

제17회 서울특별시 정보올림피아드 경시 부문 (예선)

고등학생부 문제 (40문항)

수험번호() ()고등학교 제()학년 성명 ()

※ 다음 문제(1번~40번)에 가장 알맞은 답을 하나만 골라 답안지의 해당번호에 기표하시오.

1. 다음은 무엇에 관한 설명인가?

사용자를 기준으로 다운로드(Download)와 업로드(Upload)의 데이터 전송속도가 다르다는 것을 의미하는 것으로 한 개의 전화선에서 낮은 주파수는 전화를, 높은 주파수는 데이터 통신에 이용하는 방법으로 데이터 통신과 일반전화를 동시에 사용할 수 있다.

① ATM ISDN ADSL B-ISDN VAN

2. 다음은 전자 상거래에 관련된 설명이다. 옳지 않은 것은?

- ① 전자문서는 컴퓨터와 같은 장치에 의하여 전자적인 형태로 작성, 송·수신 또는 저장된 정보를 말한다.
- ② 전자서명은 전자문서를 작성한 사람의 신원과 전자문서의 변경 여부를 확인할 수 있도록 대칭 암호화방식을 이용하여 전자 서명 생성키로 생성한 정보로서 당해 전자문서에 고유한 것을 말한다.
- ③ 전자서명키는 전자 서명 생성키와 이에 합치하는 전자서명 검증키를 말한다.
- ④ 인증은 전자 서명 검증키가 자연인 또는 법인이 소유하는 전자 서명 생성키에 합치한다는 사실을 확인·증명하는 행위를 말한다.
- ⑤ 개인정보는 생존하는 개인에 관한 정보로서 당해 정보에 포함되어 있는 성명·주민등록번호 등의 사항에 의하여 당해 개인을 식별할 수 있는 정보를 말한다.

3. 다음 데이터 베이스에 관련된 약어 설명 중 옳지 않은 것은?

- ① DIL : Data Implementation Language의 약어로써 데이터 베이스의 구조를 구현하는 언어이다.
- ② DML : Data Manipulation Language의 약어로써 데이터 베이스를 조작하는 언어이다.
- ③ DD : Data Dictionary의 약어로써 데이터 베이스의 구성요소들을 설명하는 사전이다.
- ④ SQL : Structured Query Language의 약어로써 데이터 베이스의 내용을 구조적으로 질의하는 언어이다.
- ⑤ DBA : Database Administrator의 약어로써 데이터 베이스 관리자를 의미한다.

4. 2진수, 8진수, 10진수, 16진수의 수가 보기와 같을 때 큰 주부터 나열한 것은?

<보기>

㉠ (187)₁₀ ㉡ (272)₈ ㉢ (BC)₁₆ ㉣ (10111001)₂

- ① ㉠-㉡-㉢-㉣ ㉡-㉢-㉠-㉣ ㉢-㉠-㉣-㉡
- ② ㉡-㉠-㉢-㉣ ㉡-㉢-㉠-㉣

5. 다음은 IP 주소에 대한 설명이다. 옳지 않은 것은?

- ① 인터넷은 TCP/IP를 기반으로 하는 IP주소 체계를 따른다.
- ② IP 주소는 32비트 체계로써 4 옥텟(octet)으로 표현한다.
- ③ IP 주소는 소속된 네트워크에 할당된 부분과 컴퓨터에 할당된 부분으로 구분된다.
- ④ IP주소는 네트워크의 크기에 따라 A, B, C 3개의 클래스(class)로 나뉜다.
- ⑤ 이론적으로 클래스 C의 등록가능 호스트 수는 255개이다.

6. 다음은 방화벽(Firewall)에 대한 설명이다. 옳지 않은 것은?

- ① 외부와 내부 네트워크 사이의 유일한 통로에 설치한다.
- ② 외부 사용자를 차단하기 위한 것이다.
- ③ 불법적인 접근을 거부하거나 통제한다.
- ④ 내부 호스트 시스템으로의 접근을 통제할 수 있다.
- ⑤ 내부 사용자의 해킹을 완벽히 방지할 수 있다.

7. 전 세계의 인구가 40억이고 서로 다른 이름을 가지고 있다고 가정할 때, 이 사람들의 이름을 영어의 알파벳으로 입력하여 데이터 베이스를 구축한 결과 완전 이진 트리(Complete Binary Tree)로 표현되었다. 사람 이름으로 검색을 수행할 때, 최악의 경우 데이터 베이스 검색시간은 얼마나 걸리는가? (단, 데이터 베이스에서 한 개의 데이터를 찾아서 비교할 때 소요되는 시간은 1 msec라고 가정한다.)

- ① 31 sec 32 sec 4,000,000 sec
- ② 31 msec 32 msec

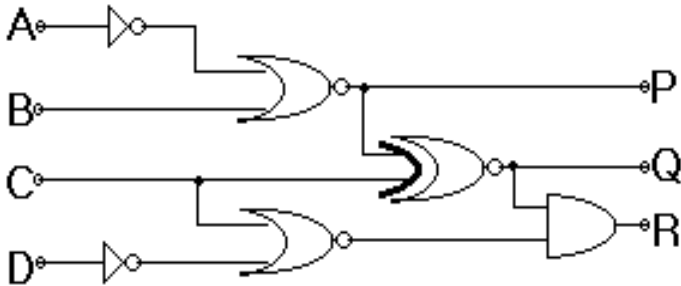
8. 알고리즘은 주어진 문제를 해결하기 위한 절차를 명령어의 집합으로 표현한 것을 말한다. 알고리즘이 갖추어야 할 조건으로 옳지 않은 것은?

- ① 입력되는 자료는 있을 수도 있고, 없을 수도 있다.
- ② 출력 결과를 적어도 한가지 이상 생성해야 한다.
- ③ 각 명령들은 명확하고 모호하지 않아야 한다.
- ④ 여러 명령들이 무한히 반복될 수 있다.
- ⑤ 실제로 프로그래밍이 가능한 것이어야 한다.

9. 다음 용어에 대한 설명 중 옳지 않은 것은?

- ① 영상정보를 사용자가 원하는 시간에, 원하는 장소에서, 원하는 내용을 실시간으로 서비스를 받을 수 있는 것을 주문형 비디오(VOD)라고 한다.
- ② 컴퓨터가 만들어낸 영상을 다양한 감각 채널을 통하여 현실 세계와 자연스러운 상호작용이 가능하도록 한 제반 기술을 가상 현실(Virtual Reality)이라고 한다.
- ③ 스크립트 언어에서 서버 측 스크립트(Server Side Script)는 서버를 거치지 않고 내장된 번역기에 의해 실행되는 스크립트를 말한다.
- ④ 확장된 기능을 제공하기 위해 클라이언트 측에 추가되는 소프트웨어 모듈을 플러그인(Plug-in)이라고 한다.
- ⑤ 웹 브라우저로부터의 입력을 바탕으로 서버상의 프로그램을 수행시켜 결과를 얻어내는 방법을 CGI(Common Gate Interface)라고 한다.

10. 다음 논리 회로에서 A, B, C, D 입력단자에 각각 1이 입력될 때, 출력단자 P, Q, R의 값은?



- ① P=0, Q=0, R=0 P=1, Q=0, R=0
- ② P=0, Q=1, R=0 P=0, Q=0, R=1
- ③ P=1, Q=0, R=1

11. 다음 중 찾고자하는 데이터를 외부 네트워크에서 검색하기 전에 지정된 서버에 이미 저장된 데이터를 검색하여 읽도록 함으로써 전송 효율을 높이는 기능을 담당하는 장치는?

