

7장 반복문 연습문제 풀이

01 다음 중 틀린 문장을 지적하고 올바르게 수정하라.

- ① 조건식의 값이 1일 때만 참으로 간주된다.
- ② 반복 루프가 중첩되어 있는 경우, break는 하나의 반복 루프만 벗어 날 수 있다.
- ③ do...while 문에서 조건식의 값이 거짓이면 한 번도 수행되지 않는다.
- ④ for 문에서 초기식, 조건식, 증감식이 전부 비어 있으면 안 된다.
- ⑤ for 문안에 다른 for 문이 들어갈 수 있다.

1.

- ① 조건식의 값이 1일 때만 참으로 간주된다.-> 조건식의 값이 0이 아니면 참으로 간주된다.
- ③ do...while 문에서 조건식의 값이 거짓이면 한 번도 수행되지 않는다.-> 거짓이라고 하더라도 한번은 수행된다.
- ④ for 문에서 초기식, 조건식, 증감식이 전부 비어 있으면 안 된다.->비어 있어도 된다.

02 다음은 무한 반복을 구현한 소스이다. 올바르게 구현된 것을 모두 골라보자.

- ① while(1) {} ② for(1) {} ③ for(; 1 ;) {} ④ for(; ;) {}

2. ① while(1) {} ③ for(; 1 ;) {} ④ for(; ;) {}

03 다음의 프로그램을 실행시키면 "Hello World!"는 몇 번이나 출력되는가?

```
#include <stdio.h>
int main()
{
    int x;
    for (x = 0; x < 10; x++)
    {
        if (x > 5)
            continue;
        if (x > 8)
            break;
        printf("Hello World! \n");
    }
    return 0;
}
```

```
int x;
for (x = 0; x < 10; x++) {
    if (x > 5)
        continue;
    if (x > 8)
        break;
    printf("x=%d Hello\n", x);
}
```

x=0 Hello
x=1 Hello
x=2 Hello
x=3 Hello
x=4 Hello
x=5 Hello
계속하려면 아무 키나 누르십시오 . . .

04 다음의 프로그램에서 생성되는 출력 결과는 무엇인가?

(a)

```
int i = 0;
while( i < 10 )
{
    printf("%d\n", i);
    i += 3;
}
```

0
3
6
9

(b)

```
int i = 0;
do
{
    printf("%d\n", i);
    i += 3;
} while( i < 10 );
```

0
3
6
9

(c)

```
int i;
for( i = 0; i < 10; i += 2 )
{
    printf("%d\n", i);
}
```

0
2
4
6
8

(d)

```
int i;
for( i = 10; i >= 0; i-- )
{
    printf("%d\n", i);
}
```

10
9
8
7
6
5
4
3
2
1
0

(e)

```
int x, y;
for( x = 0; x < 3; x++ )
    for( y = 2; y >= 0; y-- )
        printf("*");
```

(f)

```
int i = 5;
for( ; i; i--)
    printf("%d", i);
```

54321

05 동일한 결과를 생성하도록 for 루프는 while 루프로, while 루프는 for 루프로 변환하라.

(a)

```
int i;
for( i = 10; i >= 0; i-- )
{
    printf("%d\n", i);
}
```

(b)

```
int i = 10;
while( i >= 0 )
{
    printf("%d\n", i);
    i -= 3;
}
```

```
int i=10;
while(i>=0){
    printf("%d\n", i);
    i--;
}
```

```
int i;
for(i=10;i>=0;i-=3)
{
    printf("%d\n",i);
}
```

06 다음의 코드가 실행을 완료하였을 변수 i의 최종값은 얼마인가?

(a) `for(i = 2; i < 10; i += 3)`

`;`

NULL 문장이다. 처리할
내용이 없을 때 적어준다.

(b) `int i = 0;
do
{
i++;
} while(i < 10);`

6. (a) 11 (b) 10

07 다음의 코드에서 잘못된 점이 있으면 지적하고 올바르게 수정하라. 논리적인 오류도 포함된다.

