

Chapter 04 반복문

01 반복문과 배열

02 while 반복문

03 do while 반복문

04 for 반복문

05 역 for 반복문

06 foreach 반복문

07 중첩 반복문

08 break 키워드

09 continue 키워드

10 함께하는 응용예제

요약

연습문제

Section 01 반복문과 배열(1)

- 복사해서 붙여 넣기를 사용한 반복

코드 4-1

복사해서 붙여넣기를 사용한 반복

/4장/LoopSummary

```
01 static void Main(string[] args)
02 {
03     Console.WriteLine("출력");
04     Console.WriteLine("출력");
05     Console.WriteLine("출력");
06     Console.WriteLine("출력");
07     Console.WriteLine("출력");
08 }
```

- 반복문 사용 예

코드 4-2

반복문을 사용한 반복

/4장/LoopSummary

```
01 static void Main(string[] args)
02 {
03     for (int i = 0; i < 1000; i++)
04     {
05         Console.WriteLine("출력");
06     }
07 }
```

Section 01 반복문과 배열(2)

■ 기본적인 배열 생성 방법

자료형[] 이름 = {자료, 자료}

그림 4-1 배열 선언 형식

배열[인덱스]



그림 4-2 배열의 요소

■ 예

코드 4-3

기본적인 배열 생성 방법

/4장/Arrays

```
01 static void Main(string[] args)
02 {
03     int[] intArray = { 52, 273, 32, 65, 103 };
04     long[] longArray = { 52, 273, 32, 65, 103 };
05     float[] floatArray = { 1.0F, 2.0F, 3.0F, 4.0F, 5.0F };
06     double[] doubleArray = { 1.0, 2.0, 3.0, 4.0, 5.0 };
07     char[] charArray = { '가', '나', '다', '라' };
08     string[] stringArray = { "윤인성", "연하진", "윤아린" };
09 }
```

Section 01 반복문과 배열(3)

- 기본예제 4-1 배열생성하고 요소에 접근하기(교재 159p)

/4장/ArrayBasic

코드 4-4

배열 생성하고 요소에 접근

```
01 static void Main(string[] args)
02 {
03     // 배열을 생성합니다.
04     int[] intArray = { 52, 273, 32, 65, 103 };
05
06     // 배열의 요소를 변경합니다.
07     intArray[0] = 0;
08
09     // 요소를 출력합니다.
10     Console.WriteLine(intArray[0]);
11     Console.WriteLine(intArray[1]);
12     Console.WriteLine(intArray[2]);
13     Console.WriteLine(intArray[3]);
14 }
```

실행 결과

```
0
273
32
65
```

NOTE(1)

■ Length 속성

- 배열의 요소 개수 확인

코드 4-5 배열의 Length 속성

/4장/Arrays

```
static void Main(string[] args)
{
    // 배열을 생성합니다.
    int[] intArray = { 52, 273, 32, 65, 103 };

    // 배열의 길이를 출력합니다.
    Console.WriteLine(intArray.Length);
}
```

■ IndexOutOfRangeException

코드 4-6 배열의 범위를 벗어나는 인덱스에 접근

/4장/Arrays

```
static void Main(string[] args)
{
    // 배열을 생성합니다.
    int[] intArray = { 52, 273, 32, 65, 103 };

    // 요소의 길이를 출력합니다.
    Console.WriteLine(intArray[5]);
}
```

실행 결과

처리되지 않은 예외: System.IndexOutOfRangeException: 인덱스가 배열 범위를 벗어났습니다.
--생략--

Section 01 반복문과 배열(4)

■ 원하는 크기의 배열 생성 방법

코드 4-7

원하는 크기의 배열 생성 방법

/4장/Arrays

```
01 int[] array = new int[100];
```

■ 기본예제 4-2 원하는 크기의 배열 생성(교재 161p)

/4장/VariousArray

코드 4-8

특정한 크기의 배열 생성

```
01 static void Main(string[] args)
02 {
03     // 배열을 생성합니다.
04     int[] intArray = new int[100];
05
06     // 요소의 길이를 출력합니다.
07     Console.WriteLine(intArray[0]);
08     Console.WriteLine(intArray[99]);
09 }
```

일반적인 숫자 자료형 : 0
문자열 자료형 : 빈 문자열로 초기화된다.