- ① Cache Server Proxy Server
- ② Domain Name Server Buffer Server
- ③ POP Server

12. $[x]$ 는 x 보다 크지 않은 최대 정수를 나타낼 때, $\left[\frac{1 \cdot 20}{21}\right] + \left[\frac{2 \cdot 20}{21}\right] + \dots + \left[\frac{20 \cdot 20}{21}\right]$ 의 값은 190이다. 그런 데 프로그램을 다음과 같이 작성하여 실행한 결과 200이 나왔다. 잘못된 곳은?

```

DIM A AS DOUBLE      ← ㉠
DIM H AS INTEGER      ← ㉡
DIM SUM AS INTEGER      ← ㉢
CLS
FOR A = 1 TO 20
    H = (A * 20) / 21
    H = INT(H)      ← ㉣
    SUM = SUM + H      ← ㉤
NEXT
PRINT SUM
END
    
```

- ① ㉠ ㉡ ㉢ ㉣ ㉤

13. 다음은 파스칼 삼각형을 출력하는 프로그램이다. 잘못된 부분은?

```

DIM p(6, 11)
p(1, 6) = 1: p(6, 1) = 1: p(6, 11) = 1
FOR i = 2 TO 6      ← ㉠
    FOR j = 2 TO 11      ← ㉡
        k = I - 1: m = j - 1
        n = j + 1      ← ㉢
        p(i, j) = p(k, m) + p(k, n)      ← ㉣
    NEXT j
NEXT i
FOR i = 1 TO 6
    FOR j = 1 TO 11
        IF p(i, j) <> 0 THEN
            PRINT TAB(3 * j - 3); p(i, j);
        END IF
    NEXT j
    PRINT
NEXT I
    
```

【실행결과】

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
    
```

- ① ㉠ ㉡ ㉢ ㉣ ㉤ 모두 맞음

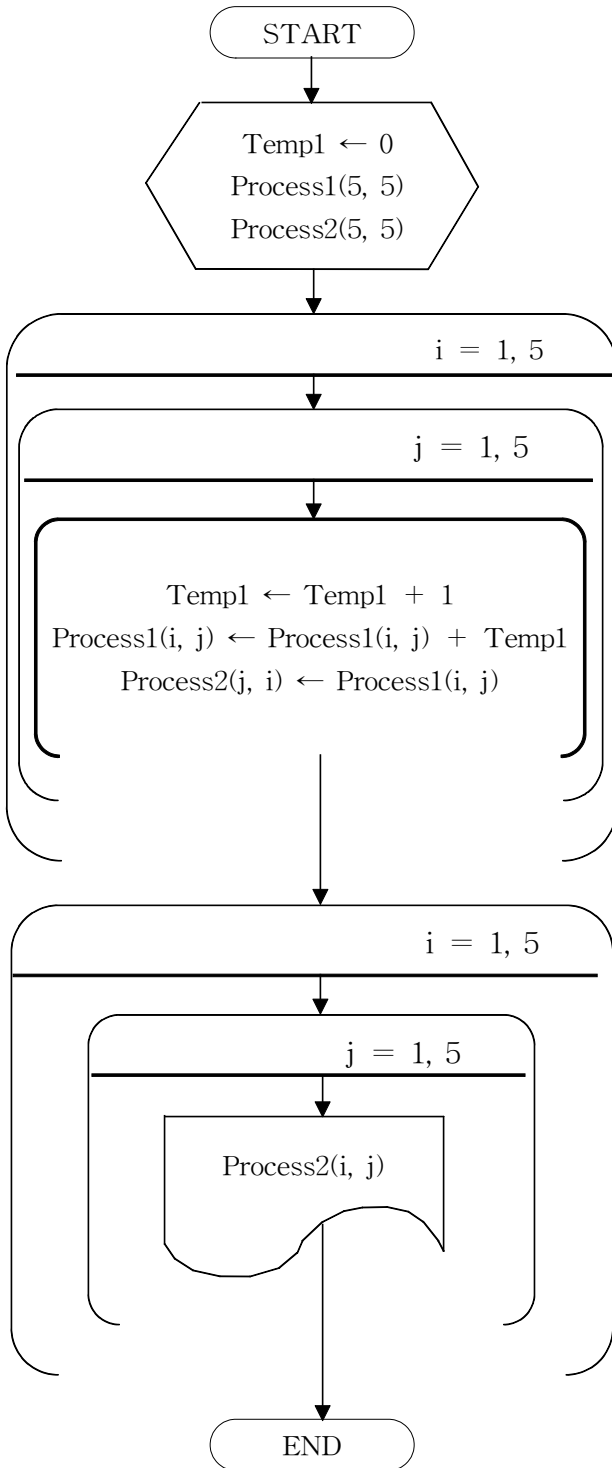
14. 다음은 순열 검색 프로그램이다. 실행 결과는?

```

READ a$
FOR i = 1 TO LEN(a$)
    IF VAL(MID$(a$, i, 1)) < 10 THEN c = c + 1
NEXT i
IF c <> LEN(a$) THEN
    PRINT "E"; : END
END IF
FOR i = 1 TO 10
    c = 0: READ b$
    FOR i = 1 TO LEN(a$)
        IF VAL(MID$(a$, i, 1)) < 10 THEN c = c + 1
    NEXT i
    IF c <> LEN(b$) THEN
        PRINT "E"; : END
    END IF
    d = 0: d = INSTR(a$ + a$, b$)
    IF d = 0 THEN PRINT "D"; ELSE PRINT "S";
NEXT i
DATA 13245768, 68132457, 86754231, 98765432, 81324576
DATA 13245768, 76813245, 02468135, 12345678, 3245768
    
