

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 1. 2
4
8

* C언어 데이터 타입 크기

- short: 2 Byte
- float: 4 Byte
- double: 8 Byte

C언어 2. boolean

논리형 데이터 타입 boolean에 대한 설명이다.

C언어 3. 6
6
6
6

* 증가/감소 연산자

printf("%d", ++number);

// ++number : number를 1 증가 시킨 후 사용(5→6)

printf("%d", number++);

// number++ : number 사용 후 1 증가(6→7)

printf("%d", --number);

// --number : number를 1 감소 시킨 후 사용(7→6)

printf("%d", number--);

// number-- : number 사용 후 1 감소(6→5)

C언어 4. 1, 0

* 논리 연산자

: 2개의 논리 값을 연산하여 True(1), False(0) 결과 반환

- result1 = A != 0 || B < 1;

// ||(OR) : A != 0 또는 B < 1 중 하나라도 참이면 참 => 1

- result2 = A > 5 && B <= 8;

// &&(AND) : A > 5 와 B <= 8 모두 참이면 참 => 0

C언어 5. 200

* 조건 연산자

: 조건 ? 수식1 : 수식 2; 조건이 참이면 수식1 실행, 거짓이면 수식2 실행

- result = a < b ? b : c;

// 조건: a < b = 100 < 200 => 참이므로 b 실행

C언어 6. 30, 10, 5, 5

* 할당(대입) 연산자

- D /= A; // D = D / A = 15 / 3 = 5

- C -= D; // C = C - D = 10 - 5 = 5

- B += C; // B = B + C = 5 + 5 = 10

- A *= B; // A = A * B = 3 * 10 = 30

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 7. ① %d ② %o ③ %x

- * 서식 문자
- %d: 10진수 정수 출력
- %o: 8진수 정수 출력
- %x: 16진수 정수 출력

C언어 8. 1, 7, 6

- * 비트 연산자: 비트별(0,1)로 연산
- &(and): 두 비트가 모두 1일때만 결과가 1, 그렇지 않으면 결과는 모두 0이다.
- |(or): 두 비트 중 어느 하나라도 1이면 결과는 1, 그렇지 않으면 결과는 0이다.
- ^(xor): 두 비트가 같으면 0, 다르면 1이다.
- a = 3 = 011(2) - b = 5 = 101(2)
- a & b = 011(2) - a | b = 011(2) - a ^ b = 011(2)
- &101(2) | 101(2) ^ 101(2)
- = 001(2) = 111(2) = 110(2)
- "%d": 10진수 정수로 출력

C언어 9. %s

- * 서식 문자
- %s: 문자열 출력

C언어 10. \n

- * 제어 문자
- \n: 커서를 다음 줄로 바꾼다. (new line)

C언어 11. ① while ② i++

```
#include <stdio.h>
int main() {
    int i = 1, a = 3, result; // 변수 선언 및 초기화
    while (i <= 9) // i <= 9 일경우 while문을 수행
    {
        result = a * i; // a*i 의 값을 result에 저장
        printf("3 * %d = %d \n", i++, result);
        // 3 * i = result 출력한 후 i를 1씩 증가
    }
}
```

횟수	a	i	result = a*i
1	3	1	3
2	3	2	6
3	3	3	9
4	3	4	12
5	3	5	15
6	3	6	18
7	3	7	21
8	3	8	24
9	3	9	27

=> i == 10일 경우
while문을 빠져나가
면서 프로그램 종료

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 12. ① do ② while(i < 5) 또는 whlie(i <= 4)

```
int main() {
    int i = 1, multi; // 변수 선언 및 초기화
    do // do~while문 수행
    {
        multi = i * (i + 1); // i×(i+1)의 값을 multi에
                             저장
        printf("%d * %d = %d Wn", i, i + 1, multi);
        // i * (i+1) = multi 출력
        i++; // i값 1증가
    } while (i < 5); // i<5 일 경우 do~while문 수행
}
```

횟수	i	i+1	multi
1	1	2	2
2	2	3	6
3	3	4	12
4	4	5	20
5	5		

=> i == 5일 경우
do~while문을 빠져
나가면서 프로그램
종료

C언어 13. ① sum >= 1000 ② break;

```
int main() {
    int sum = 0; int i = 1; // 변수 선언 및 초기화
    while(1) { // 무한 루프
        sum = sum + i; //sum+i의 값을 sum에 저장
        if(sum >= 1000) { // sum>=1000일 경우
            break; // 무한 루프문 종료
        }
        i++; //i값 1증가
    }
    printf("1~100의 합 중 최초로 1000이 넘는 위치  
는? %d", i); // i값 출력
}
```

