

[문제 1]

30

[해설]

```

③ def cal(a):
④     ary = list(range(a))
⑤     h = 0
⑥     for i in ary:
⑦         if (i % 2 == 0):
⑧             h += i
⑨     return h
①  a = 12
②⑩ result = cal(a)
⑪  print(result)
    
```

- ① a에 12를 저장한다.
- ② a의 값 12를 인수로 cal 함수를 호출한 다음 돌려받은 값을 result에 저장한다.
- ③ cal 함수의 시작점이다. ②번에서 전달받은 12를 a가 받는다.
- ④ 0에서 시작하여 a-1까지 1씩 증가하는 숫자를 저장한 리스트 ary를 생성한다.

	ary[0]	ary[1]	ary[2]	ary[3]	ary[4]	ary[5]	ary[6]	ary[7]	ary[8]	ary[9]	ary[10]	ary[11]
리스트 ary	0	1	2	3	4	5	6	7	8	9	10	11

- ⑤ h에 0을 저장한다.
- ⑥ 리스트 ary는 12개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑦, ⑧번을 12회 수행한다.
- ⑦ i를 2로 나눈 나머지가 0이면 ⑧번을 수행한다.
- ⑧ 'h = h + i'와 동일하다. h에 i의 값을 누적한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	h
0	0
1	0
2	2
3	6
4	12
5	20
6	30
7	
8	
9	
10	
11	

- ⑨ h의 값 30을 호출한 곳으로 반환한다.
- ⑩ cal 함수로부터 반환받은 값 30을 result에 저장한다.
- ⑪ result의 값 30을 출력한다.

[문제 2]

17

[해설]

```
❶ a = list(range(1,12,3))
❷ r = 0
❸ for i in a[2:]:
❹     r += i
❺ print(r)
```

❶ 1에서 시작하여 11(12-1)까지 3씩 증가하는 숫자를 저장한 리스트 a를 생성한다.

	a[0]	a[1]	a[2]	a[3]
리스트 a	1	4	7	10

❷ r에 0을 저장한다.

❸ 리스트 a의 a[2] 위치에서 마지막 위치까지의 요소를 i에 할당하면서 ❹번을 반복 수행한다.

❹ r에 i의 값을 누적한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	r
7	0
10	7
	17

❺ r의 값 17을 출력한다.

[문제 3]

5

[해설]

```
❷ def func(a):
❸     b = a * -1
❹     lst = list(range(a, b, -2))
❺     r = 0
❻     for i in lst:
❼         r += i
❽     return r
❶❹ result = func(5)
❷❺ print(result)
```

❶ 5를 인수로 func 함수를 호출한 다음 돌려받은 값을 result에 저장한다.

❷ func 함수의 시작점이다. ❶번에서 전달받은 5를 a가 받는다.

❸ a에 -1을 곱한 값을 b에 저장한다. b는 -5이다.

❹ a에서 시작하여 b+1(-4)까지 -2씩 감소하는 숫자를 저장한 리스트 lst를 생성한다.

	lst[0]	lst[1]	lst[2]	lst[3]	lst[4]
리스트 lst	5	3	1	-1	-3

❺ r에 0을 저장한다.

❻ 리스트 lst는 5개의 요소를 가지므로 각 요소를 i에 할당하면서 ❼번을 5회 수행한다.

⑦ r에 i의 값을 누적한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

func 함수			
a	b	i	r
5	-5	5	0
		3	5
		1	8
		-1	9
		-3	8
			5

⑧ r의 값 5를 호출한 곳으로 반환한다.

⑨ func 함수로부터 반환받은 값 5를 result에 저장한다.

⑩ result의 값 5를 출력한다.

#### [문제 4]

##### REFERENCE

##### ※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 R, E, F, ..., E로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

##### [해설]

```

③ def add(a):
④     c = list(a)
⑤     for i in range(1, len(c) * 2, 2):
⑥         c.insert(i, 'E')
⑦     return c
⑩ def prnt(st):
⑪     for i in st:
⑫         print(i, end = '')
① a = 'rfr'.upper()
②③ st = add(a)
⑨ prnt(st)
⑬ print('NCE')

```

① 문자열 "rfr"을 대문자로 변경하여 a에 저장한다.

② a의 값 "RFR"을 인수로 add 함수를 호출한 다음 돌려받은 값을 st에 저장한다.

③ add 함수의 시작점이다. ②번에서 전달받은 "RFR"을 a가 받는다.