```

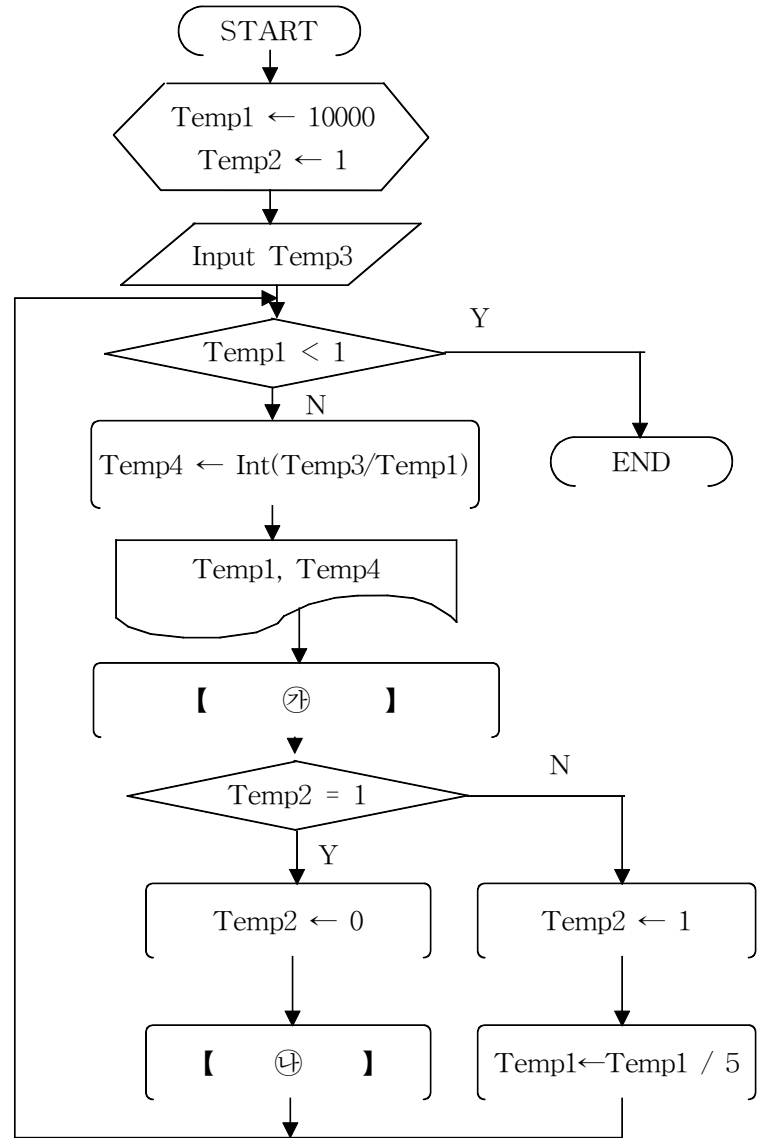
- ① SDSDSSDSE
- ② SSDSSSDSE
- ③ SDDSSSDDE
- ④ SDDSSSSDE
- ⑤ DDDSDDDDE

15. 다음의 순서도를 실행하고 난 후의 출력결과는?



- ①
- | | | | | |
|---|----|----|----|----|
| 1 | 6 | 11 | 16 | 21 |
| 2 | 7 | 12 | 17 | 22 |
| 3 | 8 | 13 | 18 | 23 |
| 4 | 9 | 14 | 19 | 24 |
| 5 | 10 | 15 | 20 | 25 |
- ②
- | | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |
- ③
- | | | | | |
|---|----|----|----|----|
| 1 | 16 | 15 | 14 | 13 |
| 2 | 17 | 24 | 23 | 12 |
| 3 | 18 | 25 | 22 | 11 |
| 4 | 19 | 20 | 21 | 10 |
| 5 | 6 | 7 | 8 | 9 |

16. 다음은 10,000원 이상의 수를 입력하여 10,000원권부터 1원권까지 화폐의 매수를 계산하는 순서도이다. ㉠, ㉡에 적합한 내용은? (단, 화폐의 종류는 10000원, 5000원, 1000원, 500원, 100원, 50원, 10원, 5원, 1원이다.)



- ① ㉠ Temp3 = Temp3 - Temp1 * Temp3
㉡ Temp1 = Temp1 / 10
- ② ㉠ Temp3 = Temp3 - Temp1 * Temp4
㉡ Temp1 = Temp1 / 2
- ③ ㉠ Temp3 = Temp3 - Temp1 * Temp4
㉡ Temp1 = Temp1 / 10
- ④ ㉠ Temp3 = Temp4 - Temp1 * Temp3
㉡ Temp1 = Temp1 / 2
- ⑤ ㉠ Temp3 = Temp4 - Temp1 * Temp3
㉡ Temp1 = Temp1 / 10

17. 다음 프로그램에서 235를 입력했을 때 출력결과는?

```

INPUT KEY$
SUM=0
FOR K=1 TO LEN(KEY$)
  A$=MID$(KEY$, K, 1)
  N= LEN(KEY$)-K
  SUM=VAL(A$)*16^N + SUM
NEXT K
PRINT SUM
END

```

- ① 512 560 565
- ② 570 575

18. TEST.DAT 파일의 내용이 보기와 같을 때, 다음 프로그램의 실행결과 TEST.DAT 파일의 내용으로 적합한 것은?

【보기】

ABCDEFGHIJKLMNQRSTUWXYZ

```
TYPE TRec
  GA AS STRING * 1
  NA AS STRING * 2
  DA AS STRING * 3
END TYPE
DIM FBuffer AS TRec
OPEN "TEST.dat" FOR RANDOM AS #1 LEN = LEN(FBuffer)
FBuffer.GA = "GAFILE"
FBuffer.NA = "NAFILE"
FBuffer.DA = "DAFILE"
PUT #1, 3, FBuffer
CLOSE #1
```

- ① GAFILENAFILEDAFILESTUWXYZ
- ② GAFILENHJKLMNQRSTUWXYZ
- ③ GNADAFGHJKLMNQRSTUWXYZ
- ④ ABCDEFGNADAFMNOPQRSTUWXYZ
- ⑤ ABCDEFGHIJKLGNADAFSTUWXYZ

19. 다음은 보기와 같이 출력하는 프로그램이다. 아래 설명 중 옳은 것은?

【보기】

45 99 1491 74.55

```
OPTION BASE 1
DIM Process(20)      ← ㉠
FOR i = 1 TO 20      ← ㉡
  READ Process(i)
NEXT i
Temp1 = Process(1)
Temp2 = Process(1)
Temp3 = 0
FOR i = 1 TO 20
  IF Process(i) > Temp2 THEN
    Temp2 = Process(i)
  END IF
  IF Process(i) < Temp1 THEN
    Temp1 = Process(i)
  END IF
  Temp3 = Temp3 + Process(i)      ← ㉢
NEXT i
Temp4 = Temp3 / (UBOUND(Process) + 1)      ← ㉣
PRINT Temp1, Temp2, Temp3, Temp4
END
DATA 69,77,74,54,71,64,77,89,85,89,67,45,76,89,57,99,67,88,97,57
```

- ① ㉠ 문장은 Process(19) 라고 표기하여야 한다.
- ② ㉡ 문장은 FOR i=0 TO 19 라고 표기하여야 한다.
- ③ ㉢ 문장은 Process(i) = Process(i) + Temp3 이라고 표기하여야 한다.
- ④ ㉣ 문장에서 제대로 된 결과값을 얻으려면 Temp4 = Temp3 / (UBOUND(Process)) 이라고 표기하여야 한다.
- ⑤ 위의 프로그램은 잘못된 문장이 없다.

20. 다음은 무엇을 수행하는 함수인가?

```
FUNCTION test$ (a$, b$, c$)
  p = 1
  st$ = a$
  n = INSTR(st$, b$)
  DO WHILE n <> 0
    st$ = LEFT$(st$, n - 1) + c$ + MID$(st$, n + LEN(b$))
    p = n + LEN(c$)
    n = INSTR(p, st$, b$)
  LOOP
  test$ = st$
END FUNCTION
```

- ① 문자열에서 지정한 문자열을 새로운 문자열로 바꾸는 함수이다.
- ② 문자열에서 지정한 문자열을 삭제하는 함수이다.
- ③ 두 개의 문자열을 한 개의 문자열로 만드는 함수이다.
- ④ 두 개의 문자열에서 공통적인 문자열을 찾아내는 함수이다.
- ⑤ 두 개의 문자열에서 공통적인 문자열을 삭제하는 함수이다.

21. 다음은 보기 A, B와 같이 실행되는 프로그램이다. ㉠, ㉡에 적합한 내용은?

【보기 A】

Input(0~200) : 5
any number? 2
3

【보기 B】

Input(0~200) : 10
any number? 7
9

```
DIM Array(200) AS INTEGER
DIM Temp1, Temp2, Temp3, Temp4 AS INTEGER
CLS
INPUT "Input(0~200) : "; Temp4
IF Temp4 > 200 OR Temp4 < 0 THEN
  PRINT "Input Error..."
  INPUT "Input(0~200) : "; Temp4
END IF
INPUT "any number "; Temp3
FOR i = 1 TO Temp4
  Array(i) = 1
NEXT i
Temp1 = 0: Temp2 = 0
FOR i = 1 TO Temp4
  IF Array(i) = 1 THEN 【 ㉠ 】
    IF Temp2 = Temp3 THEN
      Temp2 = 0
      Array(i) = 0
      Temp1 = Temp1 + 1
    END IF
    IF 【 ㉡ 】 THEN EXIT FOR
    IF i = Temp4 THEN i = 0
  NEXT i
FOR i = 1 TO Temp4
  IF Array(i) = 1 THEN PRINT i
NEXT i
END
```

- ① ㉠ Temp2 = Temp2 + 1 ㉡ Temp1 = (Temp4 + 1)
- ② ㉠ Temp2 = Temp2 - 1 ㉡ Temp1 = (Temp4 + 1)
- ③ ㉠ Temp2 = Temp2 - 1 ㉡ Temp1 = (Temp4 - 1)
- ④ ㉠ Temp2 = Temp2 + 1 ㉡ Temp1 = Temp4
- ⑤ ㉠ Temp2 = Temp2 + 1 ㉡ Temp1 = (Temp4 - 1)

22. 다음은 알파벳의 빈도수를 구하는 프로그램이다. ㉠, ㉡에 적합한 내용은?