횟수	i	sum
1	1	1
2	2	3
3	3	6
...
44	44	990
45	45	1035

=> sum >= 1000일 경우
while문을 빠져나가면서
프로그램 종료

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 14. ① for ② printf("*");

```
int main() {
    int number, i; //변수 선언
    printf("숫자를 입력하세요: "); //화면에 "숫자를
    입력하세요:" 출력
    scanf("%d", &number); //키보드로 숫자를 입
    력 받음
    for (i = 0; i < number; i++) {
        // i=0 부터 i < number까지 for문 수행
        // i는 1씩 증가하면서 i<number까지 for문 수행
        printf("*"); //화면에 "*" 출력
    }
}
```

입력	nubmer	5
----	--------	---

횟수	i	printf("*")
1	0	*
2	1	**
3	2	***
4	3	****
5	4	*****
6	5	

=> i == 5일 경우
do~while문을 빠져나
가면서 프로그램 종료

C언어 15. scanf

```
int main() {
    int a_score, b_score, average; //변수 선언
    printf("점수를 입력하세요: "); //화면에 "점수를
    입력하세요:" 출력
    scanf("%d %d", &a_score, &b_score); //키보
    드로 정수 2개를 입력 받음
    average = (a_score + b_score) / 2; // 입력받
    은 두 정수의 평균을 average에 저장
    if(average >= 80) { //average>=80일 경우
        printf("A"); //화면에 "A" 출력
    }
    else if(average >= 60) { //average>=60일 경
    우
        printf("B"); //화면에 "B"출력
    }
    else { //그 외 경우
        printf("C"); //화면에 "C"출력
    }
}
```

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 16. ① C ② B ③ A

* 입력값: 40 50

1. scanf("%d %d", &a_score, &b_score) 문법에 따라 a_score = 40, b_score = 50 이다.

2. average = (a_score + b_score) / 2 의 값을 가지므로, average = (40 + 50) / 2 = 45 이다.

3. if~else if~else 조건문에 따라 average == 45 인 경우 '그 외 경우'에 속하므로 "C" 가 출력된다.

- average >= 80 : A -average >= 60 : B

- 그 외 경우: C

C언어 17. switch

* switch ~ case문

switch(month)

case 1: // month == 1 인 경우

case 3: // month == 3 인 경우

처리문; // 처리문 수행

break; // switch문 종료

...

default: // 만족하는 조건이 없을 경우

처리문; // 처리문 수행

break; // switch문 종료

C언어 18. ① i == 5 ② continue;

```
int main() {
    int i, result; // 변수 선언
    for(i = 1; i <= 9; i++) {
        // i=1부터 1씩 증가하면서 i<=9까지 for문 수행
        if(i == 5) { // i==5인 경우
            continue; // 이후 문장을 실행하지 않고
                       // 다시 for문의 선두로 되돌아가서 실행
        }
        result = 3 * i; // 3×i 값을 result에 저장
        printf("3 * %d = %d Wn", i, result);
        // 3 * i = result 출력
    }
}
```

횟수	i	result = 3*i
1	1	3
...
4	4	12
6	6	18
...
9	9	27

=> i == 5일 경우
result = 3 * i와 printf()
문을 수행하지 않고 for
문의 선두로 돌아감

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 19. $n \% i$

```
for (i = 1; i <= n; ++i) {  
    if (n%i == 0)  
        printf("%d ", i); }  
// 1부터 n까지 증가시키면서 n을 나누어서 나머지가 0이 나올 경우 약수로 출력
```

C언어 20. $sum + n$

```
while(1) {  
    n++; // n을 1씩 증가  
    sum = sum + n; // sum에 n의 합계를 누적  
    if(sum>50)  
        break;  
} // sum이 50을 넘길 경우 break문이 실행되어  
무한 루프 종료
```