(a) `int i = 0;
while(i < 10)
{
printf("i의 값\n", i);
}`

(b) `int i = 0;
while(i++ < 10) ;
{
printf("i의 값\n", i);
}`

(a) i의 값이 변함이 없으므로 무한루프를 수행하게 된다.

```
int i = 0;  
while( i < 10 )  
{  
    printf("i의 값\n", i);  
    i++;  
}
```

(b) while이 있는 줄의 끝에 있는 세미콜론(;)을 제거하여야 한다.

```
int i = 0;  
while( i++ < 10 )  
{  
    printf("i의 값\n", i);  
}
```

(c) `int i;
for(i = 0; i < 10; i--)
printf("i = %d\n", i);`

(d) `float x;
for(x = 0.1; x != 1.0; x += 0.1)
printf("%f\n", i);`

(c) 증감의 방향을 바꾸어야 한다.

```
int i;  
for( i = 0; i < 10; i++ )  
    printf("i = %d\n", i);
```

(d) 부동 소수점 수는 오차를 가지고 있으므로 정확히 1.0이 되지 않아서 무한반복이 된다. 만약 1.0까지의 값을 보려 했던 소스라면 `x!=1.0`을 `x<1.0`으로 수정하는 편이 좋다.

```
float x;  
for(x = 0.1; x < 1.0; x += 0.1)  
    printf("%f\n", i);
```

(e) `int i;
for(; i < 10; i++)
printf("i = %d\n", i);`

(e) 변수 i가 초기화되지 않았다.
`int i;
for(i=0 ; i < 10; i++)
printf("i = %d\n", i);`

08 다음과 같은 코드에서 break 문이나 continue 문을 사용하지 않고 동일한 결과를 내도록 수정하여 보라.

(a) `int i;
for(i = 0; i < 10; i++)
{
if(i == 5)
break;
printf("%d\n", i);
}`

(b) `int i;
for(i = 0; i < 10; i++)
{
if(i == 5)
continue;
printf("%d\n", i);
}`

(a) `int i;
for(i = 0; i < 5; i++)
{
printf("%d\n", i);
}`

(b) `int i;
for(i = 0; i < 10 ; i++)
{
if(i != 5)
printf("%d\n", i);
}`

09 다음의 수학식을 계산하는 코드를 작성하라.

(a) $\sum_{i=1}^{30} (i^2 + 1)$

(b) $\sum_{i=10}^{30} \sum_{j=0}^5 (i+j)$

(a) $\sum_{i=1}^{30} (i^2 + 1)$

```
int sum=0, i;  
for(i = 1; i <= 30; i++)  
    sum += i*i+1;
```

(b) $\sum_{i=10}^{30} \sum_{j=0}^5 (i+j)$

```
int sum=0, i, j;  
for(i = 10; i <= 30; i++)  
    for(j = 0; j <= 5; j++)  
        sum += (i + j);
```

- 01 가끔은 일정한 시간 동안 아무 일도 하지 않으면서 시간을 지연시킬 목적으로 반복문을 사용하기도 한다. 사용자에게서 하나의 수를 입력받아서 변수에 저장한다. 반복문을 사용하여 이 변수의 값을 1씩 감소시키면서 이 변수의 값이 0이 될 때까지 반복한다. 반복이 끝나면 벨소리를 낸다.

실행결과

```
C:\WINDOWS\system32\cmd.exe
카운터의 초기값을 입력하시오: 10
10 9 8 7 6 5 4 3 2 1
```

HINT for 루프를 사용한다. 벨소리는 \a를 콘솔로 출력하면 된다.

```
int counter, i;
```

```
printf("카운터의 초기값을 입력하시오: ");
```

```
scanf("%d", &counter);
```

```
for (i = counter; i >= 1; i--) {
```

```
    printf("%d ", i);
```

```
}
```

```
printf("\n\a");
```

- 02 1부터 100사이의 모든 3의 배수의 합을 계산하여 출력하는 프로그램을 반복 구조를 사용하여 작성하라.

