

# 프로그래밍 언어

과정명	XR기술을 활용한 게임 개발자 과정 - 1기		
이름	김의성	평가일자	2024-07-19

- 해당 테스트는 구현능력보다 프로그래밍에 대한 내용들에 대한 기초지식을 요구합니다.
- 모르는 문제가 있다면 반드시 다시 학습하는 것을 권장합니다.
- 답안에 공백이나 줄바꿈의 여부로 오답을 처리하지 않고 출력결과만으로 채점합니다.
- 답안은 문제지의 가장 마지막 표에 입력합니다.

- 다음 중 C#의 자료형에 대한 내용 중 틀린 답을 고르시오.
  - int는 정수형 자료형이며 4byte의 크기를 가진다.
  - float는 실수형 자료형이며 4byte의 크기를 가진다.
  - double은 실수형 자료형이며 8byte의 크기를 가진다.
  - char는 문자형 자료형이며 2byte의 크기를 가진다.
  - string은 문자열 자료형이며 최대 50byte의 크기를 가진다.
- 다음 중 C#의 변수에 대한 설명으로 틀린 답을 고르시오.
  - 변수를 선언하기 위해 자료형과 변수명이 필요하다.
  - 변수에 처음으로 데이터를 저장하는 것을 초기화라 하며, 초기화 전까지는 값을 불러올 수 없다.
  - 변수명은 변수들을 구분하기 위한 이름을 선정하는 것으로 의미가 같은 경우 중복해서 사용한다.
  - 변수에 데이터를 저장하기 위해서는 변수를 =(대입연산자) 좌측에 배치한다.
  - 변수의 데이터를 불러오기 위해서는 데이터가 필요한 곳에 변수명을 배치한다.
- 다음 중 C#에서 허용하지 않은 문법을 고르시오.
  - int intValue = 10;
  - char charValue = 'a';
  - float floatValue = 20.3;
  - double doubleValue = 10.4;
  - bool boolValue = true;
- 다음 중 C#의 형변환으로 적합하지 않은 것을 고르시오.
  - int intValue = (int)1.4;
  - float floatValue = (float)"12.3";
  - char charValue = (char)79;
  - int intValue = (int)'a';
  - double doubleValue = 3.14f;

5. 다음 코드의 출력을 적으시오.

```
byte value = 255;
value++;
Console.Write(value);
```

6. 다음 코드의 오류가 없을 경우 빈칸에 알맞은 키워드를 적으시오.

```
int value = 빈칸 1.5;
```

7. 다음 코드와 출력이 틀린 것을 고르시오.

```
int intValue = 23;

Console.WriteLine(intValue + 10); // (a) 33
Console.WriteLine(intValue - 10); // (b) 13
Console.WriteLine(intValue * 10); // (c) 230
Console.WriteLine(intValue / 10); // (d) 2.3
Console.WriteLine(intValue % 10); // (e) 3
```

8. 다음 코드와 출력이 틀린 것을 고르시오.

```
int value = -5;

Console.WriteLine(+value); // (a) -5
Console.WriteLine(-value); // (b) 5
Console.WriteLine(++value); // (c) -4
Console.WriteLine(value++); // (d) -3
Console.WriteLine(value+3); // (e) 0
```

9. 다음 비교연산자의 설명으로 옳지 않은 것을 고르시오.

- a. 비교연산자는 true 또는 false 결과를 갖는다.
- b. 비교연산자 = 은 좌측값과 우측값이 같을 경우 결과가 true이다.
- c. 비교연산자 != 은 좌측값과 우측값이 다를 경우 결과가 true이다.
- d. 비교연산자 > 은 좌측값이 우측값보다 클 경우 결과가 true이다.
- e. 비교연산자 < 은 좌측값이 우측값보다 작을 경우 결과가 true이다.