C언어 21. $A + B$

```
for (A = 1; A <= 6; A++)  
    // A를 1씩 증가시키면서 6번 반복 수행  
    for (B = 1; B <= 6; B++)  
        // B를 1씩 증가시키면서 6번 반복 수행  
        if (A + B == sum)  
            // A+B 가 sum 과 같을 경우  
            printf("%d \t %d \n ", A, B);  
            // A 와 B 의 값 출력
```

C언어 22. 0번째 학생의 점수는 90입니다.
1번째 학생의 점수는 80입니다.
2번째 학생의 점수는 85입니다.

```
#define st 3 // st를 3의 상수로 정의
```

```
score[0] = 90;  
score[1] = 80;  
score[2] = 85;  
for (i = 0; i < st; i++)  
    printf("%d번째 학생의 점수는 %d입니다. \n", i, score[i]); // i를 1씩 증가시키며 각 배열의 번호에 해당하는 점수 출력
```

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 23. num[i] < min

```
for (i = 1; i < n; i++) { // i를 1씩 증가
    if (num[i] < min)
        // num[i]가 min 보다 작을 경우
        min = num[i]; // min 을 num[i]로 변경
}
printf("최소값은 %d입니다. \n", min);
// i[1]~i[4]까지 비교한 결과인 min(최소값) 출력
```

C언어 24. ① dan = 2 ② dan*i

```
for (dan = 2; dan < 10; dan++) {
    // 2단부터 9단까지 반복 수행
    for (i = 1; i < 10; i++)
        // dan*1부터 dan*9까지 반복 수행
        printf("%d*%d=%d ", dan, i, dan*i);
        // dan*1부터 dan*9까지의 결과 출력
    printf("\n");
    // 각 단이 끝나면 다음 줄로 넘어감
}
```

C언어 25. n % 2 != 0

```
while (n <= 100) {
    // n이 100 이하일 경우
    if (n % 2 != 0)
        // n을 2로 나누었을 때 0이 아닌 경우(홀수)
        sum += n; // sum에 n을 누적 합계
    n++; // n은 1씩 증가
}
```

C언어 26. ① SIZE ② n[i]

```
#define SIZE 5 // SIZE 를 5의 상수로 정의

for (i = 0; i < SIZE; ++i) {
    // i는 0부터 SIZE-1 까지 반복 수행
    printf("%d  %d  ", i, n[i]);
    // i번째와 n[i]번째의 값을 출력
    for (j = 1; j <= n[i]; j++)
        // j는 1부터 n[i]까지 반복 수행
        printf("*");
        // n[i]번째의 값만큼 * 출력
    printf("\n");
    // n[i]번째의 수만큼 * 출력했을 경우 줄바꿈
}
```

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 27. ① i-- ② %c

```
for(i = 0; i < A[i]; i++) {  
    printf("%c", A[i]);  
} // 배열 A의 문자열을 차례로 출력  
printf("\n");  
for(i = 6; i >= 0; i--) {  
    // i는 6에서부터 0까지 1씩 감소하며 반복 수행  
    printf("%c", A[i]); // A[i]번째 문자 출력  
}
```

C언어 28. n[j] = n[i]

```
for(i = 1; i < SIZE; i++) {  
    // i는 1부터 SIZE-1 까지 반복 수행  
    for(j = 0; j < SIZE-1; j++) {  
        // i는 0부터 9까지 반복 수행  
        if(n[j] > n[i]) { // n[j]가 n[i]보다 클 경우  
            min = n[j]; // min에 n[j] 값 저장  
            n[j] = n[i]; // n[j]에 n[i] 값 저장  
            n[i] = min; // n[i]에 min 값 저장  
        }  
    }  
}  
for(i = 0; i < SIZE; i++) {  
    // i는 1부터 SIZE(10)까지 반복 수행  
    printf("%d ", n[i]); // n[i]번째 숫자 출력  
}
```


