


시험에
나오는 것만
공부한다!



원리만 이해하면 쉬운

프로그래밍- 사용자정의함수 9문제

정보처리기사 실기



1. 다음 C언어로 구현된 프로그램을 분석하여 배열 <mines>의 각 칸에 들어갈 값을 쓰시오.



```
#include <stdio.h>
main( ) {
    int field[4][4] = { {0,1,0,1}, {0,0,0,1}, {1,1,1,0}, {0,1,1,1} };
    int mines[4][4] = { {0,0,0,0}, {0,0,0,0}, {0,0,0,0}, {0,0,0,0} };
    int w = 4, h = 4;
    for (int y = 0; y < h; y++) {
        for (int x = 0; x < w; x++) {
            if (field[y][x] == 0) continue;
            for (int j = y - 1; j <= y + 1; j++) {
                for (int i = x - 1; i <= x + 1; i++) {
                    if (chkover(w, h, j, i) == 1)
                        mines[j][i] += 1;
                }
            }
        }
    }
}

int chkover(int w, int h, int j, int i) {
    if (i >= 0 && i < w && j >= 0 && j < h) return 1;
    return 0;
}
```

배열 <field>

0	1	0	1
0	0	0	1
1	1	1	0
0	1	1	1

배열 <mines>

1	1	3	2
3	4	5	3
3	5	6	4
3	5	5	3

3. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
#include <stdio.h>
int isPrime(int number) {
    for (int i = 2; i < number; i++)
        if (number % i == 0) return 0;
    return 1;
}

int main() {
    int number = 13195;
    int max_div = 0;
    for (int i = 2; i < number; i++)
        if (isPrime(i) == 1 && number % i == 0) max_div = i;
    printf("%d", max_div);
}
```

답 : 29

[해설]

```
#include <stdio.h>
⑤ int isPrime(int number) {
⑥     for (int i = 2; i < number; i++)
⑦         if (number % i == 0) return 0;
⑧     return 1;
    }