실행 결과

```
0
0
```

Section 02 while 반복문(1)

- if 조건문과 형식 비슷하나, 불 표현식이 참인 동안 중괄호 안의 문장 계속 실행
- 조건이 변하지 않으면 무한히 반복, 반드시 조건을 거짓으로 만드는 문장 포함 되어야 함

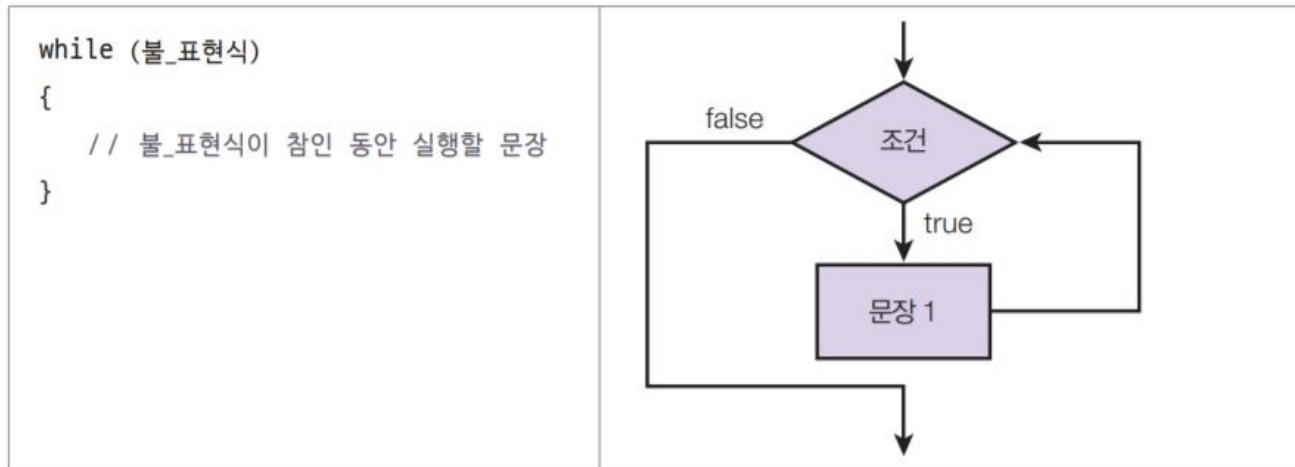


그림 4-3 while 반복문 형식과 순서도

- 예

코드 4-9

무한 반복

/4장/Loops

```
01 static void Main(string[] args)
02 {
03     while (true)
04     {
05         Console.WriteLine("무한 반복");
06     }
07 }
```


Section 02 while 반복문(2)

- 4장/ WhileBasic (교재 163p)

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    int i = 0;
    int[] intArray = { 52, 273, 32, 65, 103 };

    // 반복을 수행합니다.
    while (i < intArray.Length)
    {
        // 출력합니다.
        Console.WriteLine(i + "번째 출력:" + intArray[i]);