[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 29. $j \leq n-i$

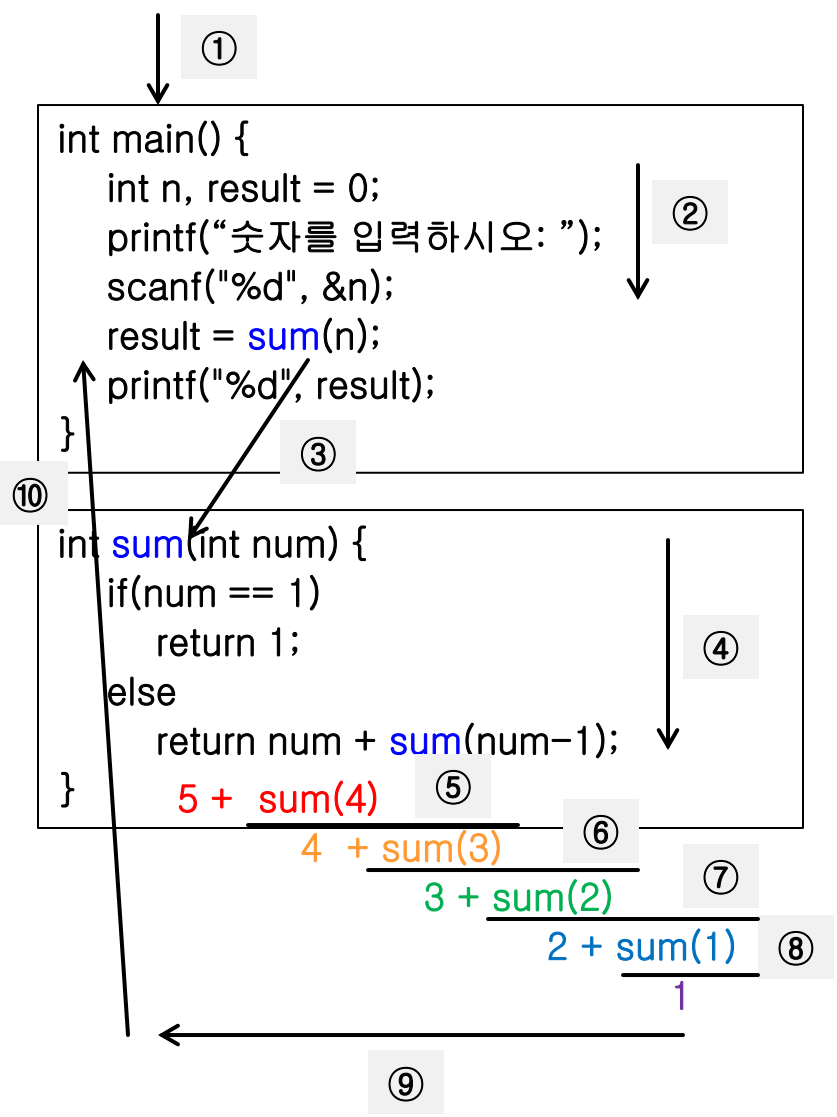
```
for(i = 0; i < n; i++) {  
    // i는 0부터 n-1까지 반복 수행  
    for(j = 1; j <= n-i; j++)  
        // j는 1부터 n-i 까지 반복 수행  
        printf("%3d", j);  
        // j 출력(3은 출력할 자리 수를 의미)  
    printf("\n");  
}
```

C언어 30. ① break; ② count++;

```
for(num = 2; num <= 100; num++) {  
    // num은 2부터 100까지 반복 수행  
    for(i = 2; i < num; i++) {  
        // i는 2부터 num-1 까지 반복 수행  
        if(num % i == 0)  
            // num 에서 i 를 나누었을 때 나머지가  
            // 0 인 경우  
            break; // for문 종료  
    }  
    if(num == i) // num 과 i 가 같을 경우  
        count++; // 소수이므로 count 증가  
}
```

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 31. num + sum(num - 1)



- ① main() 함수 실행
- ② 키보드로 5를 입력 => n = 5
- ③ sum(5) 함수 호출 => 매개변수 : 5
- ④ 매개변수 : 5 => num = 5 => else문 실행
- ⑤ sum(4) 함수 호출 => 매개변수 : 4 => num = 4 => else문 실행
- ⑥ sum(3) 함수 호출 => 매개변수 : 3 => num = 3 => else문 실행
- ⑦ sum(2) 함수 호출 => 매개변수 : 2 => num = 2 => else문 실행
- ⑧ sum(1) 함수 호출 => 매개변수 : 1 => num = 1 => if문 실행
- ⑨ return (1 + 2 + 3 + 4 + 5) 수행

sum(1) return 1	1
2 + sum(1)	3
3 + sum(2)	6
4 + sum(3)	10
5 + sum(4)	15

⑩ result = 15 => 화면에 15 출력

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 32. 1
2
6
24
120

<입력>

5

```

①
int main() {
    int n, result;
    printf("정수를 입력하세요 : ");
    scanf("%d", &n);
    result = func(n);
}
    
```

```

③
int func(int n) {
    int value;
    if(n <= 1) {
        printf("1 \n ");
        return 1;
    }
    else {
        value = (n * func(n-1));
        printf("%d \n ", value);
        return value;
    }
}
    
```