    int main() {
①         int number = 13195;
②         int max_div = 0;
③         for (int i = 2; i < number; i++)
④⑨             if (isPrime(i) == 1 && number % i == 0) max_div = i;
⑩         printf("%d", max_div);
    }
```

모든 C언어 프로그램은 반드시 main() 함수에서 시작한다.

- ① 정수형 변수 number를 선언하고 13195로 초기화한다.
- ② 정수형 변수 max_div를 선언하고 0으로 초기화한다.
- ③ 반복 변수 i가 2부터 1씩 증가하면서 number보다 작은 동안 ④번을 반복 수행한다.

첫 번째 반복

- ④ i의 값 2를 인수로 isPrime을 호출한 결과가 1이고 number를 i로 나눈 나머지가 0이면 max_div에 i의 값을 저장한다.
- ⑤ 정수를 반환하는 isPrime() 함수의 시작점이다. ④번에서 전달받은 2를 number가 받는다.

- ⑥ 반복 변수 i 가 2부터 1씩 증가하면서 2보다 작은 동안 ⑦번을 반복 수행한다. i 의 값 2는 2보다 작지 않으므로 ⑦번을 수행하지 않고 ⑧번으로 이동한다.
- ⑧ 1을 반환하면서 함수를 호출했던 ⑨번으로 이동한다.
- ⑨ ⑧번에서 돌려받은 값은 1이지만, $number$ 의 값 13195를 i 의 값 2로 나눈 나머지는 1이므로 $max_div = i$ 를 수행하지 않고 ③번으로 돌아가 i 의 값을 1 증가시킨다.

두 번째 반복

- ④ i 의 값 3을 인수로 $isPrime$ 을 호출한 결과가 1이고 $number$ 를 i 로 나눈 나머지가 0이면 max_div 에 i 의 값을 저장한다.
- ⑤ ④번에서 전달받은 3을 $number$ 가 받는다.
- ⑥ 반복 변수 i 가 2부터 1씩 증가하면서 3보다 작은 동안 ⑦번을 반복 수행한다.
- ⑦ 3을 i 로 나눈 나머지가 0이면 0을 반환하면서 함수를 호출했던 ⑨번으로 이동한다.
- ⑥~⑦번 반복문 실행에 따른 변수들의 변화는 다음과 같다.

number	i
3	2
	3

- ⑧ 1을 반환하고 함수를 호출했던 ⑨번으로 이동한다.
- ⑨ ⑧번에서 돌려받은 값은 1이지만, $number$ 의 값 13195를 i 의 값 3으로 나눈 나머지는 1이므로 $max_div = i$ 를 수행하지 않고 ③번으로 돌아가 i 의 값을 1 증가시킨다.

세 번째 반복

- ④ i 의 값 4를 인수로 $isPrime$ 을 호출한 결과가 1이고 $number$ 를 i 로 나눈 나머지가 0이면 max_div 에 i 의 값을 저장한다.
- ⑤ ④번에서 전달받은 4를 $number$ 가 받는다.
- ⑥ 반복 변수 i 가 2부터 1씩 증가하면서 4보다 작은 동안 ⑦번을 반복 수행한다.
- ⑦ 4를 i 로 나눈 나머지가 0이면 0을 반환하면서 함수를 호출했던 ⑨번으로 이동한다.
- ⑥~⑦번 반복문 실행에 따른 변수들의 변화는 다음과 같다.

number	i
4	2

- ⑨ ⑧번에서 돌려받은 값이 0이고, $number$ 의 값 13195를 i 의 값 4로 나눈 나머지는 3이므로 $max_div = i$ 를 수행하지 않고 ③번으로 돌아가 i 의 값을 1 증가시킨다.

네 번째 반복

- ④ i 의 값 5를 인수로 $isPrime$ 을 호출한 결과가 1이고 $number$ 를 i 로 나눈 나머지가 0이면 max_div 에 i 의 값을 저장한다.
- ⑤ ④번에서 전달받은 5를 $number$ 가 받는다.
- ⑥ 반복 변수 i 가 2부터 1씩 증가하면서 5보다 작은 동안 ⑦번을 반복 수행한다.
- ⑦ 5를 i 로 나눈 나머지가 0이면 0을 반환하고 함수를 호출했던 ⑨번으로 이동한다.
- ⑥~⑦번 반복문 실행에 따른 변수들의 변화는 다음과 같다.

number	i
5	2
	3
	4
	5

⑧ 1을 반환하고 함수를 호출했던 ⑨번으로 이동한다.

⑨ ⑧번에서 돌려받은 값이 1이고, number의 값 13195를 i의 값 5로 나눈 나머지도 0이므로 max_div에 5를 저장한 후 ③번으로 돌아가 i의 값을 1 증가시킨다.

위의 과정을 통해 다음 사항들을 알 수 있다.

- isPrime() 함수는 인수를 2에서 시작하여 전달받은 수보다 1 작을 때까지 나눴을 때 끝까지 나머지가 0이 아니면 1을 반환하는 것으로 보아 소수를 찾는 함수임을 알 수 있다.
- ⑨번에서 isPrime(i)가 1이라는 것은 i가 소수임을 의미하고, number를 i로 나눈 나머지가 0이라는 것은 i가 number의 약수라는 의미이므로, max_div에는 소수이자 number의 약수인 수가 저장된다.
- i의 값이 1씩 증가하면서 number보다 1 작을 때까지 위 과정을 수행하므로 number의 소수로 된 약수 중 가장 큰 소수에 해당하는 값이 max_div에 저장된다.
- 13195의 소수로 된 약수는 5, 7, 13, 29이며, 이 중 가장 큰 소수인 29가 최종적으로 max_div에 저장된다.
- 자세한 값의 변화는 다음 표를 통해 확인하자.

main() 함수			isPrime() 함수		
number	i	max_div	number	i	반환값
13195	2	0	2	2	1
⋮	3		3	2	1
⋮				3	
	4		4	2	0
	5	5	5	2	1
				3	
				4	
				5	
	⋮	⋮	⋮	⋮	⋮
	29	29	29	2	1
				3	
				4	
				⋮	
				28	
				29	
	⋮	⋮	⋮	⋮	⋮
13194			13194	2	0
13195					

⑩ max_div의 값 29를 정수로 출력한다.

결과 29



1. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
#include <stdio.h>
main() {
    int s, el = 0;
    for (int i = 6; i <= 30; i++) {
        s = 0;
        for (int j = 1; j <= i / 2; j++)
            if (i % j == 0)
                s = s + j;
        if (s == i)
            el++;
    }
    printf("%d", el);
}
```