④ 문자열 변수 a를 리스트로 변환하여 c에 저장한다.

	c[0]	c[1]	c[2]
리스트 c	R	F	R

⑤ 반복 변수 i는 1에서 시작하여 2씩 증가하면서 리스트 c의 길이에 2를 곱한 값인 6보다 작은 동안 ⑥번을 반복 수행한다.

- len(리스트) : 리스트의 요소 수를 구한다. len(c)는 3이다.

⑥ 리스트 c의 i 위치에 "E"를 삽입한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	리스트 c					
1	c[0]	c[1]	c[2]	c[3]		
	R	E	F	R		
3	c[0]	c[1]	c[2]	c[3]	c[4]	
	R	E	F	E	R	
5	c[0]	c[1]	c[2]	c[3]	c[4]	c[5]
	R	E	F	E	R	E

⑦ 리스트 c를 호출한 곳으로 반환한다.

⑧ add 함수로부터 반환받은 리스트를 st에 저장한다.

	st[0]	st[1]	st[2]	st[3]	st[4]	st[5]
리스트 st	R	E	F	E	R	E

⑨ 리스트 st를 인수로 prnt 함수를 호출한다.

⑩ prnt 함수의 시작점이다. ⑨번에서 전달받은 리스트를 st가 받는다.

⑪ 리스트 st는 6개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑫번을 6회 수행한다.

⑫ i의 값을 출력한다. 종료 문자가 “”이므로 다음 출력은 공백이나 줄 넘김 없이 그대로 이어서 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	출력
R	R
E	RE
F	REF
E	REFE
R	REFER
E	REFERE

⑬ 앞의 출력에 이어서 “NCE”를 출력한다.

출력 **REFERENCE**

## [문제 5]

0101

### ※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 콤마를 넣어 0, 1, 0, 1로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

### [해설]

문제의 코드는 숫자를 입력받아 2진수로 변환한 후 비트 연산 NOT(1의 보수)을 적용하여 출력하는 프로그램이다. 10진수를 2진수로 변환하려면 10진수를 2로 나누어 나머지를 구한 후 저장하고, 다시 몫을 2로 나누어 나머지를 구해 저장하는 과정을 반복한다. 비트 연산 NOT은 이렇게 구해진 2진수의 각 자리를 1에서 빼는 과정을 통해 구해진다.

```

③ def bin(a):
④     b = list()
⑤     while(a != 0):
⑥         b.append(a % 2)
⑦         a = int(a / 2)
⑧     return b
⑪ def cpl(r):
⑫     e = len(r) - 1
⑬     for i in range(e, -1, -1):
⑭         print(1 - r[i], end = "")
① a = int(input())
②⑨ r = bin(a)
⑩ cpl(r)

```

① 입력받은 값을 정수로 변환하여 a에 저장한다. 문제에서 10을 입력한다고 하였으므로 a는 10이다.

② a의 값 10을 인수로 bin 함수를 호출한 다음 돌려받은 값을 r에 저장한다.

③ bin 함수의 시작점이다. ②번에서 전달받은 10을 a가 받는다.

④ b를 리스트로 선언한다.

⑤ a가 0이 아닌 동안 ⑥, ⑦번 문장을 반복하여 수행한다.

⑥ a를 2로 나눈 나머지를 리스트 b의 마지막에 추가한다.

⑦ a를 2로 나누어 정수로 변환한 값을 a에 저장한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

a	리스트 b
10	<div style="display: flex; justify-content: space-around;"> <span>b[0]</span> </div> <div style="border: 1px solid black; width: 40px; margin: 0 auto; text-align: center;">0</div>
5	<div style="display: flex; justify-content: space-around;"> <span>b[0]</span> <span>b[1]</span> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">0</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">1</div> </div>
2	<div style="display: flex; justify-content: space-around;"> <span>b[0]</span> <span>b[1]</span> <span>b[2]</span> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">0</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">1</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">0</div> </div>
1	<div style="display: flex; justify-content: space-around;"> <span>b[0]</span> <span>b[1]</span> <span>b[2]</span> <span>b[3]</span> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">0</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">1</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">0</div> <div style="border: 1px solid black; width: 40px; height: 20px; text-align: center;">1</div> </div>
0	

⑧ 리스트 b를 호출한 곳으로 반환한다.

⑨ bin 함수로부터 반환받은 리스트를 r에 저장한다.

리스트 r

r[0]	r[1]	r[2]	r[3]
0	1	0	1

⑩ 리스트 r을 인수로 cpl 함수를 호출한다.