```
CLS
DIM Array(1 TO 26)
PRINT STRING$(71, "-")
PRINT SPACE$(16); "***** 사용된 알파벳 빈도수 *****"
PRINT STRING$(71, "-")
READ Temp$
DO
    FOR i = 1 TO LEN(Temp$)
        Temp1 = ASC(MID$(Temp$, i, 1)) - 96
        IF 【 ㉠ 】 THEN 【 ㉡ 】
    NEXT i
    READ Temp$
LOOP UNTIL LCASE$(Temp$) = "end"
FOR i = 1 TO 26
    IF Array(i) <> 0 THEN
        PRINT CHR$(i + 96); ":", Array(i); "Characters",
    END IF
NEXT i
PRINT STRING$(71, "-")
END
DATA "welcome to seoul information olympiad"
DATA "you are best programmer"
DATA "end"
```

- ① ㉠ Temp1 >= 1 AND Temp1 <= 26
㉡ Array(i) = Array(i) + 1
- ② ㉠ Temp1 >= 1 OR Temp1 <= 26
㉡ Array(i) = Array(i) - 1
- ③ ㉠ Temp1 < 1 AND Temp1 > 26
㉡ Array(i) = Array(i) - 1
- ④ ㉠ Temp1 >= 1 OR Temp1 <= 26
㉡ Array(Temp1) = Array(Temp1) - 1
- ⑤ ㉠ Temp1 >= 1 AND Temp1 <= 26
㉡ Array(Temp1) = Array(Temp1) + 1

23. 다음은 함수의 해를 구하는 프로그램이다. ㉠, ㉡에 적합한 내용은?

```
DECLARE FUNCTION fl# (i#)
DIM a AS DOUBLE
DIM b AS DOUBLE
DIM x AS DOUBLE
DIM n AS DOUBLE
DIM p AS DOUBLE
DIM t AS DOUBLE
a = 1.7
b = 3
p = 100
t = .0000000000001#
n = 1
DO
    x = (a + b) / 2
    IF fl(a) * fl(x) = 0 THEN
        EXIT DO
    ELSE
        IF fl(a) * fl(x) < 0 THEN
            【 ㉠ 】
        ELSE
            【 ㉡ 】
        END IF
    END IF
END IF
```

```
n = n + 1
LOOP WHILE ABS(a - b) > t AND n < p
PRINT b
END

FUNCTION fl# (i#)
    fl# = i# ^ 2 - 3 * i# + 2
END FUNCTION
```

【실행결과】

2.000000000000002

- ① ㉠ a = x ㉡ b = x ㉢ a = b ㉣ b = a
- ② ㉠ b = x ㉡ a = x ㉢ b = a ㉣ a = b
- ③ ㉠ a = a + x ㉡ b = b + x

24. 다음 프로그램의 실행 결과는?

```
CLS
DIM Temp1 AS INTEGER
INPUT "Input "; Temp1
DIM Array1(Temp1 + 1), Array2(2 * Temp1) AS INTEGER
FOR i = 1 TO Temp1
    Array1(i) = i
NEXT i
FOR j = 1 TO 2 * Temp1
    Array2(j) = 255
NEXT j
FOR k = 1 TO Temp1
    Temp2 = Temp1 + 1 - k
    FOR l = 1 TO k
        Array2(Temp2) = Array1(l)
        Temp2 = Temp2 + 1
        Array2(Temp2) = 255
        Temp2 = Temp2 + 1
    NEXT l
    FOR m = 1 TO 2 * Temp1
        IF Array2(m) = 255 THEN PRINT " ";
        IF Array2(m) <> 255 THEN PRINT Array2(m);
    NEXT m
    PRINT
NEXT k
END
```

- ① Input : 4

4	3	2	1
	3	2	1
		1	2
			1
- ② Input : 4

10	9	8	7
	6	5	4
		3	2
			1
- ③ Input : 4

			1
		2	3
	4	5	6
7	8	9	10
- ④ Input : 4

			1
		1	2
	1	2	3
1	2	3	4
- ⑤ Input : 4

0	0	0	1	0	0	0
0	0	1	0	2	0	0
0	1	0	2	0	3	0
1	0	2	0	3	0	4

25. 다음은 보기와 같이 실행되는 프로그램이다. ㉠, ㉡에 적합한 내용은?

【 보기 】

big number? 1234	
small number? 16	
77	2

```
DIM Q AS INTEGER
DIM B AS INTEGER
DIM S AS INTEGER
DIM R AS INTEGER
CLS
INPUT "big number"; B
INPUT "small number"; S
WHILE B >= S
    ㉠
    B = B - S
WEND
    ㉡
PRINT Q, R;
END
```

- ① ㉠ Q=Q+1 ㉡ R=R-1 ㉠ Q=Q-1 ㉡ R=R+1
② ㉠ Q=Q+1 ㉡ R=R+1 ㉠ Q=Q-1 ㉡ R=B
③ ㉠ Q=Q+1 ㉡ R=B

26. 다음 프로그램의 출력 중 8번째로 출력되는 내용은?

```
OPTION BASE 1
DECLARE SUB Process1 (Array1(), Temp1)
DECLARE FUNCTION Process2 (Array2(), Temp2)
Temp1 = 10
DIM Array1(Temp1)
CLS
FOR i = 1 TO Temp1
    READ Array1(i)
NEXT i
CALL Process1(Array1(), Temp1)
DATA 7,4,8,2,9,5,1,0,3,6

SUB Process1 (Array1(), Temp1)
    Temp3 = INT(Temp1 / 2)
    DO
        DO
            Temp4 = 0
            FOR i = 1 TO Temp1 - Temp3
                IF Array1(i) < Array1(i + Temp3) THEN
                    SWAP Array1(i), Array1(i + Temp3)
                    Temp4 = 1
                    a = Process2(Array1(), Temp1)
                END IF
            NEXT i
        LOOP WHILE Temp4
        Temp3 = INT(Temp3 / 2)
    LOOP WHILE Temp3
END SUB

FUNCTION Process2 (Array2(), Temp2)
    FOR i = 1 TO Temp2
        PRINT USING "###"; Array2(i);
    NEXT i
    PRINT
END FUNCTION
```

- ① 9 5 8 4 7 3 2 6 1 0
② 9 5 8 4 7 6 2 3 1 0
③ 9 5 8 6 7 4 2 3 1 0
④ 9 6 8 5 7 4 2 3 1 0
⑤ 9 8 6 5 7 4 2 3 1 0

27. 다음은 프로그램 A와 B를 비교하여 설명한 것이다. 올바른 것은?

【프로그램 A】

```
DIM Process(10)
CLS
FOR i = 1 TO 10
    READ Process(i)
NEXT i
FOR i = 1 TO 9
    FOR j = i + 1 TO 10
        IF Process(i) < Process(j) THEN
            SWAP Process(i), Process(j)
        END IF
    NEXT j
NEXT i
FOR i = 1 TO 10
    PRINT Process(i)
NEXT i
END
DATA 55, 76, 95, 48, 72, 87, 61, 69, 86, 90
```

【프로그램 B】

