

학번 : 이름 :

[다양한 포인터 실습(2)]

※ 포인터와 문자열 함수

1. [코드]와 [실행결과]를 보고, 문자열 st를 뒤쪽부터 역순으로 표시하는 함수를 완성하시오.

[코드]

```
#include <stdio.h>
void putReverseStr(const char *st);
int main(void){
    char str[100];
    printf("문자열 입력: ");
    scanf("%s", str);
    printf("역순 출력: ");
    putReverseStr(str);
    printf("\n");
    return 0;
}
void putReverseStr(const char *st){
    const char *p = st;
    while ( _____ (가) _____ )
        st++;
    if (st > p)
        while ( _____ (나) _____ )
            putchar( _____ (다) _____ );
}
```

[실행결과]

문자열 입력: sunrin4

역순 출력: nirnus

(가) *st != ' \0 ' ;

(나) st > p

(다) *--st

학번 : 이름 :

2. [코드]와 [실행결과]를 보고, 문자열 st에 포함되어 있는 숫자 문자 '0' ~ '9'의 개수를 cnt[0]~cnt[9]에 각각 저장하는 함수를 완성하시오.

[코드]

```
#include <stdio.h>
void cntDigit(const char *st, int cnt[]);
int main(void){
    char st[100];
    int cnt[10];
    printf("문자열 입력: ");
    fgets(st, sizeof(st), stdin);

    cntDigit(st, cnt);

    for(int i = 0; i < 10; i++)
        printf("cnt[%d] = %d\n", i, cnt[i]);
    return 0;
}
void cntDigit(const char *st, int cnt[]){
    for(int i=0; i<10; i++)
        *(cnt + i) = 0;

    while(*st){
        if(*st >= '0' && *st <= '9')
            cnt[_____ (나) _____]++;
            _____ (다) _____;
        }
    }
}
```

[실행결과]

문자열 입력: 260824sunrin

cnt[0] = 1

cnt[1] = 0

cnt[2] = 2

학번 : 이름 :

```
cnt[3] = 0
cnt[4] = 1
cnt[5] = 0
cnt[6] = 1
cnt[7] = 0
cnt[8] = 1
cnt[9] = 0
```

(가) i
(나) *st - '0'
(다) st++

3. [코드]와 [실행결과]를 보고, 숫자가 아닌 문자를 제거한 문자열 st를 표시하는 함수를 완성하시오.

```
[코드]
#include <stdio.h>
void putStrDgt(const char *st);
int main(void){
    char st[100];

    printf("문자열 입력: ");
    fgets(st, sizeof(st), stdin);

    printf("숫자만 출력: ");
    putStrDgt(st);
    printf("\n");

    return 0;
}
void putStrDgt(const char *st){
    while(_____ (가) _____){
        if(*st >= '0' && *st <= '9')
            putchar(_____ (나) _____);
```

학번 : 이름 :

(다) _____;
[실행결과] 문자열 입력: 26sun08rin24↵ 숫자만 출력: 260824
(가) *st != '\0' (나) *st (다) st++

4. [코드]와 [실행결과]를 보고, 문자열 s1과 s2를 연결하여 출력하는 함수를 완성하시오.

[코드] <pre>#include <stdio.h> void removeNewline(char *str); void concatStr(const char *s1, const char *s2); int main(void){ char s1[100], s2[100]; printf("첫 번째 문자열 입력: "); fgets(s1, sizeof(s1), stdin); removeNewline(s1); printf("두 번째 문자열 입력 : "); fgets(s2, sizeof(s2), stdin); removeNewline(s2); printf("연결된 문자열: "); concatStr(s1, s2); printf("\n"); return 0;</pre>
--

학번 : 이름 :

```
}  
void removeNewline(char *str){  
    while (*str) {  
        if (_____ (가) _____) {  
            _____ (나) _____;  
            break;  
        }  
        str++;  
    }  
}
```

[실행결과]

첫 번째 문자열 입력: hello,↵

두 번째 문자열 입력 : sunrin!↵

연결된 문자열: hello,sunrin!

(가) *str == ' \n '

(나) *str = ' \0 '

(다) *s1++

(라) *s2++

5. [코드]와 [실행결과]를 보고, 문자열 s1의 뒤에 문자열 s2를 붙여서 문자열 s1을 반환하는 함수를 완성하시오. 단, 문자열 s2의 길이가 n보다 긴 경우는 잘라낸다.

[참고]

학번 : 이름 :

```
size_t strcspn(const char *dest, const char *src);
```

헤더

- string.h

반환 값

- size_t : 부호없는 정수

매개변수

- dest : 검사할 대상 문자열
- src : 찾고 싶지 않은 문자들의 집합

설명

- complement span
- dest를 처음부터 한 글자씩 검사하면서, 그 글자가 src 문자열 안에 하나라도 포함되어 있는지 확인합니다. 포함된 문자를 처음 만나는 순간 멈추고, 그때까지 지나온 문자 개수를 반환합니다.

예제

```
const char *dest = "Hello, World!";  
const char *src = "ow"; // 'o' 또는 'w'를 찾음  
  
size_t len = strcspn(dest, src);  
printf("길이: %zu\n", len); // 4 출력
```

[코드]

```
#include <stdio.h>  
#include <string.h>  
char *str_ncat(char *s1, const char *s2, int n);  
int main(void){  
    char s1[100];  
    char s2[100];  
    int n;  
  
    printf("첫 번째 문자열(s1) 입력: ");  
    fgets(s1, sizeof(s1), stdin);  
    s1[strcspn(s1, "\n")] = '\0';  
    printf("두 번째 문자열(s2) 입력: ");
```

학번 : 이름 :

```
fgets(s2, sizeof(s2), stdin);
s2[strcspn(s2, "\n")] = '\0';
printf("최대 몇 글자까지 붙일지 입력(n): ");
scanf("%d", &n);
str_ncat(s1, s2, n);
printf("결과 문자열: %s\n", s1);
return 0;
}
```

```
char *str_ncat(char *s1, const char *s2, int n){
    char *tmp = s1;
```

[실행결과]

첫 번째 문자열(s1) 입력: Info↵
두 번째 문자열(s2) 입력: Security↵
최대 몇 글자까지 붙일지 입력(n): 3↵
결과 문자열: InfoSec

(가) *s1 != '\0'
(나) return tmp;
(다) *s1 = '\0'
(라) tmp