⑪ cpl 함수의 시작점이다. ⑩번에서 전달받은 리스트를 r이 받는다.

⑫ 리스트 r의 길이 4에서 1을 뺀 값 3을 e에 저장한다.

- len(리스트) : 리스트의 요소 수를 구한다. len(r)은 4이다.

⑬ 반복 변수 i는 e(3)에서 시작하여 -1씩 감소하면서 0까지 순서대로 저장하며 ⑭번을 반복 수행한다.

⑭ 1에서 r[i]를 뺀 값을 출력한다. 종료 문자가 “”이므로 다음 출력은 공백이나 줄 넘김 없이 그대로 이어서 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

	r[0]	r[1]	r[2]	r[3]
리스트 r	0	1	0	1

e	i	r[i]	1-r[i]	출력
3	3	1	0	0
	2	0	1	01
	1	1	0	010
	0	0	1	0101

#### [문제 6]

dif = t

#### [해설]

문제의 코드는 숫자를 입력받아 리스트에 저장한 후, 저장된 숫자 중 33에 가장 가까운 수를 구하는 프로그램이다. 33에 가장 가깝다는 의미는 어떤 수와 33과의 차이가 가장 작다는 말과 같다. 33과 가장 가까운 수를 구하려면 33과 다른 수들의 차를 계산한 후 차이를 비교하여 차이가 가장 작은 수를 찾으면 된다. 주의할 점은 차이를 계산할 때 음수가 나오면 안 되므로 33과 어떤 수의 대·소를 비교한 후 큰 수에서 작은 수를 뺀다.

```

❶ a = list()
❷ for i in range(10):
❸     a.append(int(input()))
❹ dif = 99
❺ for i in range(10):
❻     if (a[i] > 33):
❼         t = a[i] - 33
❽         else:
❾             t = 33 - a[i]
❿     if (t <= dif):
⓫         dif = t
⓬         r = f"dif is {dif}, value is {a[i]}"
⓭ print(r)

```

- ❶ a를 리스트로 선언한다.
- ❷ 반복 변수 i는 0에서 9까지 순서대로 저장하며 ❸번을 반복 수행한다.
- ❸ 입력받은 값을 정수로 변환하여 리스트 a의 마지막에 추가한다.
- ❹ dif에 99를 저장한다.
- ❺ 반복 변수 i는 0에서 9까지 순서대로 저장하며 ❻~⓫번을 반복 수행한다.
- ❻ a[i]가 33보다 크면 ❼번을 수행하고, 아니면 ❽번을 수행한다.
- ❼ a[i]에서 33을 뺀 값을 t에 저장한다.
- ❽ 33에서 a[i]를 뺀 값을 t에 저장한다.
- ❾ t가 dif보다 작거나 같으면 ⓫, ⓬번을 수행한다.
- ⓫ dif에 t의 값을 저장한다.
- ⓬ “dif is ” + dif의 값 + “, value is ” + a[i]의 값의 형태를 가진 문자열을 r에 저장한다.
- ⓭ r을 출력한다.

※ 리스트 a에 80, 92, 30, 42, 38, 11, 19, 23, 36, 9가 입력되었다고 가정한 반복문 실행에 따른 변수들의 값

의 변화는 다음과 같다.

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
리스트 a	80	92	30	42	38	11	19	23	36	9

dif	i	a[i]	t	r	출력
99	0	80	47	dif is 47, value is 80	dif is 3, value is 36
47	1	92	59	dif is 3, value is 30	
3	2	30	3	dif is 3, value is 36	
3	3	42	9		
	4	38	5		
	5	11	22		
	6	19	14		
	7	23	10		
	8	36	3		
	9	9	24		

#### [문제 7]

10 6 3 1 0

#### ※ 답안 작성 시 주의 사항

프로그램의 실행 결과는 부분 점수가 없으므로 정확하게 작성해야 합니다. 예를 들어 출력값 사이에 코마를 넣어 10, 6, 3, 1, 0으로 썼을 경우 부분 점수 없이 완전히 틀린 것으로 간주됩니다.