```
DIM Process(10)
CLS
FOR i = 1 TO 10
    READ Process(i)
NEXT i
DO
    Temp1 = 0
    Temp2 = Temp2 + 1
    FOR j = 1 TO 9
        IF Process(Temp2) > Process(j) THEN
            SWAP Process(Temp2), Process(j)
            Temp1 = 1
        END IF
    NEXT j
    IF Temp2 = 10 THEN
        EXIT DO
    END IF
LOOP WHILE Temp1 = 1
FOR i = 1 TO 10
    PRINT Process(i)
NEXT i
END
DATA 55, 76, 95, 48, 72, 87, 61, 69, 86, 90
```

- ① A 프로그램을 실행하면 교환이 30번 이루어진다.
② B 프로그램을 실행하면 교환이 28번 이루어진다.
③ A 프로그램은 선택(Selection) 정렬 알고리즘이다.
④ B 프로그램은 힙(Heap) 정렬 알고리즘이다.
⑤ A 프로그램은 정렬 알고리즘 중 가장 효율이 좋다.

28. 다음 프로그램들의 실행결과가 같지 않은 것은?

①

```
Cnt = 1
S=0
INPUT "Number="; N
DO WHILE Cnt <= N
    S=S+Cnt
    Cnt = Cnt + 1
LOOP
PRINT S
END
```

```
Cnt = 0
S=0
INPUT "Number="; N
DO
    Cnt = Cnt + 1
    S=S+Cnt
LOOP WHILE Cnt <= N
PRINT S
END
```

②

```
Cnt = 0
S = 0
INPUT "Number="; N
DO
    Cnt = Cnt + 1
    S = S + Cnt
LOOP UNTIL Cnt >= N
PRINT S
END
```

```
Cnt = 1
S=0
INPUT "Number="; N
DO
    S=S+Cnt
    Cnt = Cnt + 1
LOOP UNTIL Cnt > N
PRINT S
END
```

③

```
Cnt = 1
S=0
INPUT "Number="; N
WHILE Cnt<=N
    S=S+Cnt
    Cnt = Cnt + 1
WEND
PRINT S
END
```

29. 다음은 10진수를 입력받아 2진수로 변환하는 프로그램이다. ㉠, ㉡, ㉢에 적합한 내용은?

```
DECLARE FUNCTION Fun1$ (Temp2 AS INTEGER)
DIM Temp1 AS INTEGER
DO WHILE 1
    INPUT "Input : "; Temp1
    IF Temp1 = 0 THEN END
    PRINT Fun1$(Temp1)
LOOP

FUNCTION Fun1$ (Temp2 AS INTEGER)

DIM Array(1 TO 8) AS INTEGER
FOR i = 1 TO 8
    Array(i) = ㉠
    Temp2 = ㉡
NEXT i
Temp3$ = ""
FOR i = 8 TO 1 STEP -1
    Temp3$ = Temp3$ + ㉢
NEXT i
Fun1$ = Temp3$
END FUNCTION
```

- ① ㉠ Temp2 / 2 ㉡ Temp2 MOD 2
 ㉢ RIGHT\$(STR\$(Array(i)), 1)
- ② ㉠ Temp2 / 2 ㉡ Temp2 MOD 2
 ㉢ LEFT\$(STR\$(Array(i)), 1)
- ③ ㉠ Temp2 MOD 2 ㉡ Temp2 / 2
 ㉢ LEFT\$(STR\$(Array(i)), 1)
- ④ ㉠ Temp2 MOD 2 ㉡ Temp2 \ 2
 ㉢ LEFT\$(STR\$(Array(i)), 1)
- ⑤ ㉠ Temp2 MOD 2 ㉡ Temp2 \ 2
 ㉢ RIGHT\$(STR\$(Array(i)), 1)

30. 다음은 영어 학습시간은 1시간, 논술학습시간은 2시간이라 할 때 n시간 동안의 학습시간표를 작성하는 방법의 수 a_n 을 구하는 프로그램이다. 예를 들어 a_5 의 값이 보기와 같다. ㉠, ㉡에 적합한 내용은?

【보기】

5시간 동안의 학습 시간표 작성 방법
(영어, 영어, 영어, 영어, 영어), (영어, 영어, 영어, 논술),
(영어, 영어, 논술, 영어), (영어, 논술, 영어, 영어),
(논술, 영어, 영어, 영어), (영어, 논술, 논술),
(논술, 영어, 논술), (논술, 논술, 영어)
그러므로 $a_5=8$ 이다.

```
DECLARE SUB work (cnt AS INTEGER)
DIM SHARED time1, time2 AS INTEGER
DIM SHARED n AS INTEGER, result AS INTEGER
INPUT "input study time"; n
time1 = 1: time2 = 2: result = 0
IF n < 3 THEN
    result = 1
ELSE work 3
END IF
PRINT STR$(n) + "'s study time = "; result

SUB work (cnt AS INTEGER)
    ㉠
    time1 = time2
    ㉡
    IF cnt = n THEN
        EXIT SUB
    ELSE
        cnt = cnt + 1
        work cnt
    END IF
END SUB
```

【실행결과】

```
input study time? 5
5's study time = 8
```

- ① ㉠ time2=time1 ㉡ result=time2-time1
- ② ㉠ time2=time1+time2 ㉡ result=time1-time2
- ③ ㉠ result=time1-time2 ㉡ time2=result+time1
- ④ ㉠ time2=result ㉡ result=time1+time2
- ⑤ ㉠ result=time1+time2 ㉡ time2=result

31. 다음 프로그램의 실행결과는?

```
CLS
Temp1 = 1
WHILE Temp1 < 11
  READ Temp2
  Temp3 = Temp2 / 4 - INT(Temp2 / 4)
  IF Temp3 <> 0 THEN
    Temp4 = Temp4 + 1
  END IF
  Temp1 = Temp1 + 1
WEND
PRINT Temp4
END
DATA 3,8,12,15,36,22,41,26,33,32
```

- ① 2 4 5 6 10

32. 다음은 보기와 같이 출력되는 프로그램이다. ㉠, ㉡에 적합한 내용은?

【 보기 】			
1	15	14	4
12	6	7	9
8	10	11	5
13	3	2	16

```
DIM Process1(4, 4)
CLS
FOR i = 1 TO 4
  FOR j = 1 TO 4
    IF i = j OR i = 5 - j THEN
      Process1(i, j) = 【 ㉠ 】
    ELSE
      Process1(i, j) = 【 ㉡ 】
    END IF
  NEXT j
NEXT i
FOR i = 1 TO 4
  FOR j = 1 TO 4
    PRINT USING "####"; Process1(i, j);
  NEXT j
  PRINT
NEXT i
```

- ① ㉠ (i - 1) * 4 + j
 ㉡ (4 - i) * 4 + (5 - j)
- ② ㉠ (i - 1) * 5 + j
 ㉡ (5 - i) * 4 + (5 - j)
- ③ ㉠ (i + 1) * 4 + j
 ㉡ (4 - i) * 4 + (4 - j)
- ④ ㉠ (i - 1) * 4 + j
 ㉡ (4 - i) * 4 + (4 - j)
- ⑤ ㉠ (i + 1) * 5 + j
 ㉡ (5 - i) * 4 + (4 - j)

33. 다음은 가위, 바위, 보 게임을 처리 조건에 따라 구현한 프로그램과 실행 결과이다. ㉠, ㉡에 적합한 내용은?

【처리 조건】

- ▶ 가위는 1, 바위는 2, 보는 3으로 한다.
- ▶ 게임은 사람이 입력한 가위, 바위, 보의 숫자와 컴퓨터가 선택한 가위, 바위, 보와의 대결로 한다.
- ▶ 컴퓨터는 난수를 발생하여 가위, 바위, 보를 결정한다.