        // 탈출을 위해 변수를 더합니다.
        i++;
    }
}
```

Section 03 do while 반복문(1)

- while 반복문과 형태 비슷하나, 조건 비교 부분이 마지막에 위치

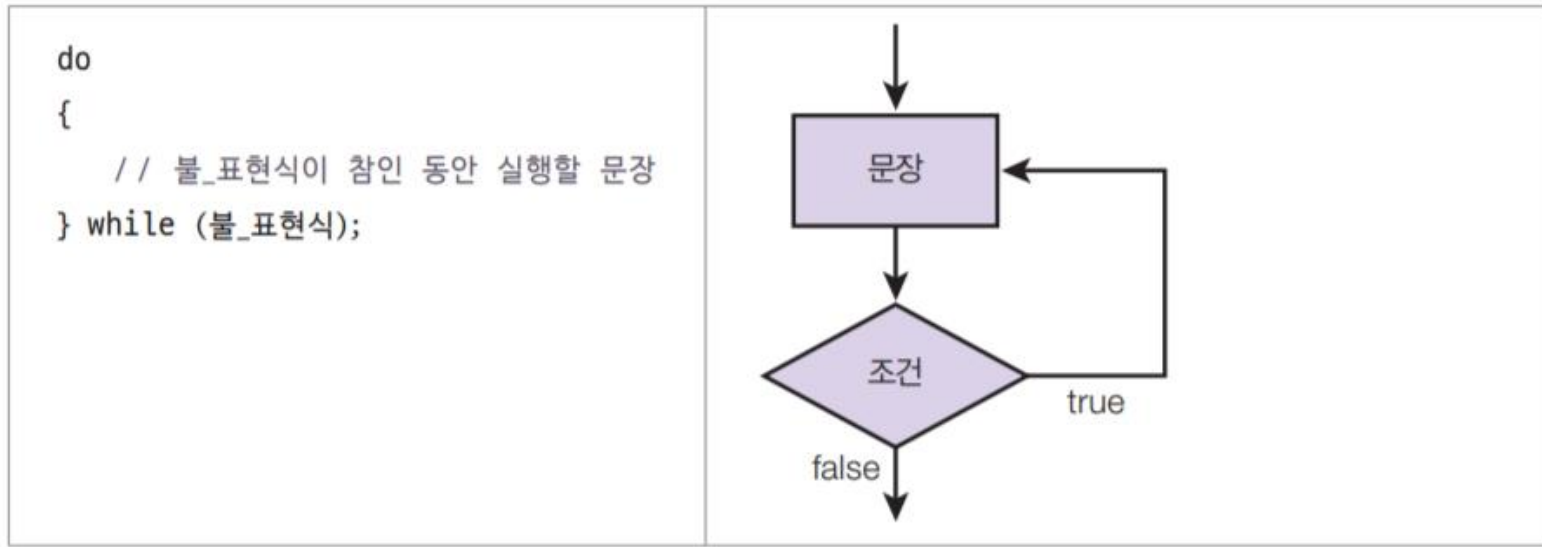


그림 4-4 do while 반복문 형식과 순서도

Section 03 do while 반복문(2)

- 기본예제 4-4 do while 반복문 활용하기(교재 165p)

/4장/ DoWhileBasic

코드 4-11 do while 반복문 활용

```
01 static void Main(string[] args)
02 {
03     string input;
04     do
05     {
06         Console.Write("입력(exit을 입력하면 종료): ");
07         input = Console.ReadLine();
08     } while (input != "exit");
09 }
```

실행 결과

```
입력(exit을 입력하면 종료): ㅇㅂㅇ
입력(exit을 입력하면 종료): ㅇㅅㅇ
입력(exit을 입력하면 종료): exit
```

Section 04 for 반복문(1)

- 원하는 횟수만큼 반복
- 맨 먼저 초기식 실행 후 조건식 확인
- for 반복문의 각 단계
 - 초기식 비교
 - 조건식 비교(조건 거짓이면 반복문 종료)
 - 문장을 실행한다.
 - 종결식을 실행한다.
 - 2단계로 이동한다.

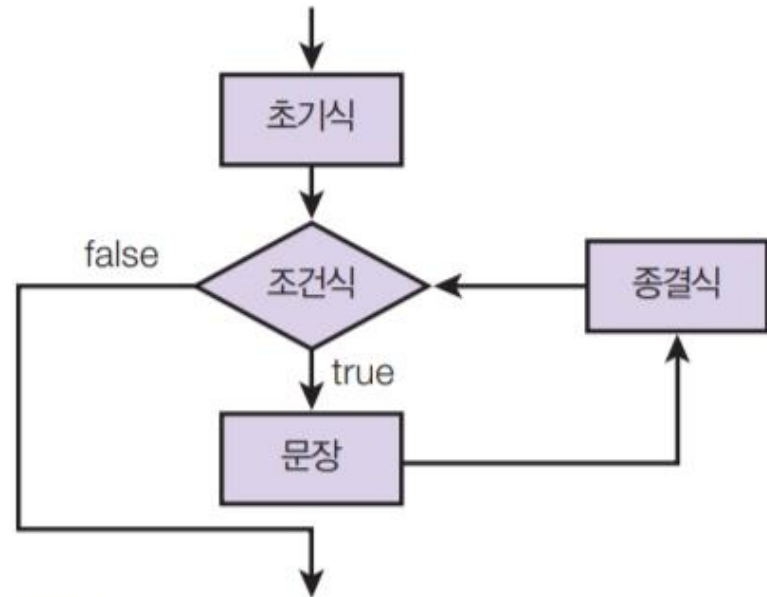


그림 4-5 for 반복문 순서도