답 : 2

[해설]

어떤 정수의 약수 중 자신을 제외한 약수를 모두 합하면 자신과 같아지는 수가 있다. 예를 들어 6의 약수 1, 2, 3, 6 중 6을 제외한 1, 2, 3을 더하면 6이 되어 자신과 같아진다. 다음은 6부터 30까지의 정수 중 이러한 약수를 갖는 수를 찾아 출력하는 알고리즘이다.

```
#include <stdio.h>
main( ) {
    ① int s, el = 0;
    ② for (int i = 6; i <= 30; i++) {
    ③     s = 0;
    ④     for (int j = 1; j <= i / 2; j++)
    ⑤         if (i % j == 0)
    ⑥             s = s + j;
    ⑦     if (s == i)
    ⑧         el++;
    }
    ⑨ printf("%d", el);
}
```

- ① 정수형 변수 s, el을 선언하고, el을 0으로 초기화한다.
- ② 반복 변수 i가 6부터 1씩 증가하면서 30보다 작거나 같은 동안 ③~⑧번을 반복 수행한다.
- ③ s에 0을 저장한다.

- ④ 반복 변수 j 가 1부터 1씩 증가하면서 $i/2$ 보다 작거나 같은 동안 ⑤, ⑥번을 반복 수행한다.
- ⑤ i 를 j 로 나눈 나머지가 0이면 ⑥번으로 이동하고, 아니면 현재 반복문의 처음인 ④번으로 이동한다.
- ⑥ s 에 j 의 값을 누적시킨다. 구해진 약수를 더하는 과정이다.
- ⑦ s 와 i 의 값이 같으면 약수를 모두 더한 값과 자신이 같은 수를 찾은 것이므로, ⑧번으로 이동하고, 아니면 현재 반복문의 처음인 ②번으로 이동한다.
- ⑧ ' $el = el + 1$;'과 동일하다. 약수를 모두 더한 값과 자신이 같은 수의 개수를 누적시키는 과정이다. 반복문 실행에 따른 변수들의 변화는 다음과 같다.

i	j	s	el
6		0	0
	1	1	
	2	3	
	3	6	
	4		1
7		0	
	1	1	
	2		
	3		
	4		
⋮	⋮	⋮	⋮
28		0	
	1	1	
	2	3	
	3		
	4	7	
	5		
	6		
	7	14	
	⋮	⋮	⋮
	14	28	
	15		2
⋮	⋮	⋮	⋮
31			

- ⑨ el 의 값 2를 정수로 출력한다.