```
a1$ = "가위"
a2$ = "바위"
a3$ = "보"
DO
PRINT "(가위=1, 바위=2, 보=3, 종료=4)";
DO
  c$ = INKEY$
  LOOP WHILE c$ = "" OR VAL(c$) > 4 OR VAL(c$) < 1
  x = VAL(c$)
  PRINT
  SELECT CASE x
    CASE 1
      x1$ = a1$
    CASE 2
      x1$ = a2$
    CASE 3
      x1$ = a3$
    CASE 4
      EXIT DO
  END SELECT
  y = 【 ㉠ 】
  y = y + 1
  SELECT CASE y
    CASE 1
      y1$ = a1$
    CASE 2
      y1$ = a2$
    CASE 3
      y1$ = a3$
  END SELECT
  IF x MOD 3 = y MOD 3 THEN
    w$ = "tie"
  ELSE
    IF 【 ㉡ 】 THEN
      w$ = "you win"
    ELSE w$ = "com win"
    END IF
  END IF
  PRINT "you            computer"
  PRINT x1$; TAB(15); y1$
  PRINT TAB(4); "****"; w$; "****"
  PRINT
LOOP WHILE (1)
END
```

【실행결과】

```
(가위=1, 바위=2, 보=3, 종료=4) 1
you            computer
가위            보
***you win***

(가위=1, 바위=2, 보=3, 종료=4) 4
```

- ① ㉠ int(rnd(3)*10) mod 3
 ㉡ x mod 3=(y+1) mod 3
- ② ㉠ int(rnd(3)) mod 3
 ㉡ x mod 3<(y+1)
- ③ ㉠ int(rnd(3)*10) mod 3
 ㉡ x>y and x mod 3>(y+1) mod 3
- ④ ㉠ int(rnd(3)) mod 3
 ㉡ x mod 3=(y-1) mod 3
- ⑤ ㉠ int(rnd(3)*10) mod 3
 ㉡ x mod 3=y mod 3

34. 다음은 임의의 연도를 입력하면 육십갑자를 출력하고, 육십갑자를 입력하면 해당 연도를 출력하는 프로그램이다. ㉠, ㉡, ㉢에 적합한 내용은? (단, 입력 연도의 범위는 1900년~2000년으로 하며, 갑자 년은 1924년과 1984년이다.)

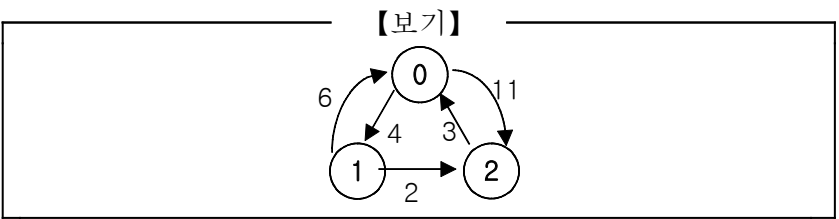
```

DECLARE SUB Process1 ()
DECLARE SUB Process2 ()
DECLARE SUB Process3 ()
DIM SHARED Array1$(1 TO 10)
DIM SHARED Array2$(1 TO 12)
DIM SHARED Array3$(1 TO 60)
DIM SHARED Array4(1 TO 1000)
Process1
Process2
Process3
END
DATA "갑", "을", "병", "정", "무", "기", "경", "신", "임", "계"
DATA "자", "축", "인", "묘", "진", "사", "오", "미", "신", "유", "술", "해"
SUB Process1
    FOR i = 1 TO 10
        READ Array1$(i)
    NEXT i
    FOR i = 1 TO 12
        READ Array2$(i)
    NEXT i
END SUB
SUB Process2
    FOR i = 1 TO 60
        【 ㉠ 】
        IF Temp1 > 10 THEN Temp1 = 1
        Temp2 = Temp2 + 1
        IF Temp2 > 12 THEN Temp2 = 1
        Array3$(i) = Array1$(Temp1) + Array2$(Temp2)
    NEXT i
END SUB
SUB Process3
    INPUT "연도입력 : 1    육십갑자입력 : 2"; Temp3
    IF Temp3 = 0 THEN END
    SELECT CASE Temp3
    CASE 1
        INPUT "연도 = "; Temp4
        【 ㉡ 】
        Temp5 = Temp5 - 3
        IF Temp5 < 1 THEN Temp5 = 60 - Temp5
        PRINT "-----> "; Array3$(Temp5); "년"
    CASE 2
        INPUT "육십갑자 = "; Temp6$
        FOR i = 1 TO 60
            IF Temp6$ = Array3$(i) THEN Temp4 = i
        NEXT i
        FOR i = 1900 TO 2000
            Temp7 = i MOD 60
            【 ㉢ 】
            IF Temp7 = Temp4 THEN
                Temp8 = Temp8 + 1
                Array4(Temp8) = i
            END IF
        NEXT i
        PRINT " ----- ";
        FOR i = 1 TO Temp8
            PRINT Array4(i); "년 ";
        NEXT i
        PRINT " ----- "
    END SELECT
END SUB

```

- ① ㉠ Temp1 = Temp1 + 1
 ㉡ Temp5 = Temp4 MOD 60
 ㉢ Temp7 = Temp7 - 3
- ② ㉠ Temp1 = Temp1 + 1
 ㉡ Temp5 = Temp4 MOD 4
 ㉢ Temp7 = Temp7 + 3
- ③ ㉠ Temp1 = Temp1 - 1
 ㉡ Temp5 = Temp4 MOD 60
 ㉢ Temp7 = Temp7 + 3
- ④ ㉠ Temp1 = Temp1 - 1
 ㉡ Temp5 = Temp4 MOD 400
 ㉢ Temp7 = Temp7 - 3
- ⑤ ㉠ Temp1 = Temp1 + 1
 ㉡ Temp5 = Temp4 MOD 400
 ㉢ Temp7 = Temp7 - 3

35. 다음은 보기와 같은 방향 그래프에서 각 정점간에 최단 거리를 구하는 프로그램이다. ㉠에 적합한 내용은?



```

n = 3
DIM length(0 TO 2, 0 TO 2) AS INTEGER
DIM a(0 TO 2, 0 TO 2) AS INTEGER
FOR i = 0 TO 2
    FOR j = 0 TO 2
        READ length(i, j)
    NEXT j
NEXT i
DATA 0,4,11,6,0,2,3,999,0
FOR i = 0 TO n - 1
    FOR j = 0 TO n - 1
        a(i, j) = length(i, j)
    NEXT j
NEXT i
FOR k = 0 TO n - 1
    FOR i = 0 TO n - 1
        FOR j = 0 TO n - 1
            【 ㉠ 】
        NEXT j
    NEXT i
NEXT k
FOR i = 0 TO 2
    FOR j = 0 TO 2
        PRINT STR$(i); "->"; STR$(j); " : "; STR$(a(i, j))
    NEXT j
NEXT i

```

- ① a(i,j) = a(i,k) + a(k,j)
- ② a(i,j) = a(i,j) + a(i,k) + a(j,k)
- ③ a(i,j) = a(j,k) + a(k,i)
- ④ IF a(k, j) < a(i, j) THEN a(i, j) = a(k, j)
- ⑤ IF (a(i, k) + a(k, j)) < a(i, j) THEN a(i, j) = a(i, k) + a(k, j)

36. 다음 프로그램을 실행하였을 때 입력내용에 따른 출력결과가 옳지 않은 것은?