```
for (int i = 0; i < 반복_횟수; i++)  
{  
  
}
```

Section 04 for 반복문(2)

- 기본예제 4-5 for 반복문으로 덧셈하기(교재 167p)

/4장/SumWithFor

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    int output = 0;

    // 반복을 수행합니다.
    for (int i = 0; i <= 100; i++)
        output += i;

    // 출력합니다.
    Console.WriteLine(output);
}
```

실행 결과

5050

Section 04 for 반복문(2)

- 기본예제 4-5 for 반복문으로 곱셈하기(교재 168p)

/4장/MultiplyWithFor

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    int output = 1;

    // 반복을 수행합니다.
    for (int i = 1; i <= 20; i++)
        output *= i;

    // 출력합니다.
    Console.WriteLine(output);
}
```

Section 04 for 반복문(2)

- 기본예제 4-7 한글 전부 출력하기(교재 168p)

/4장/KoreanCharacters

```
static void Main(string[] args)
{
    for (int i = '가'; i <= '힣'; i++)
        Console.Write((char)i);
}
```

- '가'부터 '힣'까지 11,172자 출력!
- 한글은 컴퓨터 내부적으로 글자를 표현하는 유니코드 방식에서
- 한자 다음으로 많은 글자수를 차지하는 언어
- 한자는 한국, 중국, 일본, 베트남에서 공용으로 사용

Section 05 역 for 반복문

- 배열 뒤에서부터 실행 하는 경우 사용
- 기본예제 4-8** 역 for 반복문(교재 170p)

/4장/ForReverse

```
static void Main(string[] args)
{
    // 배열을 생성합니다.
    int[] intArray = { 1, 2, 3, 4, 5, 6 };

    // 요소의 길이를 출력합니다.
    for (int i = intArray.Length - 1; i >= 0; i--)
        Console.WriteLine(intArray[i]);
}
```

실행 결과

6
5
4
3
2
1

Section 06 foreach 반복문(1)

- 반복문을 컬렉션에 쉽게 적용할 때 사용
 - 컬렉션 : 여러 개체가 모여서 집합을 이룬 것
 - 컬렉션 예

```
foreach (자료형 변수 in 컬렉션)
{
    ...
}
```

- foreach 반복문 예

```
for (int i = 0; i < 컬렉션.길이; i++)
{
    자료형 변수 = 컬렉션[i];
    ...
}
```

Section 06 foreach 반복문(2)

- **기본예제 4-9** foreach 반복문과 배열(교재 171p)

/4장/ForeachBasic

```
static void Main(string[] args)
{
    string[] array = { "사과", "배", "포도", "딸기", "바나나" };

    // 반복을 수행합니다.
    foreach (string item in array)
        Console.WriteLine(item);
}
```

- **기본예제 4-10** foreach 반복문과 var 키워드(교재 172p)

/4장/ForeachWithVar

```
static void Main(string[] args)
{
    string[] array = { "사과", "배", "포도", "딸기", "바나나" };