결과 2

5. 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
public class Test {
    public static void main(String[] args) {
        int w = 3, x = 4, y = 3, z = 5;
        if((w == 2 | w == y) & !(y > z) & (1 == x ^ y != z)) {
            w = x + y;
            if(7 == x ^ y != w)
                System.out.println(w);
            else
                System.out.println(x);
        }
        else {
            w = y + z;
            if(7 == y ^ z != w)
                System.out.println(w);
            else
                System.out.println(z);
        }
    }
}
```

답 : 7

[해설]

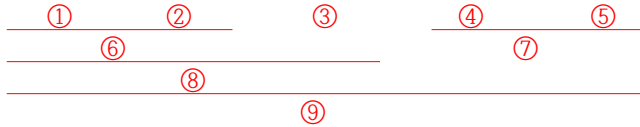
```
public class Test {
    public static void main(String[] args) {
        ① int w = 3, x = 4, y = 3, z = 5;
        ② if((w == 2 | w == y) & !(y > z) & (1 == x ^ y != z)) {
        ③     w = x + y;
        ④     if(7 == x ^ y != w)
        ⑤         System.out.println(w);
            else
                System.out.println(x);
        } ⑥
        else {
            w = y + z;
            if(7 == y ^ z != w)
                System.out.println(w);
            else
                System.out.println(z);
        } ⑦
    }
}
```

① 정수형 변수 w, x, y, z를 선언하고 각각 3, 4, 3, 5로 초기화한다.

② 조건이 참이면 ③번부터 ⑥번 이전까지의 문장을, 거짓이면 ⑥번 아래 else의 다음 문장부터 ⑦번

이전까지의 문장을 수행한다. 연산자 우선순위에 따라 다음의 순서로 조건의 참/거짓을 확인한다.

• $(w == 2 \mid w == y) \& !(y > z) \& (1 == x \wedge y != z)$



- ① : w의 값 3과 2는 같지 않으므로 거짓(0)이다.
- ② : w의 값 3과 y의 값 3은 같으므로 참(1)이다.
- ③ : y의 값 3은 z의 값 5보다 크지 않으므로 거짓(0)이지만, 앞에 !(논리 not)가 있으므로 참(1)이다.
- ④ : 1과 x의 값 4는 같지 않으므로 거짓(0)이다.
- ⑤ : y의 값 3과 z의 값 5는 같지 않으므로 참(1)이다.

+ end = 이진계산기

- ⑥ ① | ② : ①의 결과 0과 ②의 결과 1을 |(비트 or) 연산하면
$$\begin{array}{r} 0000(0) \\ | 0001(1) \\ \hline 0001(1) \end{array}$$
이므로 결과는 1이다.

+ 비트연산 음수의 논리값을 구하는 방법

- ⑦ ④ ^ ⑤ : ④의 결과 0과 ⑤의 결과 1을 ^(비트 xor) 연산하면
$$\begin{array}{r} 0000(0) \\ ^ 0001(1) \\ \hline 0001(1) \end{array}$$
이므로 결과는 1이다.

X
동작

Xor / XAND 코드인

- ⑧ ⑥ & ③ : ⑥의 결과 1과 ③의 결과 1을 &(비트 and) 연산하면
$$\begin{array}{r} 0001(1) \\ \& 0001(1) \\ \hline 0001(1) \end{array}$$
이므로 결과는 1이다.

- ⑨ ⑧ & ⑦ : ⑧의 결과 1과 ⑦의 결과 1을 &(비트 and) 연산하면 결과는 1이다.

∴ 최종 결과는 1이며, 1은 조건에서 참을 의미하므로 ③번으로 이동한다.

③ w에 x와 y의 합을 저장한다. (w=7)

④ 조건이 참이면 ⑤번 문장을, 거짓이면 ⑤번 아래 else 다음 문장을 수행한다. 연산자 우선순위에 따라 다음의 순서로 조건의 참/거짓을 확인한다.

• $7 == x \wedge y != w$

- ① : 7과 x의 값 4는 같지 않으므로 결과는 거짓(0)이다.
- ② : y의 값 3과 w의 값 7은 같지 않으므로 결과는 참(1)이다.
- ③ ① ^ ② : ①의 결과 0과 ②의 결과 1을 ^(비트 xor) 연산하면 결과는 1이다.

∴ 최종 결과는 1이며, 1은 조건에서 참을 의미하므로 ⑤번 문장을 수행한다.

⑤ w의 값 7을 출력하고 커서를 다음 줄의 처음으로 옮긴다. 모든 if문이 종료되었으므로 ⑦번으로 이동하여 프로그램을 종료한다.

결과 7

9. 다음은 변수 n에 저장된 10진수를 2진수로 변환하여 출력하는 Java 프로그램이다. 프로그램을 분석하여 괄호(①, ②)에 들어갈 알맞은 답을 쓰시오.