- ① main() 함수 실행
- ② 키보드로 5를 입력 => n = 5
- ③ func(5) 함수 호출 => 매개변수 : 5
- ④ 매개변수 : 5 => n = 5 => else문 실행
- ⑤ func(4) 함수 호출 => 매개변수 : 4 => n = 4 => else문 실행
- ⑥ func(3) 함수 호출 => 매개변수 : 3 => n = 3 => else문 실행
- ⑦ func(2) 함수 호출 => 매개변수 : 2 => n = 2 => else문 실행
- ⑧ func(1) 함수 호출 => 매개변수 : 1 => n 1 => if문 실행
- ⑨ return (5 * 4 * 3 * 2 * 1) 수행

func(1) return 1	출력
2 * func(1)	1
3 * func(2)	2
4 * func(3)	6
5 * func(4)	24
	120

⑩ result = 120

$$\begin{array}{r}
 5 * \text{sum}(4) \quad \textcircled{5} \\
 \hline
 4 * \text{sum}(3) \quad \textcircled{6} \\
 \hline
 3 * \text{sum}(2) \quad \textcircled{7} \\
 \hline
 2 * \text{sum}(1) \quad \textcircled{8} \\
 \hline
 1
 \end{array}$$

[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 33. ① MAX_STACK_SIZE - 1 ② ++top ③ top-- ④ push(100) ⑤ push(500)

```
int main() {  
    push(100);  
    push(500);  
    push(200);  
    pop();  
    printStack();  
}
```

```
#include <stdio.h>  
#define MAX_STACK_SIZE 5  
int stack[MAX_STACK_SIZE];  
int top = -1;  
void push(int item) {  
    if(top == MAX_STACK_SIZE - 1){  
        printf("Stack is full!");  
        return;  
    }  
    stack[++top] = item;  
}  
int pop() {  
    if(top == -1) {  
        printf("Stack is empty!");  
        return 0;  
    }  
    return stack[top--];  
}  
int printStack() {  
    int i;  
    for(i = top; i > -1; i--) {  
        printf("%d \n", stack[i]);  
    }  
}
```

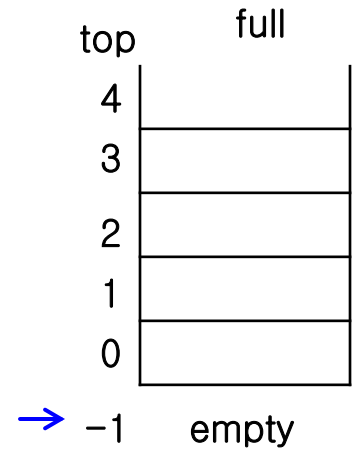
- ① main() 함수 실행
- ② push(100) 함수 호출 => 매개변수 : 100
- ③ 현재 top == -1 이므로 stack[++top] = item; 실행
: top을 1 증가시킨 후 현재 top에 '100' 삽입
- ④ 현재 top == 0 이므로 stack[++top] = item; 실행
: top을 1 증가시킨 후 현재 top에 '500' 삽입
현재 top == 1 이므로 stack[++top] = item; 실행
: top을 1 증가시킨 후 현재 top에 '200' 삽입
- ⑤ pop() 함수 호출
현재 top == 2 이므로 return stack[top--]; 실행
: 현재 top의 원소를 삭제한 후 top 1 감소
- ⑥ printStack() 함수 호출
i = top부터 1씩 감소하면서 i > -1일때 까지 for문
수행: 화면에 stack[i] 출력

[정답 및 해설] [프로그래밍 언어 활용>C언어]

```
int main() {
    push(100);
    push(500);
    push(200);
    pop();
    printStack();
}
```