    // 반복을 수행합니다.
    foreach (var item in array)
        Console.WriteLine(item);
}
```

■ 코드 조각

- 자주 사용되는 코드 정의해서 사용하기 쉽게 만든 것
- for를 입력 후, tab 키 두 번 누름

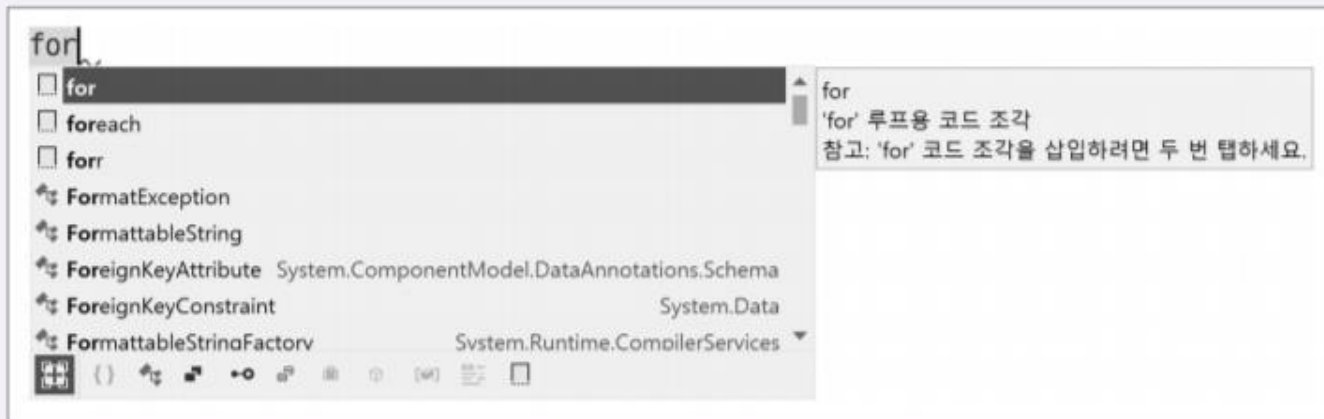


그림 4-6 코드 조각

코드 4-19 for 코드 조각

```
for (int i = 0; i < length; i++)  
{  
  
}
```

Section 07 중첩 반복문

- 기본예제 4-11 별 피라미드(1)(교재 174p)

/4장/PyramidA

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < i + 1; j++)
            Console.Write('*');
        Console.WriteLine();
    }
}
```

실행 결과

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
```

Section 07 중첩 반복문

- 기본예제 4-12 별 피라미드(2)(교재 175p)

/4장/PyramidB

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10 - i; j++)
            Console.Write(' ');
        for (int j = 0; j < i + 1; j++)
            Console.Write('*');
        Console.WriteLine();
    }
}
```



Section 08 break 키워드

- switch조건문 또는 반복문을 벗어날 때에 사용하는 키워드
- 기본예제 4-13 break 키워드(교재 177p) [/4장/BreakBasic](#)

```
static void Main(string[] args)
{
    while (true)
    {
        Console.Write("숫자를 입력해주세요(짝수를 입력하면 종료): ");
        int input = int.Parse(Console.ReadLine());
        if (input % 2 == 0)
            break;
    }
}
```

실행 결과

```
숫자를 입력해주세요(짝수를 입력하면 종료): 31
숫자를 입력해주세요(짝수를 입력하면 종료): 51
숫자를 입력해주세요(짝수를 입력하면 종료): 61
숫자를 입력해주세요(짝수를 입력하면 종료): 52
```

Section 09 continue 키워드

- 반복문 내부에서 현재 반복을 멈추고 다음 반복을 진행하게 하는 키워드

- 기본예제 4-14 continue 키워드(교재 179p)

/4장/ContinueBasic

```
static void Main(string[] args)
{
    for (int i = 1; i < 10; i++)
    {
        if (i % 2 == 0)
            continue;
        Console.WriteLine(i);
    }
}
```

➔ "홀수를 출력하라" 와 같음

실행 결과

1
3
5
7
9

Section 10 함께 하는 응용 예제(1)

■ 응용예제 4-1 문자열 처리(교재 181p)