10. 다음 코드의 출력을 적으시오.

```
Console.WriteLine( true && true);  
Console.WriteLine( true && false);  
Console.WriteLine(false && true);  
Console.WriteLine(false && false);
```

11. 다음 코드의 출력을 적으시오.

```
Console.WriteLine( true || true);  
Console.WriteLine( true || false);  
Console.WriteLine(false || true);  
Console.WriteLine(false || false);
```

12. 다음 코드의 출력을 적으시오.

```
Console.WriteLine( !true);  
Console.WriteLine(!false);
```

13. 다음 코드의 출력을 적으시오.

```
Console.WriteLine(true && false || true);
```

14. 다음 코드의 출력을 적으시오.

```
bool value = false && (true || false) && true;  
Console.WriteLine(value);
```

15. 다음 코드의 출력을 적으시오.

```
int value = 20;
value += 10;
Console.WriteLine(value);
```

16. 다음 코드의 출력을 적으시오.

```
int value = 30;
value -= 10;
value /= 2;
value *= 3;
Console.WriteLine(value);
```

17. 다음 코드의 출력을 적으시오.

```
int value = 2 + 2 * 2;
Console.WriteLine(value);
```

18. 다음 코드의 출력을 적으시오.

```
int value = (2 + 3) * 4 - 10 / (3 - 2);
Console.WriteLine(value);
```

19. 다음 코드의 출력을 적으시오.

```
bool value = true;
if (value)
{
    Console.WriteLine("감자");
}
else
{
    Console.WriteLine("고구마");
}
```

20. 다음 코드의 출력을 적으시오.

```
int score = 89;
if (score <= 80)
{
    Console.WriteLine("브론즈 입니다.");
}
else if (score <= 90)
{
    Console.WriteLine("실버 입니다.");
}
else if (score <= 100)
{
    Console.WriteLine("골드 입니다.");
}
```

21. 다음 코드의 출력을 적으시오.

```
int score = 89;
if (score <= 80)
{
    Console.WriteLine("브론즈 입니다.");
}
if (score <= 90)
{
    Console.WriteLine("실버 입니다.");
}
if (score <= 100)
{
    Console.WriteLine("골드 입니다.");
}
```

22. 다음 코드의 출력을 적으시오.

```
int score = 89;
if (score <= 100)
{
    Console.WriteLine("골드 입니다.");
}
else if (score <= 90)
{
    Console.WriteLine("실버 입니다.");
}
else if (score <= 80)
{
    Console.WriteLine("브론즈 입니다.");
}
```



23. 다음 코드는 C# 에서 허용하는 문법이 아니다. 이를 수정하여 조건문을 구성하시오.

```
int value = 70;
if (50 <= value <= 100)
{
    Console.WriteLine("값은 50 이상 100 이하입니다.");
}
```

24. 다음 코드의 빈칸으로 알맞은 내용을 적으시오.

```
char key = 's';
switch (key)
{
    빈칸 'w':
    빈칸 'W':
        Console.WriteLine("위쪽으로 이동합니다.");
        break;
    빈칸 'a':
    빈칸 'A':
        Console.WriteLine("왼쪽으로 이동합니다.");
        break;
    빈칸 's':
    빈칸 'S':
        Console.WriteLine("아래쪽으로 이동합니다.");
        break;
    빈칸 'd':
    빈칸 'D':
        Console.WriteLine("오른쪽으로 이동합니다.");
        break;
}
```

25. 다음 코드의 빈칸에 알맞는 키워드를 작성하시오.

```
switch (input)
{
    case 1:
        Console.WriteLine("값은 1");
        break;
    case 2:
        Console.WriteLine("값은 2");
        break;
    빈칸
        Console.WriteLine("그 외의 값");
        break;
}
```

26. 다음 코드의 출력을 적으시오.

```
int value = 1;
for (int i = 0; i < 5; i++)
{
    value++;
}

Console.WriteLine(value);
```

27. 다음 코드의 출력을 적으시오.

```
int value = 0;
for (int i = 12; i < 34; i++)
{
    value++;
}

Console.WriteLine(value);
```

28. 다음 코드의 출력을 적으시오.

```
int value = 0;
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 4; j++)
    {
        value++;
    }
}

Console.WriteLine(value);
```