```

DECLARE SUB Process2 ()
DECLARE SUB Process1 ()
DIM SHARED Array1(5), Array2(5), Temp5
Array1(0) = 0
Array1(1) = 4
Array1(2) = 2
Array1(3) = 3
Array1(4) = 1
INPUT "입력내용 : "; Temp1, Temp2
Temp3 = 0
Temp4 = 0
Temp5 = Temp1
Process1
FOR i = Array2(0) TO 1 STEP -1
    Temp3 = Temp3 * 5 + Array1(Array2(i))
NEXT i
Temp5 = Temp2
Process1
PRINT
FOR i = Array2(0) TO 1 STEP -1
    Temp4 = Temp4 * 5 + Array1(Array2(i))
NEXT i
Temp6 = Temp3 + Temp4: Temp5 = Temp6
Process2
PRINT
PRINT "출력결과 : ";
FOR i = Array2(0) TO 1 STEP -1
    PRINT Array1(Array2(i));
NEXT i
PRINT
END

SUB Process1
    Temp7 = 1
    DO WHILE (1)
        Temp8 = INT(Temp5 / 10)
        Temp9 = Temp5 MOD 10
        Array2(Temp7) = Temp9
        IF Temp8 = 0 THEN
            Array2(0) = Temp7
            EXIT SUB
        END IF
        Temp5 = Temp8
        Temp7 = Temp7 + 1
    LOOP
END SUB

SUB Process2
    Temp7 = 1
    DO WHILE (1)
        Temp8 = INT(Temp5 / 5)
        Temp9 = Temp5 MOD 5
        Array2(Temp7) = Temp9
        IF Temp8 = 0 THEN
            Array2(0) = Temp7
            EXIT SUB
        END IF
        Temp5 = Temp8
        Temp7 = Temp7 + 1
    LOOP
END SUB
    
```

- | | |
|------------------|----------------|
| ① 입력내용 : 142,431 | 입력내용 : 13,3421 |
| 출력결과 : 4 4 0 4 | 출력결과 : 3 2 2 2 |
| ② 입력내용 : 250,150 | 입력내용 : 154,300 |
| 출력결과 : 4 4 0 0 | 출력결과 : 4 2 0 3 |
| ③ 입력내용 : 400,402 | |
| 출력결과 : 2 0 2 | |

37. 다음은 사다리 타기 게임 프로그램이다. 처리조건이 다음과 같을 때 ㉠, ㉡, ㉢, ㉣에 적합한 내용은?

【처리조건】

- ▶ 게임 하는 사람의 수와 지정할 사다리 가로대의 번호를 입력한다.
- ▶ 사다리 가로대는 인접되어 있는 가로대와 겹치게 만들 수는 없다.
- ▶ 만약 3명이 게임을 하고 첫 번째 칸은 1, 3, 5, 7번, 두 번째 칸은 2, 4, 6번에 가로대를 설치하려면 다음과 같이 입력한다. 그러면 보기와 같이 사다리가 그려진다.

【보기】

Man,FirstLine,SecondLine= 3,1357,246

	[A]	[B]	[C]
	I	I	I
가로대의 번호 → 1	IIIIII		I
	I	I	I
2	I	IIIIII	
	I	I	I
3	IIIIII		I
	I	I	I
4	I	IIIIII	
	I	I	I
5	IIIIII		I
	I	I	I
6	I	IIIIII	
	I	I	I
7	IIIIII		I
	I	I	I
8	I	I	I
	I	I	I
9	I	I	I
	I	I	I

Start Man(a or b....) =

```

DECLARE SUB Process1 ()
DECLARE SUB Process2 ()
DECLARE SUB Process3 (Temp1, Temp2)
CLS
DIM SHARED Temp$
DIM SHARED Temp3$, Temp4$, Temp4
Process1
Cnt = 0
FOR i = 1 TO LEN(Temp$)
    Temp5$ = MID$(Temp$, i, 1)
    IF Temp5$ <> "," THEN
        Temp6$ = Temp6$ + Temp5$
        IF i = LEN(Temp$) THEN
            Cnt = Cnt + 1: Temp5$(Cnt) = Temp6$: Temp6$ = ""
        END IF
    ELSE
        Cnt = Cnt + 1: Temp5$(Cnt) = Temp6$: Temp6$ = ""
    END IF
NEXT i
    
```

```

j = 3
DO WHILE j <= Cnt
  FOR k = 1 TO LEN(Temp5$(j - 1))
    FOR l = 1 TO LEN(Temp5$(j))
      IF MID$(Temp5$(j - 1), k, 1) = MID$(Temp5$(j), l, 1) THEN
        Process1
        Cnt = 0: i = 0: EXIT DO
      END IF
    NEXT l
  NEXT k
  j = j + 1
LOOP
NEXT i
Temp2 = VAL(Temp5$(1))
Temp1 = (Temp2 - 1) * 7 + 1
DIM SHARED Array$(Temp1, 19)
FOR i = 1 TO 19
  FOR j = 1 TO Temp1
    IF j = 1 OR (j - 1) / 7 = INT((j - 1) / 7) THEN
      Array$(j, i) = "I"
    ELSE Array$(j, i) = " "
    END IF
  NEXT j
NEXT i
FOR i = 1 TO Temp2 - 1
  FOR j = 1 TO LEN(Temp5$(i + 1))
    FOR k = 1 TO 6
      Array$((i-1)*7+k+1, VAL(MID$(Temp5$(i+1), j, 1))*2) = "I"
    NEXT k
  NEXT j
NEXT i
CALL Process3(Temp1, Temp2)
DO
  Process2
  IF Temp4 > Temp2 OR Temp4 < 1 THEN
    LOCATE 22, 30: PRINT "<----- Input Error"
    Process2
  END IF
  LOCATE 22, 30: PRINT "<----- Good Data  "
  Temp7 = (Temp4 - 1) * 7 + 1: Temp9 = 1
  Linechar$ = CHR$(Temp4 + 96)
  Array$(Temp7, Temp9) = Linechar$: Temp8 = Temp7
  FOR i = 2 TO 19
    Temp7 = Temp8: Array$(Temp7, i) = Linechar$
    IF Temp7 = 1 THEN
      IF Array$(Temp7 + 1, i) <> " " THEN
        FOR j = 1 TO 7
          【 ㉔ 】
        NEXT j
        Temp8 = Temp7 + 7
      END IF
    ELSEIF Temp7 = Temp1 THEN
      IF Array$(Temp7 - 1, i) <> " " THEN
        FOR j = 1 TO 7
          【 ㉕ 】
        NEXT j
        Temp8 = Temp7 - 7
      END IF
    ELSEIF Array$(Temp7 + 1, i) <> " " THEN
      FOR j = 1 TO 7
        【 ㉖ 】
      NEXT j
      Temp8 = Temp7 + 7
    ELSEIF Array$(Temp7 - 1, i) <> " " THEN
      FOR j = 1 TO 7
        【 ㉗ 】
      NEXT j
      Temp8 = Temp7 - 7
    END IF
  NEXT i

```