/4장/StringProcess

① 대문자화와 소문자화

표 4-1 대문자화와 소문자화 메서드

메서드	설명
ToUpper()	문자열 내부의 문자를 모두 대문자로 변경합니다.
ToLower()	문자열 내부의 문자를 모두 소문자로 변경합니다.

```
// 대문자화와 소문자화
string input = "Potato Tomato";
Console.WriteLine(input.ToUpper());
Console.WriteLine(input.ToLower());
```

실행 결과

```
POTATO TOMATO
potato tomato
```

- C#의 모든 문자열 처리 메서드는 자신을 변경하지 않고 반환
- 비파괴적 메서드 : 자신을 변경하지 않고 반환하는 메서드

Section 10 함께 하는 응용 예제(2)

② 문자열 자르기

표 4-2 문자열 자르기 메서드

메서드	설명
Split()	문자열을 특정한 문자 또는 문자열로 자릅니다.

```
input = "감자 고구마 토마토";  
string[] inputs = input.Split(new char[] { ' ' });
```

```
foreach (var item in inputs)  
{  
    Console.WriteLine(item);  
}
```

실행 결과

```
감자  
고구마  
토마토
```

Section 10 함께 하는 응용 예제(2)

③ 문자열 양 옆의 공백 제거

표 4-3 공백 제거 메서드

메서드	설명
Trim()	문자열 양 옆의 공백을 제거합니다.
TrimStart()	문자열 앞의 공백을 제거합니다.
TrimEnd()	문자열 뒤의 공백을 제거합니다.

```
input = " test      \n";  
Console.WriteLine("::" + input.Trim() + "::");  
Console.Read();
```

실행 결과

```
::test::
```

Section 10 함께 하는 응용 예제(3)

- ④ 배열을 문자열로 변환 : 지정한 문자로 배열의 요소가 연결되어 출력됨

표 4-4 배열의 요소 연결 메서드

메서드	설명
<code>string.Join()</code>	배열의 요소를 뭉쳐 문자열로 변경합니다.

```
// 배열을 문자열로  
string[] array = { "감자", "고구마", "토마토", "가지" };  
Console.WriteLine(string.Join(",", array));
```

실행 결과

감자,고구마,토마토,가지

Section 10 함께 하는 응용 예제(4)

- 응용예제 4-2 이동하는 달팽이(교재 185p)

/4장/MovingAt

표 4-5 추가적인 메서드

메서드	설명
<code>Console.Clear()</code>	콘솔 화면을 지웁니다.
<code>Console.SetCursorPosition()</code>	콘솔 화면의 특정한 위치로 커서를 옮깁니다.
<code>Thread.Sleep()</code>	특정한 시간만큼 스레드를 정지합니다.

현재는 그냥 프로그램을 정지한다고
기억해도 됩니다. 스레드와 관련된
내용은 이후에 살펴보겠습니다.

`Thread.Sleep(1000);` ← 1초 동안 정지

Section 10 함께 하는 응용 예제(4)

- 응용예제 4-2 이동하는 달팽이(교재 185p)

/4장/MovingAt

```
static void Main(string[] args)
{
    int x = 1;
    while (x < 50)
    {
        // 화면을 지우고 커서를 이동합니다.
        Console.Clear();
        Console.SetCursorPosition(x, 5);

        // 출력합니다.
        if (x % 3 == 0)
            Console.WriteLine(" __@");
        else if (x % 3 == 1)
            Console.WriteLine("_ ^@");
        else
            Console.WriteLine("^ _@");

        Thread.Sleep(100);
        x++;
    }
}
```

Section 10 함께 하는 응용 예제(5)

■ 응용예제 4-3 switch 조건문과 무한 반복문(교재 188p)

/4장/SwitchWithWhile

```
static void Main(string[] args)
{
    bool state = true;
    while (state)
    {
        ConsoleKeyInfo info = Console.ReadKey();
        switch (info.Key)
        {
            case ConsoleKey.UpArrow:
                Console.WriteLine("위로 이동");
                break;
            case ConsoleKey.RightArrow:
                Console.WriteLine("오른쪽으로 이동");
                break;
            case ConsoleKey.DownArrow:
                Console.WriteLine("아래로 이동");
                break;
            case ConsoleKey.LeftArrow:
                Console.WriteLine("왼쪽으로 이동");
                break;
            case ConsoleKey.X:
                state = false;
                break;
        }
    }
}
```