시작 상태

현재 top 위치 →



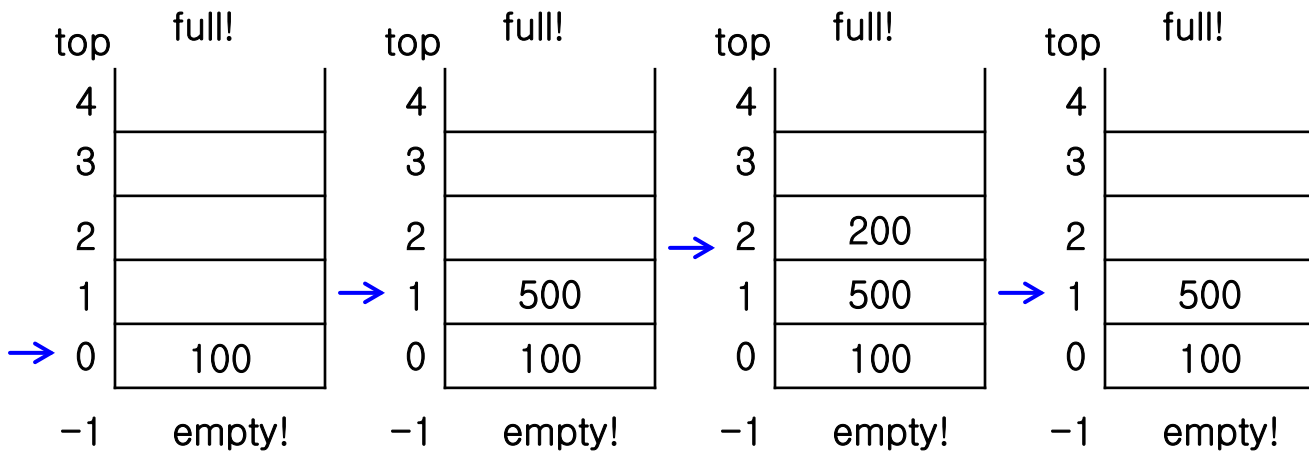
push(100)

push(500)

push(200)

pop()

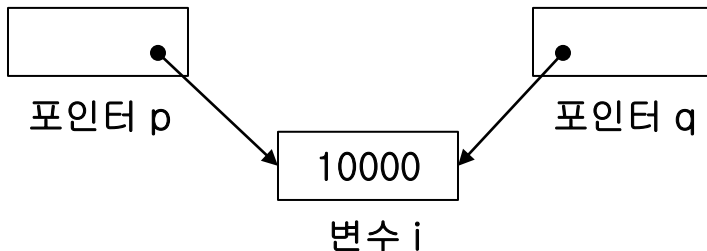
printStack()



[정답 및 해설] [프로그래밍 언어 활용>C언어]

C언어 34. 10002

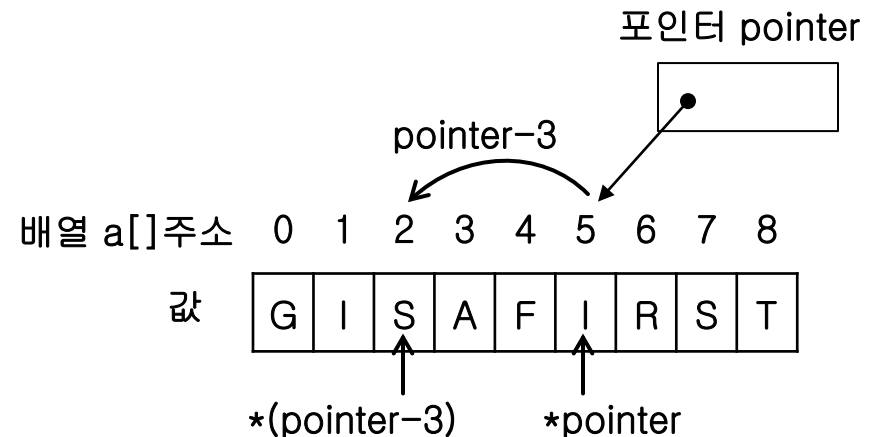
```
int main() {  
    int i = 10000; // 변수 i 초기화  
    int *p, *q; // 포인터 p, q 선언  
    p = &i; // 포인터 p와 변수 i를 연결  
    q = &i; // 포인터 q와 변수 i를 연결  
    *p = *p + 1; // 포인터 p를 통하여 1 증가  
    // 포인터 p와 변수 i가 연결되어 있으므로  
    // *p는 포인터에 연결된 값인 i이다.  
    // *p == i => i = i + 1 => 10001  
    *q = *q + 1; // 포인터 q를 통하여 1 증가  
    // 포인터 q와 변수 i가 연결되어 있으므로  
    // *q는 포인터에 연결된 값인 i이다.  
    // *q = i => i = i + 1 => 10002  
    printf("%d", i);  
}
```



*p: 포인터 p가 가르키고 있는 주소 영역에 접근하여 값을 가져오거나 변경한다.