29. 다음 코드의 출력을 적으시오.

```
for (int i = 128; i >= 20; i /= 2)
{
    Console.Write(i);
}
```

30. 다음 코드의 출력을 적으시오.

```
int value = 0;
do
{
    value++;
} while (value < 0);

Console.WriteLine(value);
```

31. 다음 코드의 출력을 적으시오.

```
int value = 0;
while (value < 20)
{
    value++;
    if (value > 10)
        break;
}

Console.WriteLine(value);
```



32. 다음 코드의 출력을 적으시오.

```
int value = 0;
for (int i = 0; i < 10; i++)
{
    if (i % 2 == 0)
        continue;
    value++;
}

Console.WriteLine(value);
```

33. 다음 코드의 출력을 적으시오.

```
참조 1개
static int Add(int left, int right)
{
    return left + right;
}

참조 0개
static void Main(string[] args)
{
    Console.WriteLine(Add(10, 20));
}
```

34. 다음 코드의 출력을 적으시오.

```
참조 3개
static int Add(int left, int right)
{
    return left + right;
}

참조 0개
static void Main(string[] args)
{
    Console.WriteLine(
        Add(Add(20, 30), Add(5, 7)));
}
```

35. 아래의 코드를 보고 빈칸을 적으시오.

```
참조 1개
static void Swap(int left, int right)
{
    int temp = left;
    left = right;
    right = temp;
}

참조 0개
static void Main(string[] args)
{
    int left = 10;
    int right = 20;
    Swap(left, right);
    Console.WriteLine($"{left}{right}");
}
```

36. 다음 코드의 출력을 적으시오.

```
참조 1개
static void Func2()
{
    Console.Write("2전");
    //
    Console.Write("2후");
}

참조 1개
static void Func1()
{
    Console.Write("1전");
    Func2();
    Console.Write("1후");
}

참조 0개
static void Main()
{
    Console.Write("Main전");
    Func1();
    Console.Write("Main후");
}
```

37. 아래의 코드를 보고 출력을 적으시오.

```
참조 1개
static void Func2(int value)
{
    Console.Write(value++);
}

참조 1개
static void Func1(int value)
{
    Console.Write(value++);
    Func2(value);
    Console.Write(value++);
}

참조 0개
static void Main(string[] args)
{
    Func1(5);
}
```

38. 아래의 코드 출력결과가 주석과 같을 때 빈칸에 알맞은 키워드를 적으시오. (복수 정답 인정)

```
참조 1개
static void Function(빈칸 int value)
{
    value = 10;
}

참조 0개
static void Main(string[] args)
{
    int value = 0;
    Function(빈칸 value);
    Console.WriteLine(value); // 출력 : 10
}
```

39. 다음 중 구조체에 대한 설명으로 옳지 않은 것을 고르시오.

- a. 구조체는 여러 데이터를 하나로 묶어 관리하는 자료형을 정의하는 방법이다.
- b. 구조체 변수를 선언하는 경우 구조체 내용에 포함된 모든 변수가 같이 포함된다.
- c. 구조체의 변수에 접근하고 싶을 때 . 을 통해 접근할 수 있다.
- d. 구조체 변수에 접근제한자를 public으로 설정할 경우 외부에서 접근이 차단된다.
- e. 구조체 변수에 접근제한자를 설정하지 않는 경우 기본적으로 private으로 설정된다.

40. 다음 코드의 출력을 적으시오.

```
참조 4개
struct Score
{
    public int math;
    public int english;
}

참조 0개
static void Main()
{
    Score score1 = new Score() { math = 10, english = 20 };
    Score score2 = new Score() { math = 20, english = 30 };

    Console.WriteLine(score1.math + score2.english);
}
```

41. 다음 코드의 출력을 적으시오.

```
참조 3개
struct MyStruct
{
    public int value1;
    public int value2;
}

참조 0개
static void Main(string[] args)
{
    MyStruct s = new MyStruct() { value1 = 1, value2 = 2 };
    MyStruct t = s;

    s.value1 = 3;
    t.value2 = 4;

    Console.WriteLine(s.value1 + s.value2 + t.value1 + t.value2);
}
```

42. 다음 중 열거형에 대한 설명을 옳지 않은 것을 고르시오.

