

컴퓨터일반	국가 전산 9급	2024년 3월 23일
--------------	-----------------	---------------------

♣ 필기합격인원/합격선(???/??점) - 선발예정인원 118명 ♣

1. 컴퓨터에서 사용하는 정보량의 단위를 크기가 작은 것부터 큰 것 순서대로 바르게 나열한 것은? [2024년 국가 9급]

- ① EB, GB, PB, TB
- ② EB, PB, GB, TB
- ③ GB, TB, EB, PB
- ④ GB, TB, PB, EB

♣ 표준화 단위

• 다음은 컴퓨터에서 사용하는 세계 표준화 단위이다.

이름	뜻		기호
yotta	10^{24}	1,000,000,000,000,000,000,000,000	Y
zetta	10^{21}	1,000,000,000,000,000,000,000,000	Z
exa	10^{18}	1,000,000,000,000,000,000,000,000	E
peta	10^{15}	1,000,000,000,000,000,000,000,000	P
tera	10^{12}	1,000,000,000,000,000,000,000,000	T
giga	10^9	1,000,000,000,000,000,000,000,000	G
mega	10^6	1,000,000,000,000,000,000,000,000	M
kilo	10^3	1,000,000,000,000,000,000,000,000	k
mili	10^{-3}	0.001	m
micro	10^{-6}	0.000 001	μ
nano	10^{-9}	0.000 000 001	n
pico	10^{-12}	0.000 000 000 001	p
femto	10^{-15}	0.000 000 000 000 001	f
atto	10^{-18}	0.000 000 000 000 000 001	a
zepto	10^{-21}	0.000 000 000 000 000 000 001	z
yocto	10^{-24}	0.000 000 000 000 000 000 000 001	y

• 1,024KB(Byte) = 1MB

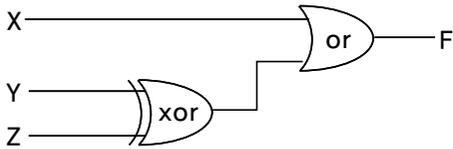
• 1,024MB(Byte) = 1GB

• 1,024GB(Byte) = 1TB

정답 : ④

4 한성미디어 <https://pass25.com>

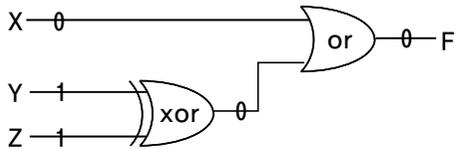
2. 다음 논리회로도에서 출력 F가 0이 되는 입력 조합을 바르게 연결한 것은? [2024년 국가 9급]



- | | | | |
|---|---|---|---|
| | X | Y | Z |
| ① | 0 | 0 | 1 |
| ② | 0 | 1 | 0 |
| ③ | 0 | 1 | 1 |
| ④ | 1 | 0 | 0 |

☞ 논리회로도에서 출력 F가 0이 되는 입력 조합

• 이런 문제는 다음 그림처럼 0과 1을 직접 논리회로에 대입하여 푸는 것이 좋다.



• XOR 연산은 두 입력을 비교해서 같으면 0, 다르면 1이 되는 연산이다.

정답 : ③

3. 암호화 및 복호화를 위하여 개인키와 공개키가 필요한 비대칭키 암호화 기법은? [2024년 국가 9급]

- | | |
|-------|--------|
| ① AES | ② DES |
| ③ RSA | ④ SEED |

☞ 암호시스템

대칭키 암호	DES, SEED, AES, ARIA, HIGHT, IDEA, RC4, SkipJack
비대칭키 암호	RSA, Rabin, ElGamal, ECC, DSA, KCDSA, ECDSA (공개키 암호)
해시함수	MD5, HAS-160, SHA-1, SHA-2, WHIRLPOOL(월풀)

• 공개키 암호는 암호화를 위해서 반드시 한 쌍의 키가 필요하다.(공개키, 개인키)

정답 : ③

4. OSI 모형의 네트워크층 프로토콜에 속하지 않는 것은? [2024년 국가 9급]

- ① ICMP
- ② IGMP
- ③ IP
- ④ SLIP

♣ OSI 모형의 프로토콜

응용층	응용층	DNS, HTTP, FTP, IRC, SSH, TELNET, DHCP SNMP, SMTP, IMAP4, POP3, SOAP
표현층		
세션층	전송층	TCP, UDP, SCTP
전송층		
네트워크층	네트워크층	IP, ICMP, IGMP, ARP, RARP
데이터링크층	데이터링크층	PPP, SLIP, X.25, Frame Relay, ATM, 이더넷, 토큰링
물리층	물리층	RS-232, RS-422, RS-485, USB

[OSI 모델]

[인터넷]

[인터넷에서 사용되는 프로토콜]

정답 : ④

5. 클라우드 컴퓨팅 서비스에서 애플리케이션을 구축, 테스트, 설치할 수 있도록 통합환경을 제공하는 것은? [2024년 국가 9급]