C언어 35. I, S

```
int main() {  
    char a[] = { 'G', 'I', 'S', 'A', 'F', 'I', 'R', 'S',  
                'T' }; // 문자열 배열 초기화  
    char *pointer; // 포인터 pointer 선언  
    pointer = &a[5]; // 포인터 pointer와 a[5] 연결  
    printf("%c, %c", *pointer, *(pointer-3));  
    // 포인터 pointer와 a[5]가 연결되어 있으므로  
    // *(pointer-3)은 a[2]이다.  
}
```



[정답 및 해설] [프로그래밍 언어 활용>C언어]

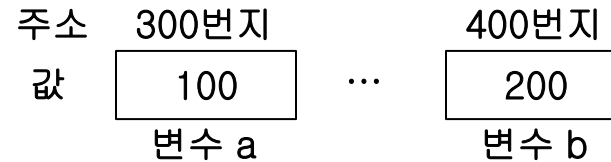
C언어 36. 200 100

```

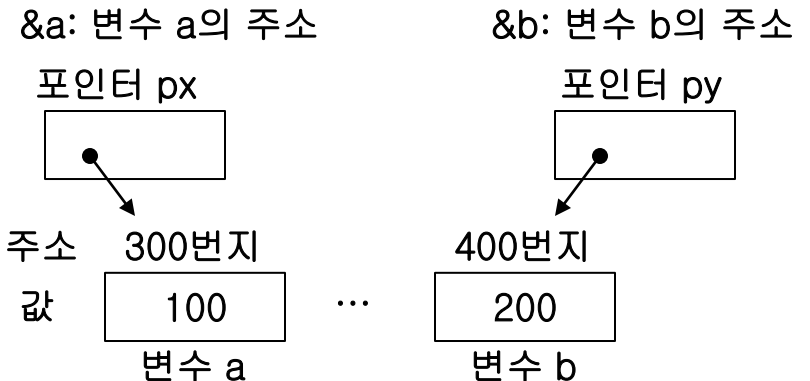
①
int main(void)
{
    int a = 100, b = 200;
    func(&a, &b);
    printf("%d %d", a, b); ④
}

void func(int *px, int *py)
{
    int tmp;
    tmp = *px;
    *px = *py;
    *py = tmp; ③
}
    
```

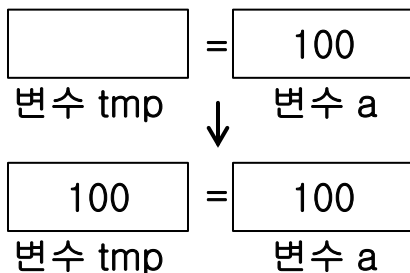
① main() 함수 실행, 변수 a, b 초기화



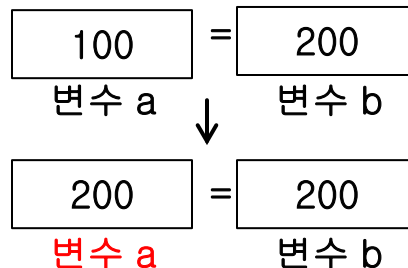
② func() 함수 호출 => 매개변수 &a, &b
=> int *px = &a, int *py = &b



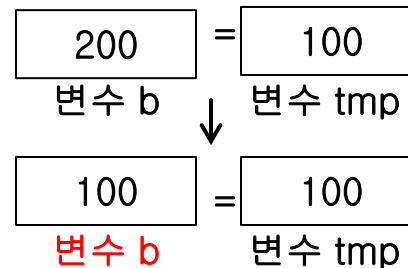
③ tmp = *px;
*px를 tmp에 대입



*px = *py
*py를 *px에 대입



*py = tmp
*tmp를 *py에 대입



④ a, b 값 출력