실행결과

```
C:\WINDOWS\system32\cmd.exe
1부터 100 사이의 모든 3의 배수의 합은 1683입니다.
```

HINT 3의 배수의 합은 $i \% 3 == 0$ 의 조건으로 검사할 수 있다.

```
int i, sum;
```

```
i = 1;
```

```
sum = 0;
```

```
while (i <= 100)
```

```
{
```

```
    if ((i % 3) == 0)
```

```
        sum += i;
```

```
    i++;
```

```
}
```

```
printf("1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
```

```
// for 사용
```

```
sum = 0;
```

```
for (i = 1; i <= 100; i++) {
```

```
{
```

```
    if ((i % 3) == 0)
```

```
        sum += i;
```

```
}
```

```
printf("(for 사용 1) 1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
```

```
sum = 0;
```

```
for (i = 3; i <= 100; i += 3) {
```

```
{
```

```
    sum += i;
```

```
}
```

```
printf("(for 사용 2) 1부터 100 사이의 모든 3의 배수의 합은 %d입니다. \n", sum);
```

03 사용자가 입력한 정수의 모든 약수를 화면에 출력하는 프로그램을 작성하라.

실행결과



```
C:\WINDOWS\system32\cmd.exe
정수를 입력하시오: 60
약수: 1 2 3 4 5 6 10 12 15 20 30 60
```

HINT 약수는 % 연산자로 알 수 있다.

```
int x;

printf("정수를 입력하시오: ");
scanf("%d", &x);

printf("약수: ");
for (int i = 1; i <= x; i++) {
    if (x % i == 0)
        printf("%d ", i);
}
printf("\n");
```

04 반복 루프를 사용하여 다음과 같은 패턴을 출력하는 프로그램을 작성하라.

실행결과



```
C:\WINDOWS\system32\cmd.exe
*
**
***
****
*****
*****
*****
*****
```

HINT 중첩 반복 구조를 사용한다.

```
int y, x;

for (y = 0; y < 7; y++) {
    for (x = 0; x < (6 - y); x++)
        printf(" ");
    for (x = 0; x <= y; x++)
        printf("*");
    printf("\n");
}
```

```
for (y = 0; y < 7; y++)
{
    for (x = 0; x < 7; x++)
    {
        if (x < (6-y))
        {
            printf(" ");
        }
        else
        {
            printf("*");
        }
    }
    printf("\n");
}
```


05 중첩 반복문을 사용하여 다음과 같이 출력하는 프로그램을 작성하여 보자.

실행결과

```
C:\WINDOWS\system32\cmd.exe
정수를 입력하시오: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
int x, y, number;
```

```
printf("정수를 입력하시오: ");
scanf("%d", &number);
for (y = 1; y <= number; y++) {
    for (x = 1; x <= y; x++)
        printf("%d ", x);
    printf("\n");
}
```

06 앞에서 간단한 정수 계산기를 만들어본 적이 있다. 이 계산기 프로그램에 메뉴를 추가하도록 한다. 다음과 같은 메뉴를 화면에 출력하고 사용자가 메뉴 중에서 하나를 선택할 때까지 반복을 계속한다. do...while 반복문을 사용하여 사용자가 적절한 선택을 했는지를 검사하도록 하라. 만약 사용자가 A, S, M, D, Q가 아닌 다른 문자를 입력하면 "연산을 선택하시오:" 메시지를 계속해서 출력한다. 하나의 메뉴가 선택되면 해당되는 연산을 실행하고 다시 메뉴를 선택할 수 있도록 하라. 반복을 종료하는 메뉴인 Q는 break 문을 이용하여 구현하도록 하라.

실행결과

```
C:\WINDOWS\system32\cmd.exe
*****
A---- Add
S---- Subtract
M---- Multiply
D---- Divide
Q---- Quit
*****
연산을 선택하시오:A
두수를 공백으로 분리하여 입력하시오: 10 20
30
```

HINT 연산을 나타내는 문자는 scanf("%c", &op)를 이용하여 입력받도록 하라. 무한 루프를 사용하고 'Q'가 입력되면 break; 문을 실행하여서 반복 루프를 빠져나간다.

