

1. 다음의 물음에 답하시오.

1-1. 함수에 대한 설명 중 잘못된 것을 모두 고르시오. (② , ③)

- ① 함수를 호출하려면 반드시 함수의 선언 또는 정의가 필요하다.
- 함수의 정의는 생략할 수 없다.
- 함수 선언 시 매개변수 이름을 생략할 수 없다.
- ④ 함수를 호출할 때는 매개변수의 개수와 인자의 개수가 같아야 한다.
- ⑤ 리턴 값을 갖지 않는 함수는 리턴 값을 생략한다.

1-2. 함수의 리턴형과 매개변수에 대한 설명 중 잘못된 것을 모두 고르시오. (② , ③)

- ① 함수의 리턴형은 함수가 처리 결과로 리턴하는 값의 데이터형이다.
- 함수의 리턴형을 여러 개 지정할 수 있다.
- 함수의 매개변수는 함수 안에 선언된 변수를 말한다.
- ④ 함수의 매개변수의 개수에는 제한이 없다.
- ⑤ 함수 호출 시 넘겨준 인자는 매개변수로 전달된다.
- ⑥ 함수의 매개변수도 일종의 지역변수이다.

1-3. 배열의 사용에 대한 설명 중 올바른 것을 모두 고르시오. (① , ②)

- 배열의 각 원소에 접근하려면 배열의 인덱스를 사용한다.
- 배열의 인덱스는 0부터 시작한다.
- ③ 배열의 각 원소를 사용할 때는 배열의 크기를 신경 쓰지 않아도 된다.
- ④ 크기가 6인 배열은 인덱스로 6까지 사용할 수 있다.
- ⑤ 배열의 인덱스로 잘못된 값을 사용하면 컴파일 에러가 발생한다.

1-4. 다음 중 문자배열의 선언 및 초기화가 잘못된 것은? (④)

- ① `char str[5] = {'1', '2', '3', '4', '\0'};`
- ② `char str[1] = 'A';`
- ③ `char str[5] = "";`
- `char str[] = '1234';`

1-5. 다음과 같이 선언된 문자배열이 있을 때 컴파일 에러가 발생하는 코드는? (②)

`char s[10] = "xyz";`

- ① `s[0] = 'A';`
- `s = "dankook";`
- ③ `s[0] = 0;`
- ④ `scanf("%s", s);`

1-6. 포인터에 대한 설명 중 잘못된 것을 모두 고르시오. (① , ②)

- 포인터는 다른 변수를 가리키는 변수이다.
- 포인터 변수는 다른 변수의 이름을 저장한다.
- ③ 포인터 변수도 변수이므로 주소를 구할 수 있다.
- ④ 포인터 변수의 크기는 포인터가 가리키는 변수의 크기와 같다.
- ⑤ 포인터 변수가 가리키는 변수에 접근하려면 * 연산자를 이용한다.
- ⑥ 변수의 주소를 구할 때는 & 연산자를 이용한다.

1-7. 다음과 같은 int*형의 포인터 변수가 선언되어 있을 때, 포인터 연산자의 결과를 구하시오.

```
int arr[10] = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
int *p = arr;    // p에 저장된 arr의 주소가 0x8000번지라고 가정하자
```

- ① $p + 1 = 0x8004$
- ② $p + 3 = 0x8012$
- ③ $*p = 10$
- ④ $*(p+2) = 30$
- ⑤ $p[5] = 60$

2. 다음의 설명에 부합하는 함수를 아래 박스에서 선택하여 빈칸을 작성하시오.

strcmp(), strtok(), strcat(), strcpy(), strlen(), sprintf(), gest()

- ① strcat() 함수는 하나의 문자열의 끝에 다른 문자열을 연결한다.
- ② strcpy() 함수는 문자열을 복사한다.
- ③ strtok() 함수는 문자열에서 토큰을 찾는다.
- ④ sprintf() 함수는 표준 입력에서 하나의 문자열을 읽는다.
- ⑤ strlen() 함수는 문자열을 이루는 문자의 개수를 반환한다.