Random 클래스

- Random 클래스의 객체를 생성하여 6개의 랜덤 숫자 출력하기
 - Next()
 - Next(100) // 0~99까지 랜덤 값
 - NextDouble() : 0~1 사이의 랜덤 실수값

```
Random r = new Random();

int[] intArray = new int[6];

for (int i = 0; i < 6; i++)
    intArray[i] = r.Next(1, 46);    // 1 ~ 45까지 숫자

Console.WriteLine("생성된 6개의 랜덤한 숫자 출력: ");

foreach (int value in intArray)
    Console.Write("{0, 5}", value);
```

■ 문제 1) 놀이 공원 입장

- 9세 이상이면서, 키가 130cm 이상인 어린이는 "고속 롤러코스트 입장 가능", 9세 이상이면서 키가 130cm 미만인 어린이는 "저속 롤러코스트 입장 가능", 나이가 9세 미만인 어린이는 "입장 불가!"를 출력하는 프로그램이 **무한 반복**되도록 한다.

단, 나이 입력에서 exit 입력할 경우 프로그램 종료! (대문자, 소문자, 대소문자 섞여서 입력해도 모두 종료될 수 있도록 한다.

- 실행 결과)

나이 입력(종료 : exit) : 9
키 입력 : 132
고속 롤러코스트 입장 가능

나이 입력 (종료 : exit) : 9
키 입력 : 120
저속 롤러코스트 입장 가능

나이 입력 (종료 : exit) : 7
입장 불가!

■ 문제 2)

- 1부터 입력 받은 수 중에서 2의 배수이면서 3의 배수가 아닌 수를 모두 출력하는 프로그램을 작성하시오. (단, 조건이 무한 반복이 아닌 while문 이용)

- 실행 결과)

정수 하나 입력(종료 : 0) : 20

1 ~ 20까지 2의 배수이면서 3의 배수가 아닌 수 : 2 4 8 10 14 16 20

정수 하나 입력(종료 : 0) : 10

1 ~ 20까지 2의 배수이면서 3의 배수가 아닌 수 : 2 4 8 10

정수 하나 입력(종료 : 0) : 0

감사합니다.

■ 문제 3)

- 입력 받은 라인 수에서 ★의 개수가 1개씩 점차 줄어들게 출력되도록 프로그램을 작성하시오.

- 실행 결과)

라인 수 입력 : 6

6: ★★★★★★

5: ★★★★★

4: ★★★★★

3: ★★★

2: ★★

1: ★

■ 문제 4)

- 구구단을 출력하는 프로그램을 아래와 같이 무한 반복되도록 구현하시오.

- 실행 결과)

프로그램 종료: (q or Q)

출력할 구구단(2~9) : -3

> 유효하지 않은 데이터 값!

> 다시 입력하세요.

프로그램 종료: (q or Q)

출력할 구구단(2~9) : 5

----- 5단 -----

$5 * 1 = 5$

$5 * 2 = 10$

.

.

■ 문제 5)

- 입력한 숫자의 합계와 평균을 출력하는 프로그램을 작성하시오.
 - 0~10 사이의 수만 입력 가능
 - 평균은 소수점이하 2자리까지 출력

- 실행 결과)

입력할 숫자의 개수 : 4

수 4개 입력

1 번째 : 1

2 번째 : 100

유효한 수가 아님! 다시 입력

2 번째 : 2

3 번째 : 3

4 번째 : 4

합계 : 10, 평균 : 2.50

■ 문제 6)

- 랜덤하게 생성할 수의 개수를 입력 받아 0~99 사이의 랜덤 값을 저장하고, 최대값과 최소값을 출력하는 프로그램을 작성하시오.

- 실행 결과)

랜덤하게 생성할 수의 개수 : 7

랜덤값 출력 : 18 8 79 5 94 51 25

최대값 : 94

최소값 : 5