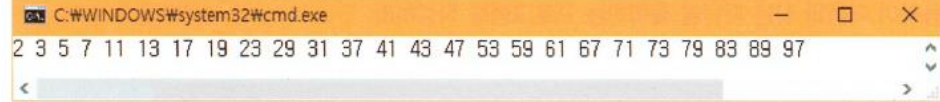
```
char op;
int x, y;
```

```
while (1) {
    printf("*****\n");
    printf("A---- Add \n");
    printf("S---- Subtract \n");
    printf("M---- Multiply \n");
    printf("D---- Divide \n");
    printf("Q---- Quit \n");
    printf("*****\n");
    do {
        printf("연산을 선택하시오:");
        scanf(" %c", &op);
    } while (op != 'A' && op != 'S' && op != 'M' && op != 'D' && op != 'Q');
    if (op == 'Q')
        break;
    printf("두수를 공백으로 분리하여 입력하시오: ");
    scanf("%d %d", &x, &y);

    if (op == 'A')
        printf("%d \n", x + y);
    else if (op == 'S')
        printf("%d \n", x - y);
    else if (op == 'M')
        printf("%d \n", x * y);
    else if (op == 'D')
        printf("%d \n", x / y);
    else
        printf("지원되지 않는 연산자입니다. \n");
}
```

07 2와 100 사이에 있는 모든 소수(prime number)를 찾는 프로그램을 작성하라. 정수가 소수가 되려면 1과 자기 자신만을 약수로 가져야 한다.

실행결과



HINT 소수의 약수는 1과 자기자신뿐이다. 따라서 1부터 자기 자신 사이에 약수가 하나라도 있다면 소수가 아니다.

```
int i, j;
int isprime; // 소수인지 판단하는 변수

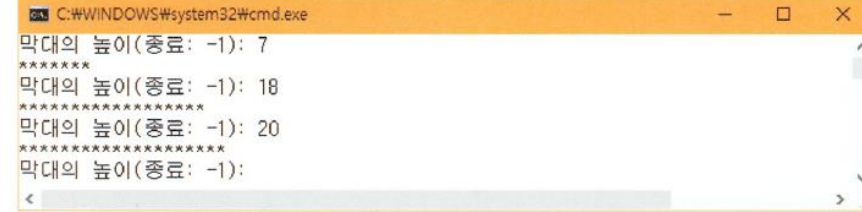
for (i = 2; i <= 100; i++) {
    isprime = 1; // 소수일 것이라고 가정하자
    for (j = 2; j < i; j++) {
        if (i % j == 0) { // 약수가 존재한다면 소수가 아니다.
            isprime = 0; // 소수가 아니다
            break;
        }
    }
    // 소수의 경우 isprime == 1 이다.
    if (isprime == 1) {
        printf("%d ", i);
    }
}
```

```
int i, j;

for (i = 2; i <= 100; i++) {
    for (j = 2; j <= i/2; j++) { // 가장 큰 약수는 i/2
        if (i % j == 0) // 약수가 존재한다면 소수가 아니다.
            break;
    }
    // 소수의 경우 j > i/2 로 나오고
    // 소수가 아닌 경우 j <= i/2 로 나온다.
    if (j > i/2) // i/2 까지 약수가 없다면, 소수다
        printf("%d ", i);
}
printf("\n");
```

08 컴퓨터는 막대 그래프를 그리는 데도 사용된다. 사용자로부터 1부터 50사이의 숫자를 입력받아서 숫자만큼의 별표를 출력하는 프로그램을 작성하라. 막대는 가로로 그려지게 된다.

실행결과



HINT 입력받은 막대의 높이만큼 반복하면서 "*"을 출력하면 된다. 중첩 반복 구조를 사용한다.

```
int i, j, height;
while (1)
{
    printf("막대의 높이(종료: -1): ");
    scanf("%d", &height);
    if (height == -1)
        break;
    for (j = 0; j < height; j++)
        printf("*");
    printf("\n");
}
```

09 (1+2+3+...+n)가 10000을 넘지 않으면서 가장 큰 값과 그 때의 n의 값을 구하라.

실행결과



HINT 무한 루프를 실행하면서 sum에 i의 값을 더한다. 반복할 때마다 i는 1씩 증가된다. sum이 10000을 넘으면 sum에서 i를 빼고 i를 감소시킨 후에 break를 실행한다.

```
int i, sum;
i = 0;
sum = 0;
while (1)
{
    i++;
    sum += i;
    if (sum >= 10000)
        break;
}
printf("1부터 %d까지의 합이 %d입니다.\n", (i - 1), sum - i);
```

10 실수의 거듭 제곱값을 계산하는 프로그램을 작성하여 보자. 사용자로부터 하나의 실수 r과 거듭 제곱 횟수를 나타내는 정수 n을 입력받아서 r^n 을 구하여 화면에 출력한다.