#### [해설]

문제의 코드는 리스트에 저장된 숫자를 삽입 정렬 알고리즘을 이용하여 내림차순으로 정렬하는 프로그램이다. 삽입 정렬은 두 번째 자료부터 시작하여 그 앞(왼쪽)의 자료들과 비교하여 삽입할 위치를 지정한 후 자료를 뒤로 옮기고 지정한 자리에 자료를 삽입하여 정렬하는 알고리즘이다. 즉, 두 번째 자료는 첫 번째 자료, 세 번째 자료는 두 번째와 첫 번째 자료, 네 번째 자료는 세 번째, 두 번째, 첫 번째 자료와 비교한 후 자료가 삽입될 위치를 찾는다. 자료가 삽입될 위치를 찾았다면 그 위치에 자료를 삽입하기 위해 자료를 한 칸씩 뒤로 이동시킨다.

```

② def init():
③     lst = [ 0 ]
④     for i in range(1, 5):
⑤         lst.append(lst[i - 1] + i)
⑥     return lst
⑨ def sort(a):
⑩     for i in range(1, 5):
⑪         j = i - 1
⑫         k = a[i]
⑬         while (j >= 0 and a[j] < k):
⑭             a[j + 1] = a[j]
⑮             j -= 1
⑯         a[j + 1] = k
⑰ a = init()
⑧ sort(a)
⑰ for i in a:
⑱     print(i, end = ' ')

```

① 인수 없이 init 함수를 호출한 다음 돌려받은 값을 a에 저장한다.

- ② init 함수의 시작점이다.
- ③ 하나의 요소를 갖는 리스트 lst를 생성한다.
- ④ 반복 변수 i는 1에서 4까지 순서대로 저장하며 ⑤번을 반복 수행한다.
- ⑤  $lst[i - 1]$ 에 i를 더한 값을 리스트 lst의 마지막에 추가한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	리스트 lst
	<div>lst[0]</div> <div>0</div>
1	<div>lst[0] lst[1]</div> <div>0 1</div>
2	<div>lst[0] lst[1] lst[2]</div> <div>0 1 3</div>
3	<div>lst[0] lst[1] lst[2] lst[3]</div> <div>0 1 3 6</div>
4	<div>lst[0] lst[1] lst[2] lst[3] lst[4]</div> <div>0 1 3 6 10</div>

- ⑥ 리스트 lst를 호출한 곳으로 반환한다.
- ⑦ init 함수로부터 반환받은 리스트를 a에 저장한다.
- ⑧ 리스트 a를 인수로 sort 함수를 호출한다. 리스트를 인수로 하는 경우 리스트가 저장된 메모리 주소가 전달되므로 sort 함수에서 변경된 리스트의 각 요소의 값이 리스트 a에도 적용된다.
- ⑨ sort 함수의 시작점이다. ⑧번에서 전달받은 리스트를 a가 받는다.
- ⑩ 반복 변수 i는 1에서 4까지 순서대로 저장하며 ⑪~⑯번을 반복 수행한다.
- ⑪ j에 i에서 1을 뺀 값을 저장한다.
- ⑫ k에 a[i]의 값을 저장한다.
- ⑬ j가 0보다 크거나 같고, a[j]가 k보다 작으면 ⑭, ⑮번을 반복 수행한다.
- ⑭  $a[j + 1]$ 에 a[j]의 값을 저장한다.
- ⑮  $j = j - 1$ 과 동일하다. j에서 1을 뺀 값을 j에 저장한다.
- ⑯  $a[j + 1]$ 에 k의 값을 저장한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

i	j	k	리스트 a
1	0 -1	1	<div>a[0] a[1] a[2] a[3] a[4]</div> <div>0 1 3 6 10</div> <div>1 0</div>
2	1 0 -1	3	<div>a[0] a[1] a[2] a[3] a[4]</div> <div>1 0 3 6 10</div> <div>3 1 0</div>
3	2 1 0 -1	6	<div>a[0] a[1] a[2] a[3] a[4]</div> <div>3 1 0 6 10</div> <div>6 3 1 0</div>



4	3	10	a[0]	a[1]	a[2]	a[3]	a[4]
	2		6	3	1	0	10
	1		10	6	3	1	0
	0						
	-1						

⑰ 리스트 a는 5개의 요소를 가지므로 각 요소를 i에 할당하면서 ⑱번을 5회 수행한다.

⑱ i의 값을 출력한다. 종료 문자가 “ ”이므로 다음 출력은 공백 한 칸을 띄운 후 그대로 이어서 출력한다.

※ 반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

	a[0]	a[1]	a[2]	a[3]	a[4]
리스트 a	10	6	3	1	0

i	출력
10	10
6	10 6
3	10 6 3
1	10 6 3 1
0	10 6 3 1 0