3. 다음과 같이 정의되어 있다고 가정하고 다음의 수식의 값을 답하시오.

char str[20] = "First things first";

- ① $str[3] \rightarrow s$
- ② $*str \rightarrow F$
- ③ $*(str + 1) \rightarrow i$

4. 다음 두 문장의 차이점을 설명하시오.

```
char a[] = "Hello World";
char *p = "Hello World";
```

두 문장 중 `char *p = "Hello World";`는 p라는 포인터 변수를 사용하여 문자열을 메모리에 할당하는 상태이고, `char a[] = "Hello World";` a라는 배열을 사용하여 문자열을 메모리에 할당하는 상태이다. 이때, 포인터는 변수로써 선언되었고, 배열은 상수로써 선언되었다는 차이점이 존재한다. p는 고유의 메모리를 차지하고 있고 언제든지 다른 대상을 가리킬 수 있지만, a는 선언할 때 그 위치가 이미 고정되므로 다른 대상을 가리킬 수 없다. a로는 오로지 배열의 선두 번지를 읽을 수 있을 뿐이다.

5. 정수값을 10개 입력받아서 그중 양수의 개수와 음수의 개수, 0의 개수를 출력하는 프로그램을 작성하시오.
(단, 10개의 정수는 배열에 저장해야 하며, 반복문과 조건문을 사용해야 함)

[Source Code]

```
#include <stdio.h>
#define SIZE 10 // 배열 인덱스를 매크로로 정의
int main() {
    /*
        10개의 정수를 입력받기 위해 input이라는 배열을 선언하고,
        각 배열의 요소들을 접근하기 위해 i라는 변수를 선언한다.
        양수의 갯수, 음수의 갯수, 0의 갯수를 출력하기 위한 변수로
        계산을 위해 모두 0으로 초기화 한다.
    */
    int input[SIZE];
    int i;
    int pcnt = 0;
    int ncnt = 0;
    int zcnt = 0;

    printf("10개의 정수를 입력하세요 :");

    /*
        i가 0부터 SIZE(10)까지 1씩 증가되며, i의 값은 배열의 인덱스에 들어가고
        이에 따라 각 배열의 요소별로 데이터를 입력받는다.
    */
    for (i = 0; i < SIZE; i++) {
        scanf("%d", &input[i]); // 10개의 정수 입력
    }

    printf("입력된 배열 :");
    /*
        입력받을 때와 동일하게 i의 값을 증가시키며 각 배열의 요소들을 출력한다.
        또한, if문 통해 양수, 음수, 0을 판별하여, 판별된 수만큼 계속 증가시켜
        양수, 음수, 0의 갯수를 각각 구한다. 이후, printf 함수를 통해 출력한다.
    */
    for (i = 0; i < SIZE; i++) {
        printf("%4d", input[i]); // 입력된 배열
        if (input[i] > 0) // 양수의 개수
            pcnt++;
        if (input[i] < 0) // 음수의 개수
            ncnt++;
        if (input[i] == 0) // 0의 개수
            zcnt++;
    }
    printf("\n양수 = %d개, 음수 = %d개, 0 = %d개\n", pcnt, ncnt, zcnt);
    return 0;
}
```

[Result]

```
10개의 정수를 입력하세요 :
-3
10
2
-4
-6
34
21
-9
0
0
입력된 배열 : -3 10 2 -4 -6 34 21 -9 0 0
양수 = 4개, 음수 = 4개, 0 = 2개
```

6. 사용자로부터 하나의 수를 입력받아 rand() 함수를 사용하여 1부터 100까지의 숫자 중에서 입력받은 수 만큼을 생성하고 그중 가장 큰 수와 가장 작은 수를 출력하는 프로그램을 작성하시오. (srand() 함수를 사용하여, 매번 실행할 때 다른 임의의 수가 발생할 수 있도록 작성해야 함)