- a. 열거형은 이름으로 관리될 수 있어 코드의 가독성적인 측면에 도움이 된다.
- b. 열거형은 기본 정수 숫자의 명명된 상수 집합에 의해 정의되는 값 형식이다.
- c. 열거형의 멤버는 정수로 표현 가능하며 서로 형변환이 가능하다.
- d. 열거형의 멤버는 자동으로 지정되며 변경할 수 없다.
- e. 열거형의 멤버는 자동으로 0부터 시작하고 순서에 따라 1씩 증가한다.

43. 다음 코드의 출력을 적으시오.

```
참조 2개
enum Direction { Left, Right, Up, Down }

참조 0개
static void Main()
{
    Console.WriteLine((int)(Direction.Left));
    Console.WriteLine((Direction)2);
}
```

44. 다음 코드의 출력을 적으시오.

```
참조 2개
enum Season { Spring, Summer, Autumn = 20, Winter }

참조 0개
static void Main()
{
    Console.WriteLine((Season)21);
    Console.WriteLine((Season)50);
}
```

45. 다음 중 배열에 대한 설명으로 옳지 않은 것을 고르시오.

- a. 배열은 동일한 자료형의 요소들로 구성된 데이터 집합이다.
- b. 배열은 인덱스를 통하여 배열요소에 접근할 수 있다.
- c. 배열의 처음 요소의 인덱스는 1부터 시작한다.
- d. 크기가 n인 배열의 마지막 요소의 인덱스는 n-1이다.
- e. 배열이 갯수 뿐만 아니라 차원수를 늘려 관리가 필요한 경우 다차원 배열을 사용할 수 있다.

46. 아래의 코드의 출력을 적으시오.

```
int[] array = { 1, 2, 3, 4, 5 };  
Console.WriteLine(array[2]);
```

47. 아래의 코드의 출력을 적으시오.

```
int[] array = { 1, 2, 3, 4, 5 };  
Console.WriteLine(array.Length);
```

48. 아래의 코드의 출력을 적으시오.

```
int[] array = { 1, 3, 5, 7, 9 };  
int sum = 0;  
for (int i = 0; i < array.Length; i++)  
{  
    sum += array[i];  
}  
Console.WriteLine(sum);
```

49. 아래의 코드의 출력을 적으시오.

```
int[,] matrix =  
{  
    { 1, 2, 3, 4, 5 },  
    { 6, 7, 8, 9, 10 },  
    { 11, 12, 13, 14, 15 },  
    { 16, 17, 18, 19, 20 },  
};  
Console.WriteLine(matrix[3, 1]);
```



50. 다음 코드의 출력을 적으시오.

```
int[,] matrix =  
{  
    { 1, 2, 3, 4, 5 },  
    { 6, 7, 8, 9, 10 },  
    { 11, 12, 13, 14, 15 },  
    { 16, 17, 18, 19, 20 },  
};  
  
int sum = 0;  
for (int y = 0; y < 2; y++)  
{  
    for (int x = 0; x < 4; x++)  
    {  
        sum += matrix[y, x];  
    }  
}  
  
Console.WriteLine(sum);
```

## 답안 제출지

1	e	11	true,true,true, false	21	실버 입니 다. , 골드 입 니다.	31	11	41	10
2	c	12	false,tue	22	실버 입니다.	32	5	42	d
3	c	13	true	23	50<=value && 100>=value	33	30	43	0 Up
4	e	14	false	24	case	34	62	44	Winter 50
5	0	15	30	25	default	35	10,20	45	c
6	float	16	30	26	6	36	Main전1전2 전2후1후 Main후	46	3
7	e	17	6	27	21	37	5 6 6	47	5
8	d	18	10	28	12	38	ref	48	25
9	d	19	감자	29	128,64,32	39	d	49	17
10	true,false,fals e,false	20	실버 입니다.	30	1	40	40	50	40