- ① IaaS ② NAS
- ③ PaaS ④ SaaS

♣ 클라우드 컴퓨팅 서비스 형태

- SaaS(Software as a Service) : 이용자가 원하는 소프트웨어를 임대·제공
- PaaS(Platform as a Service) : 소프트웨어 개발에 필요한 플랫폼을 임대·제공 - 통합환경
- IaaS(Infrastructure as a Service) : 서버, 스토리지 등 하드웨어 자원을 임대·제공

정답 : ③

6 한성미디어 <https://pass25.com>

6. 10진수 **빨셈 (7 - 12)**를 2의보수를 이용하여 계산한 결과는? (단, 저장공간은 8비트로 한다)
[2024년 국가 9급]

- ① 0000 0100 ② 0000 0101
- ③ 1111 0101 ④ 1111 1011

☞ 2의보수

• $7 - 12 = -5$ → 연산 결과인 -5를 2의보수 표현으로 2진수로 나타내면 된다.

5 : 0000 0101 _____
 1111 1010 ← _____ 1의보수
+ 1 _____
 1111 1011 ← _____ 2의보수로 -5 ← 이 결과는 사람 생각으로 푸는 것이다.

- 이런 유형의 문제는 사람이 생각하는 방식 또는 컴퓨터 내부 작동 방식으로 풀 수 있다.
- 우리는 사람이므로 사람이 생각하는 방식으로 푸는 것이 좋다.

정답 : ④

7. RAID(Redundant Array of Inexpensive Disks) 레벨에 대한 설명으로 옳지 않은 것은? [2024년 국가 9급]

- ① RAID 레벨 0: 패리티 없이 데이터를 분산 저장한다.
- ② RAID 레벨 1: 패리티 비트를 사용하여 오류를 검출한다.
- ③ RAID 레벨 2: 해밍코드를 사용하여 오류 검출 및 정정이 가능하다.
- ④ RAID 레벨 5: 데이터와 함께 패리티 정보를 블록 단위로 분산 저장한다.

☞ RAID

- RAID 레벨 1: **패리티 비트**를 사용하여 오류를 검출한다.(×)
→ RAID 레벨 1: 패리티(오류 검출 기능) **없음**
- RAID 0 : 적어도 2개의 디스크, **분산**, 패리티(오류 검출 기능) **없음** → **신속한 입출력**
- RAID 1 : 적어도 2개의 디스크, **복사**, 패리티(오류 검출 기능) **없음** → **안정성**
- RAID 2 : 적어도 3개의 디스크, 분산, **비트 레벨**, **전용 해밍코드 패리티** 디스크 사용
- RAID 3 : 적어도 3개의 디스크, 분산, **바이트 레벨**, **전용 패리티** 디스크 사용
- RAID 4 : 적어도 3개의 디스크, 분산, **블록 레벨**, **전용 패리티** 디스크 사용
- RAID 5 : 적어도 3개의 디스크, 분산, **블록 레벨**, **패리티 각 디스크에 분산** → RAID 4 **개선**

정답 : ②

8. RISC와 비교하여 CISC의 특징으로 옳지 않은 것은? [2024년 국가 9급]

- ① 명령어의 종류가 많다.
- ② 명령어의 길이가 고정적이다.
- ③ 명령어 파이프라인이 비효율적이다.
- ④ 회로 구성이 복잡하다.

♣ RISC / CISC

- 명령어의 길이가 고정적이다.(x)
→ CISC : 명령어의 길이가 가변적이다.
- CISC(Complex Instruction Set Computer) : 복잡 명령어 집합 컴퓨터
- RISC(Reduced Instruction Set Computer) : 축소 명령어 집합 컴퓨터

CISC CPU	RISC CPU
<ul style="list-style-type: none"> • CISC CPU는 복잡한 명령어 집합을 가진다. • 명령어 해석에 많은 시간이 소요된다. 	<ul style="list-style-type: none"> • RISC CPU는 명령어 종류와 구성이 단순하다. • 명령어가 하드웨어에서 구현하기 좋다.
<ul style="list-style-type: none"> • CISC는 RISC보다 복잡하고 세밀한 조작을 유연하게 처리할 수 있는 장점이 있다. 	<ul style="list-style-type: none"> • RISC는 많은 레지스터와 캐시를 가질 수 있다. • 메모리 접근회수가 줄어든다.
<ul style="list-style-type: none"> • CISC는 명령어 길이가 다르다.(가변 길이) • 명령어 해석이 복잡하다.(시간 소요) 	<ul style="list-style-type: none"> • RISC는 명령어 길이가 같다.(고정길이 명령어) • 명령어 해석이 간단하다.
<ul style="list-style-type: none"> • CISC는 많은 주소 모드를 갖추고 있다. • CISC는 명령어의 직교성이 좋다. 	<ul style="list-style-type: none"> • RISC는 전력 소모가 적다. • 임베디드 프로세서에 이용된다.
<ul style="list-style-type: none"> • 스택은 하드웨어적으로 지원한다. 	<ul style="list-style-type: none"> • 스택은 소프트웨어적으로 처리한다.