```
public class Test {
    public static void main(String[] args) {
        int a[] = new int[8];
        int i = 0;
        int n = 10;
        while( ( ① ) ) {
            a[i++] = ( ② );
            n /= 2;
        }
        for(i = 7; i >= 0; i--)
            System.out.print(a[i]);
    }
}
```

답

- ① $n > 0$
- ② $n \% 2$

[해설]

```
public class Test {
    public static void main(String[] args) {
        ①      int a[] = new int[8];
        ②      int i = 0;
        ③      int n = 10;
        ④      while(n > 0) {
        ⑤          a[i++] = n % 2;
        ⑥          n /= 2;
        }
        ⑦      for(i = 7; i >= 0; i--)
        ⑧          System.out.print(a[i]);
    }
}
```

① 8개의 요소를 갖는 정수형 배열 a를 선언한다.

배열 a

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

※ Java는 배열 선언 시 초기화를 하지 않아도 자동으로 0으로 초기화된다.

② 정수형 변수 i를 선언하고 0으로 초기화한다.

③ 정수형 변수 n을 선언하고 10으로 초기화한다.

④ n이 0보다 큰 동안 ⑤~⑥번을 반복 수행한다.

⑤ i++은 후치 증가 연산자이므로, a[i]에 n을 2로 나눈 나머지를 저장한 후, i의 값에 1을 더한다.

⑥ 'n = n / 2;'와 동일하다. n을 2로 나눈 값을 n에 저장한다.

반복문 실행에 따른 변수들의 값의 변화는 다음과 같다.

12. 다음 Java로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)



```
public class Test{
    public static void main(String[] args){
        int a = 0, sum = 0;
        while (a < 10) {
            a++;
            if (a%2 == 1)
                continue;
            sum +=a;
        }
        System.out.println(sum);
    }
}
```

답 : 30

[해설]

```
public class Test{
    public static void main(String[] args){
        ① int a = 0, sum = 0;
        ② while (a < 10) {
            ③ a++;
            ④ if (a%2 == 1)
            ⑤ continue;
            ⑥ sum +=a;
        }
        ⑦ System.out.println(sum);
    }
}
```

- ① 정수형 변수 a와 sum을 선언하고 각각 0으로 초기화한다.
 - ② a가 10보다 작은 동안 ③~⑥번을 반복 수행한다.
 - ③ 'a = a + 1;'과 동일하다. a의 값에 1을 누적시킨다.
 - ④ a%2, 즉 a를 2로 나눈 나머지가 1이면 ⑤번을 수행하고, 아니면 ⑥번으로 이동한다.
 - ⑤ while문의 시작점인 ②번으로 제어를 이동시킨다.
 - ⑥ 'sum = sum + a;'와 동일하다. sum에 a의 값을 누적시킨다.
- 반복문 실행에 따른 변수들의 변화는 다음과 같다.

a	sum
0	0
1	
2	2
3	

3. 다음은 <제품>(제품명, 단가, 제조사) 테이블을 대상으로 “H” 제조사에서 생산한 제품들의 ‘단가’보다 높은 ‘단가’를 가진 제품의 정보를 조회하는 <SQL문>이다. 괄호에 알맞은 답을 적어 <SQL문>을 완성하시오.



<SQL문>

