 :
k
k
G:\3_2_sem\com_pro\4week\Lab3\64\Debug\Lab3-12
이 창을 닫으려면 아무 키나 누르세요...

```

putchar.c의 경우 먼저 char형 변수 a, b를 선언하고 a에 공백에 해당하는 "w0"값을 저장하였다. 이후 getchar()함수에 의해 입력된 문자 k가 변수a에 저장되고 putchar를 통해 콘솔창에 변수 a에 저장된 값이 출력된다. 따라서, 입력을 위해 사용자가 입력한 k가 한번, 이후에 putchar출력 k가 한번, putchar에 의해 줄바꿈이 한번 출력되고 해당 프로그램이 마무리된다.

[2]

```

Microsoft Visual Studio 디버그 콘솔
문자하나 입력 :
a
문자 두개 입력 :
ab
a
G:\3_2_sem\com_pro\4week\Lab3\64\Debug\Lab3-12
이 창을 닫으려면 아무 키나 누르세요...

```

Putchar1.c의 경우 putchar.c와 다르게 이후에 문자 두개를 입력받는다. 문자를 2개 입력 받기전에 입력버퍼에 문자 한 개 입력시에 입력된 "\0"이 저장되어 있다. 문자 2개 입력 이후 입력버퍼에서 입력의 순서대로 꺼내져 저장된다. 따라서, a에는 'wn'이 저장되고 b에는 'a'가 저장된다. 따라서, putchar함수를 이용해 출력시 a에 저장된 줄바꿈 2개가 출력되고 이후 b에저장된 'a'가 출력되며 마지막으로 줄바꿈 1개가 출력되며 프로그램이 종료된다.

5) 검토의견

- 입력버퍼에 대한 이해가 필요, 한문자 입력할 때 한문자가 아니라 여러문자를 입력하면 두문자 입력 출력 이후에 입력을 하지 않고 계속 입력버퍼에 저장된값이 사라질 때까지 출력이 진행됨

5. Lab 3-13)

1) 실습 문제

표준입력으로 문자 하나를 입력 받아 다음 조건을 만족하는 프로그램을 작성하시오

- 함수 getchar()로 문자 하나 입력 받음
- 함수 putchar()와 printf()로 문자 그대로 출력
- 문자의 코드 값을 십진수, 십육진수로 출력

2) 배경 지식

[1] 아스키 코드

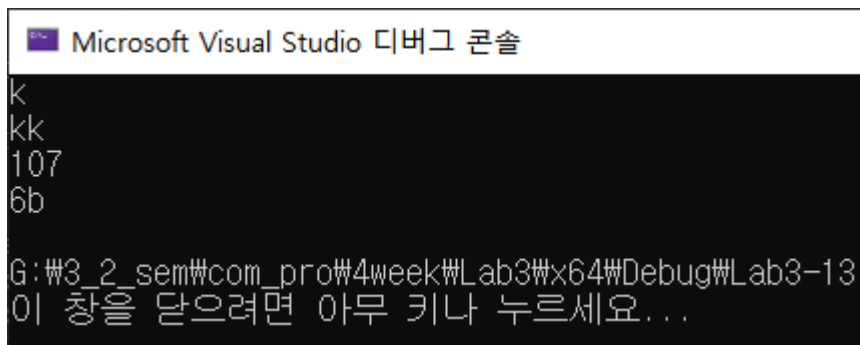
아스키 코드는 C언어에서 문자형 자료공간에 저장되는 값이다. 문자형 자료공간에 문자가 입력되면 실제로는 정수값이 입력된다. 이때 저장되는 정수값은 아스키 코드 표에 의해 정해진 정수값이 저장된다. 아스키코드는 ANSI(American National Standards Institute)에서 제정한 정보 교환용 표준 코드로 총 127개의 문자로 구성되어 있다. 특수문자를 포함하여 숫자, 영어 소문자, 영어 대문자가 지정되어 있다.

3) 소스 코드

```
#pragma warning(disable:4996)
#include <stdio.h>

int main(void)
{
    char a = getchar();
    putchar(a);
    printf("%c\n", a);
    printf("%d\n", a);
    printf("%x\n", a);
    return 0;
}
```

4) 실행결과



```
Microsoft Visual Studio 디버그 콘솔
k
k
107
6b

G:\3_2_sem\com_pro\4week\Lab3\64\Debug\Lab3-13
이 창을 닫으려면 아무 키나 누르세요...
```

5) 검토의견

- putchar는 형식지정자가 필요없이 문자 그대로 출력, printf는 문자출력을 위해 형식 지정자 %c를 사용해야함
- 영어 소문자 'k'의 경우 아스키코드에 의해 10진수로 107이다. 따라서, 10진수 출력시 %d를 사용하여 107을 출력하고 16진수 출력시 %x를 사용하여 6b가 출력된다.