- CISC / RISC는 마이크로프로세서를 설계하는 방법에 의해 분류된다.

// CISC와 RISC의 일반적인 비교

	명령어 길이	캐시 용량	속도	레지스터	전력소모	주소모드	명령어 직교성
CISC	가변길이	적다	느리다	적다	많다	복잡	좋다
RISC	고정길이	많다	빠르다	많다	적다	간단	부족하다

- 직교성은 서로 방해 또는 간섭이 없는 경우이다.
- 상호 의존성이 없을수록 직교성은 높다. 좋은 함수는 우수한 직교성을 가지고 있다.

8 한성미디어 <https://pass25.com>

9. 다음 파이썬 코드는 이진탐색을 이용하여 자연수 데이터를 탐색하는 함수이다. (가), (나)에 들어갈 내용을 바르게 연결한 것은? (단, ds는 오름차순으로 정렬된 중복 없는 자연수 리스트이고, key는 찾고자 하는 값이다) [2024년 국가 9급]

```
def binary(ds, key):
    low = 0
    high = len(ds) - 1
    while low <= high:
        mid = (low+high) // 2
        if key == ds[mid]:
            return mid
        elif key < ds[mid]:
            (가)
        else:
            (나)
    return
```

(가) (나)

- ① high = mid - 1 low = mid - 1
- ② high = mid - 1 low = mid + 1
- ③ high = mid + 1 low = mid - 1
- ④ high = mid + 1 low = mid + 1

☞ 이진탐색 - 파이썬 코드

```
def binary(ds, key):
    low = 0                #첫 번째 자료 인덱스
    high = len(ds) - 1    #마지막 자료 인덱스
    while low <= high:    #반복조건
        mid = (low+high) // 2 #연산자 //는 정수 나눗셈 연산 - 탐색자료 중간 위치 결정
        if key == ds[mid]:    #탐색에 성공한 경우
            return mid        #탐색에 성공한 경우 : 인덱스 mid 반환
        elif key < ds[mid]:    #탐색 자료가 중간 위치의 값보다 작은 경우
            (가) high = mid - 1 #탐색 대상 마지막 자료 인덱스 조정
        else:                 #탐색 자료가 중간 위치의 값보다 큰 경우
            (나) low = mid + 1 #탐색 대상 첫 번째 자료 인덱스 조정
    return                 #탐색에 실패한 경우 : None 반환
```

// 이진검색(binary search) - 정렬된 리스트에서 검색 가능

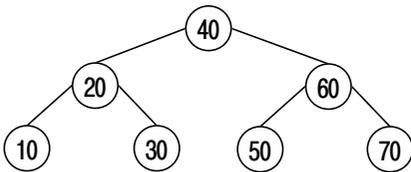
먼저, 이진검색은 리스트의 중간 키값을 검사한다.

만약, 일치하지 않으면 왼쪽 또는 오른쪽 부분만을 같은 방법으로 검색을 계속 실시한다.

	0	1	2	3	4	5	6	← 자료위치(인덱스)
[예]	10	20	30	40	50	60	70	에서 60을 찾는 경우

- 처음 2등분 위치의 자료가 40이므로 검색할 자료는 뒤쪽 부분에 있다.
- 다시 뒤쪽 부분을 2등분하면 검색대상자료 60을 찾게 된다.

이를 이진트리로 표현하여 설명하면 다음과 같다.



L = 0, H = 6

M = $\lfloor (L + H) / 2 \rfloor = \lfloor (0 + 6) / 2 \rfloor = 3$

// 만약, 중간값이 일치하지 않으면

작은쪽	H = M - 1 = 3 - 1 = 2 M = (L + H) / 2 = (0 + 2) / 2 = 1
큰 쪽	L = M + 1 = 3 + 1 = 4 M = (L + H) / 2 = (4 + 6) / 2 = 5

// 이진탐색을 파이썬 코드로 완성하면 다음과 같다.

```

def binary(ds, key):
    low = 0 #첫 번째 자료 인덱스
    high = len(ds) - 1 #마지막 자료 인덱스
    while low <= high: #반복조건
        mid = (low+high) // 2 #연산자 //는 정수 나눗셈 연산 - 탐색자료 중간 위치 결정
        if key == ds[mid]: #탐색에 성공한 경우
            return mid #탐색에 성공한 경우 : 인덱스 mid 반환
        elif key < ds[mid]: #탐색 자료가 중간 위치의 값보다 작은 경우
            high = mid - 1 #탐색 대상 마지막 자료 인덱스 조정
        else: #탐색 자료가 중간 위치의 값보다 큰 경우
            low = mid + 1 #탐색 대상 첫 번째 자료 인덱스 조정
    return #탐색에 실패한 경우 : None 반환
ds = 10, 20, 30, 40, 50, 60, 70 #정렬된 리스트 - 탐색 대상
key = 60 #탐색할 키
print(binary(ds, key)) #탐색 결과 출력 : 5 ← 탐색할 키 20의 인덱스 5
    
```