실행결과



HINT i를 1부터 n까지 증가시키면서 result에 r을 곱해준다. result의 초기값은 1.0이어야 한다.

```
int n, i;
double r, result;

printf("실수의 값을 입력하시오: ");
scanf("%lf", &r);
printf("거듭제곱횟수를 입력하시오: ");
scanf("%d", &n);

result = 1.0;
for (i = 0; i < n; i++)
    result *= r;
printf("결과값은 %f\n", result);
```


11 $1^2 + 2^2 + 3^2 + \dots n^2$ 의 값을 계산하여 출력하여 보자.

실행결과



```
C:\WINDOWS\system32\cmd.exe
n의 값을 입력하시오: 10
계산값은 385입니다.
```

HINT i를 1부터 n까지 증가시키면서 result에 i^2 를 더한다. result의 초기값은 0이어야 한다.

```
int n, i;
int result;
```

```
printf("n의 값을 입력하시오: ");
scanf("%d", &n);
```

```
result = 0;
for (i = 1; i <= n; i++)
    result += i * i;
printf("계산값은 %d입니다.\n", result);
```

12 피보나치 수열은 다음과 같이 정의되는 수열이다.

$$f_0 = 0$$

$$f_1 = 1$$

$$f_{i+1} = f_i + f_{i-1} \quad \text{for } i = 1, 2, \dots$$

피보나치 수열에서는 앞의 2개의 원소를 합하여 뒤의 원소를 만든다. 피보나치 수열은 컴퓨터에서도 탐색 문제 등에 사용되기도 한다. 피보나치 수열을 생성하여 출력하는 프로그램을 작성하여 보자.

실행결과



```
C:\WINDOWS\system32\cmd.exe
몇번째 항까지 구할까요? 10
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,
```

HINT 3개의 변수 a=0, b=1, c를 사용하여 c=a+b; a=b; b=c;을 반복한다.

```
int i, f0 = 0, f1 = 1, f2, n;
```

```
printf("몇번째 항까지 구할까요? ");
scanf("%d", &n);
for (i = 0; i <= n; i++) {
    printf("%d, ", f0);
    f2 = f0 + f1;
    f0 = f1;
    f1 = f2;
}
printf("\n");
```

- 13 서로 다른 n 개에서 r 개를 택하여 일렬로 나열하는 방법의 수를 순열(permutation)이라 하고, ${}_nP_r$ 로 표시한다. 순열은 다음과 같은 식을 이용하여 구할 수 있다. 순열을 구하는 프로그램을 작성하라. n 과 r 은 사용자가 입력할 수 있도록 하라.

$${}_nP_r = n(n-1)(n-2) \dots (n-r+1)$$

실행결과

```
C:\WINDOWS\system32\cmd.exe
n의 값: 10
r의 값: 3
순열의 값은 720입니다.
```

HINT for(i=n; i>=(n-r+1); i--) 스타일의 반복문을 사용하는 것이 편리하다. 부등호의 방향에 주의하라.

```
int n, r, result, i;

printf("n의 값: ");
scanf("%d", &n);
printf("r의 값: ");
scanf("%d", &r);
result = 1;
for (i = n; i >= (n - r + 1); i--)
    result = result * i;
printf("순열의 값은 %d입니다.\n", result);
return 0;
```

- 14 사용자가 입력한 특정한 정수의 자리수를 반대로 출력하는 프로그램을 작성하라. 예를 들어서 사용자가 정수 1206을 입력하였다면 6021이 출력되어야 한다. do...while 문을 사용하여 보라.

```
C:\WINDOWS\system32\cmd.exe
정수를 입력하시오: 123456
654321
```

HINT 1의 자리수는 $n \% 10$ 으로 구할 수 있다. 10의 자리수는 먼저 n 을 10으로 나눈 후에 $n \% 10$ 하면 된다. 100의 자리수는 n 을 100으로 나눈 후에 $n \% 10$ 하면 된다. 한번 반복할 때마다 하나의 자리수가 구해지도록 하라.

```
int x, digit;
```

```
printf("정수를 입력하시오: ");
scanf("%d", &x);
```

```
do {
    digit = x % 10;
    x /= 10;
    printf("%d", digit);
} while (x != 0);
```