```
SELECT 제품명, 단가, 제조사
FROM 제품
WHERE 단가 > (      ) (SELECT 단가 FROM 제품 WHERE 제조사 = 'H');
```

답 : ALL

[해설]

- ② SELECT 제품명, 단가, 제조사
FROM 제품
WHERE 단가 > ALL (

① SELECT 단가
FROM 제품
WHERE 제조사 = 'H');
- ‘제품명’, ‘단가’, ‘제조사’ 속성을 표시한다.
<제품> 테이블에서 검색한다.
‘단가’가 하위 질의로 검색된 모든(ALL) 단가보다 큰 자료만을 대상으로 한다.
‘단가’를 표시한다.
<제품> 테이블에서 검색한다.
제조사가 “H”인 자료만을 대상으로 한다.

- 문제의 질의문은 하위 질의가 있는 질의문입니다.
- 먼저 WHERE 조건에 지정된 하위 질의의 SELECT문을 해석한 다음 그 결과를 본 질의의 조건에 있는 ‘단가’ 속성과 비교합니다.
- <제품> 테이블에 다음과 같은 자료가 들어있다고 가정하여 설명합니다.

<제품>

제품명	단가	제조사
냉장고	200	H
TV	150	H
세탁기	300	H
건조기	250	A
핸드폰	400	B
컴퓨터	500	C

- ① <제품> 테이블에서 ‘제조사’ 속성의 값이 “H”인 튜플의 ‘단가’ 속성의 값을 검색합니다.

단가
200
150
300

- ② <제품> 테이블에서 ‘단가’ 속성의 값이 ①번에서 검색된 모든 단가보다 큰 자료를 대상으로 ‘제품명’, ‘단가’, ‘제조사’를 표시합니다.

제품명	단가	제조사
핸드폰	400	B
컴퓨터	500	C



10. 다음 <학생> 테이블을 참고하여 <처리 조건>에서 요구하는 SQL문을 작성하시오.

<학생>

학번 (varchar)	이름 (varchar)	학년 (number)	수강과목 (varchar)	점수 (number)	연락처 (varchar)
20E0232	김인영	3	세무행정	4.5	010-5412-4544
19D0024	이성화	2	토목개론	3	010-1548-4796
20E0135	성유수	4	실용법학	3.5	010-9945-7411
20E0511	우인혁	1	데이터론	2	010-3451-4972

<처리 조건>

- 3, 4학년의 학번, 이름을 조회한다.
- IN 예약어를 사용해야 한다.
- 속성명 아래의 괄호는 속성의 자료형을 의미한다.

답 : SELECT 학번, 이름 FROM 학생 WHERE 학년 IN (3, 4);

[해설]

SELECT 학번, 이름
FROM 학생
WHERE 학년 IN (3, 4);

'학번', '이름'을 표시한다.
<학생> 테이블에서 검색한다.
'학년'의 값이 3 또는 4인 자료만을 대상으로 한다.

<결과>

학번	이름
20E0232	김인영
20E0135	성유수

11. 다음 <student> 테이블을 참고하여 'name' 속성으로 'idx_name'이라는 인덱스를 생성하는 SQL문을 작성하시오.



<student>

stid	name	score	deptid
2001	brown	85	PE01
2002	white	45	EF03
2003	black	67	UW11

답 : `CREATE INDEX idx_name ON student(name);`

[해설]

CREATE INDEX idx_name	'idx_name'이라는 이름의 인덱스를 생성한다.
ON student(name);	<student> 테이블의 'name' 속성을 사용한다.

이건 논리적으로 맞지 않아요

[그룹]

13. 다음은 <회원> 테이블에서 '이름'이 "이"로 시작하는 회원들을 '가입일' 순으로 내림차순 정렬하는 <SQL문>이다. 괄호(①, ②)에 들어갈 알맞은 답을 쓰시오.



<회원> 테이블

회원번호	이름	성별	가입일
1001	이진성	남	2021-06-23
1002	조이령	여	2021-06-24
1003	최민수	남	2021-06-28
1004	김차희	여	2021-07-03
1005	이미경	여	2021-07-10

<SQL문>

SELECT * FROM 회원 WHERE 이름 LIKE '()' ORDER BY 가입일 ();

답

- ① 이%
- ② DESC

[해설]

• SQL문

SELECT *
FROM 회원
WHERE 이름 LIKE '이%'
ORDER BY 가입일 DESC;

모든 속성을 표시한다.
<회원> 테이블에서 검색한다.
'이름'이 '이'로 시작하는 튜플만을 대상으로 한다.
'가입일'을 기준으로 내림차순 정렬한다.

• SQL 실행 결과

회원번호	이름	성별	가입일
1005	이미경	여	2021-07-10
1001	이진성	남	2021-06-23

가림 검색
정렬순서 내림차순
그리고
이름
+
Join
가림
이름
내림차순