```

END IF
NEXT i
CALL Process3(Temp1, Temp2)
LOOP
SUB Process1
  LOCATE 1, 1
  LINE INPUT "Man,FirstLine,SecondLine = "; Temp$
  IF LCASE$(Temp$) = "end" THEN END
END SUB
SUB Process2
  LOCATE 22, 2: PRINT "Start Man(a or b....) = ";
  Temp3$ = INPUT$(1)
  IF Temp3$ = CHR$(27) THEN END
  PRINT Temp3$: Temp4$ = LCASE$(Temp3$)
  Temp4 = ASC(Temp4$) - 96
END SUB
SUB Process3 (Temp1, Temp2)
  LOCATE 2, 4
  FOR i = 1 TO Temp2
    PRINT "["; CHR$(64 + i); "]"; SPC(4);
  NEXT i
  FOR i = 1 TO 9
    LOCATE (i + 1) * 2, 1: PRINT i
  NEXT i
  FOR i = 1 TO 19
    FOR j = 1 TO Temp1
      LOCATE 2 + i, 4 + j
      PRINT Array$(j, i);
    NEXT j
    PRINT
  NEXT i
END SUB

```

- ① ㉔ Array\$(Temp7 + i, j) = Linechar\$
 ㉕ Array\$(Temp7 - i, j) = Linechar\$
 ㉖ Array\$(Temp7 + i, j) = Linechar\$
 ㉗ Array\$(Temp7 - i, j) = Linechar\$
- ② ㉔ Array\$(Temp7 - i, j) = Linechar\$
 ㉕ Array\$(Temp7 + i, j) = Linechar\$
 ㉖ Array\$(Temp7 - i, j) = Linechar\$
 ㉗ Array\$(Temp7 + i, j) = Linechar\$
- ③ ㉔ Array\$(Temp7 + j, i) = Linechar\$
 ㉕ Array\$(Temp7 - j, i) = Linechar\$
 ㉖ Array\$(Temp7 + j, i) = Linechar\$
 ㉗ Array\$(Temp7 - j, i) = Linechar\$
- ④ ㉔ Array\$(Temp7 - j, i) = Linechar\$
 ㉕ Array\$(Temp7 + j, i) = Linechar\$
 ㉖ Array\$(Temp7 - j, i) = Linechar\$
 ㉗ Array\$(Temp7 + j, i) = Linechar\$
- ⑤ ㉔ Array\$(Temp7 + j, i) = Linechar\$
 ㉕ Array\$(Temp7 - j, i) = Linechar\$
 ㉖ Array\$(Temp7 - j, i) = Linechar\$
 ㉗ Array\$(Temp7 + j, i) = Linechar\$

38. 다음은 풀다운(Pull-Down) 메뉴를 만드는 프로그램이다.

㉠, ㉡, ㉢에 적합한 내용은?

```

DIM Process1$(3)
Temp1 = 1
Temp2 = 1
Process1$(1) = " 파일 "
Process1$(2) = " 편집 "
Process1$(3) = " 저장 "
CLS
FOR i = 1 TO 3
    PRINT Process1$(i)
NEXT i
【 ㉠ 】
WHILE Temp$ <> CHR$(27) AND Temp$ <> CHR$(13)
    COLOR 7, 0
    LOCATE Temp1, 1
    PRINT Process1$(Temp1)
    COLOR 0, 7
    【 ㉡ 】
    PRINT Process1$(Temp2)
    DO
        Temp$ = INKEY$
    LOOP WHILE 【 ㉢ 】
    Temp1 = Temp2
    IF Temp$ = CHR$(0) + CHR$(80) AND Temp2 < 3 THEN
        Temp2 = Temp2 + 1
    END IF
    IF Temp$ = CHR$(0) + CHR$(72) AND Temp2 > 1 THEN
        Temp2 = Temp2 - 1
    END IF
WEND
END
    
```

- | | |
|----------------------|-------------------|
| ① ㉠ Temp\$ = INKEY\$ | ㉡ LOCATE Temp1, 1 |
| ㉢ Temp\$ = "" | |
| ② ㉠ Temp\$ = INPUT\$ | ㉡ LOCATE Temp2, 1 |
| ㉢ Temp\$ = INKEY\$ | |
| ③ ㉠ Temp\$ = INKEY\$ | ㉡ LOCATE Temp2, 1 |
| ㉢ Temp\$ = "" | |
| ④ ㉠ Temp\$ = INPUT\$ | ㉡ LOCATE Temp1, 1 |
| ㉢ Temp\$ = INKEY\$ | |
| ⑤ ㉠ Temp\$ = INKEY\$ | ㉡ LOCATE Temp1, 2 |
| ㉢ Temp\$ = INPUT\$ | |

※ 다음은 저울로 잴 수 있는 돌들의 최대 무게를 측정하는 프로그램이다. 돌의 무게는 여러 종류이며 저울이 잴 수 있는 무게는 한계치가 있다고 할 때, 다음 물음에 답하시오.(39번-40번)

```

DECLARE FUNCTION f1! (idx AS INTEGER, num AS INTEGER)
DECLARE FUNCTION f2! (idx AS INTEGER)
DIM nn(1 TO 100) AS INTEGER
DIM SHARED ct AS INTEGER
DIM id AS INTEGER
c = 110
ct = 8 ← ㉠
REDIM s(1 TO ct) AS INTEGER
REDIM SHARED bg(1 TO 100, 1 TO ct) AS INTEGER
FOR i = 1 TO ct
    READ s(i)
NEXT i
DATA 55,45,43,41,31,27,26,23,15,3
    
```

```

mf = 0
FOR i = 1 TO ct
    bg(i, 1) = s(i)
    nn(i) = 1
NEXT i
FOR id = 1 TO ct
    mm = f2(id)
    IF mm < c THEN
        FOR j = 1 TO ct
            IF (f1(id, 【 ㉡ 】) = 0) AND (mm + 【 ㉢ 】 <= c) THEN
                mm = mm + 【 ㉣ 】
                nn(id) = nn(id) + 1
                bg(id, 【 ㉤ 】) = 【 ㉥ 】
            END IF
        NEXT j
    END IF
    IF mf < mm THEN
        mf = mm: mbg = id
    END IF
NEXT id
FOR id = 1 TO ct
    j = 1
    DO WHILE (bg(id, j) <> 0)
        PRINT USING "###"; bg(id, j);
        j = j + 1
    LOOP
    PRINT
NEXT id
PRINT mf
j = 1
DO WHILE (bg(mbg, j) <> 0)
    PRINT USING "###"; bg(mbg, j);
    j = j + 1
LOOP

FUNCTION f1 (idx AS INTEGER, num AS INTEGER)
    f1 = 0
    FOR i = 1 TO ct
        IF bg(idx, i) = num THEN
            f1 = 1
            EXIT FOR
        END IF
    NEXT i
END FUNCTION

FUNCTION f2 (idx AS INTEGER)
    FOR i = 1 TO ct
        temp = temp + bg(idx, i)
    NEXT i
    f2 = temp
END FUNCTION
    
```

39. 위 프로그램의 ㉠ 부분을 「ct=10」으로 교체하여 실행시킬 때, 출력되는 값끼리 짝지어진 것은 ?

【보기】											
㉠ 110	㉡ 31	55	15	㉢ 15	55	31	3				
㉣ 109	㉤ 3	45	43	15	㉥ 45	55	3				

- | | | |
|-------------|---------|---------|
| ① ㉠-㉡-㉢-㉤-㉥ | ㉡-㉢-㉣-㉤ | ㉠-㉢-㉤-㉥ |
| ② ㉢-㉣-㉥ | ㉣-㉤-㉥ | |

40. 위 프로그램의 ㉡~㉥ 부분 중 네 부분은 「s(j)」가 들어가야 한다. 「s(j)」가 들어가야 할 부분으로 적합한 것은?

- | | | |
|-----------|---------|---------|
| ① ㉡-㉢-㉣-㉤ | ㉡-㉢-㉣-㉥ | ㉡-㉣-㉤-㉥ |
| ② ㉡-㉢-㉤-㉥ | ㉢-㉣-㉤-㉥ | |