[Source Code]

```
#include <stdio.h>
#include <stdlib.h> // rand(). srand()를 위한 헤더파일
#include <time.h> // time()을 위한 헤더파일
#define MAX_SIZE 100
int main() {
    /*
    num, size : 임의의 변수 선언
    random, max, min : 변수 선언 후 변수값 초기화
    */
    int num, size;
    int random = 0;
    int max = 0;
    int min = 100;
    // 사용자로부터 하나의 수를 입력받는다.
    printf("하나의 수를 입력하세요 : ");
    scanf("%d", &size);
    /*
    현재 시간은 프로그램 수행시 매번 달라지므로
    난수 발생기의 시드(seed)를 현재 시간으로 설정
    */
    srand((long)time(NULL));
    // num값이 사용자가 입력한 수보다 작을 때까지 1씩 증가
    for (num = 0; num < size; num++) {
        random = (rand() % MAX_SIZE) + 1; // 1부터 100까지의 난수를 생성
        printf("랜덤으로 생성된 수 : %d\n", random);
        if (random > max) { // random과 max 비교
            max = random; // 최종적으로 가장 큰 값이 max에 저장
        }
        if (random < min) { // random과 min 비교
            min = random; // 최종적으로 가장 작은 값이 min에 저장
        }
    }
    // random값 중 가장 큰 수와 작은 수를 각각 출력
    printf("가장 큰 수 : %d\n", max);
    printf("가장 작은 수 : %d\n", min);
    return 0;
}
```

[Result]

5를 입력했을 경우

하나의 수를 입력하세요 : 5
 랜덤으로 생성된 수 : 16
 랜덤으로 생성된 수 : 66
 랜덤으로 생성된 수 : 84
 랜덤으로 생성된 수 : 23
 랜덤으로 생성된 수 : 18
 가장 큰 수 : 84
 가장 작은 수 : 16

10를 입력했을 경우

하나의 수를 입력하세요 : 10
 랜덤으로 생성된 수 : 73
 랜덤으로 생성된 수 : 70
 랜덤으로 생성된 수 : 28
 랜덤으로 생성된 수 : 82
 랜덤으로 생성된 수 : 98
 랜덤으로 생성된 수 : 44
 랜덤으로 생성된 수 : 71
 랜덤으로 생성된 수 : 78
 랜덤으로 생성된 수 : 56
 랜덤으로 생성된 수 : 60
 가장 큰 수 : 98
 가장 작은 수 : 28

20를 입력했을 경우

하나의 수를 입력하세요 : 20
 랜덤으로 생성된 수 : 90
 랜덤으로 생성된 수 : 96
 랜덤으로 생성된 수 : 46
 랜덤으로 생성된 수 : 25
 랜덤으로 생성된 수 : 6
 랜덤으로 생성된 수 : 35
 랜덤으로 생성된 수 : 47
 랜덤으로 생성된 수 : 100
 랜덤으로 생성된 수 : 49
 랜덤으로 생성된 수 : 37
 랜덤으로 생성된 수 : 62
 랜덤으로 생성된 수 : 20
 랜덤으로 생성된 수 : 12
 랜덤으로 생성된 수 : 69
 랜덤으로 생성된 수 : 21
 랜덤으로 생성된 수 : 63
 랜덤으로 생성된 수 : 47
 랜덤으로 생성된 수 : 27
 랜덤으로 생성된 수 : 78
 랜덤으로 생성된 수 : 43
 가장 큰 수 : 100
 가장 작은 수 : 6

7. 다음 보기와 같은 문자열의 배열을 생성하고, 배열의 요소들을 선택(selection) 정렬하는 프로그램을 작성하시오. 여기에서 선택정렬이란 배열의 모든 요소 중에서 가장 작은 요소를 찾아 배열의 처음 위치에 저장하고, 다시 남은 요소 중에서 가장 작은 요소를 찾아 두 번째 위치에 저장하는 방식으로 이를 반복하는 정렬방법이다. (strcmp0, strcpy0 함수를 사용)

[보기]

{"seoul", "daejeon", "daegu", "kwangju", "inchon", "jeju", "busan"}

[Sorce Code]

```
#include <stdio.h>
#include <string.h> // string 함수 사용 가능하게 하기 위해서
#define Max_index 7
#define Length 10
// selection sort
void selection_sort(char (*city)[Length], int len){
    int i, j, resert;
    char tmp[Length];
    for(i = 0; i < len - 1; i++){
        for(j = i + 1; j < len; j++){
            resert = strcmp(city[i], city[j]); // city[i]와 city[j] 값 서로 비교
            if(resert > 0){ // 참이면
                strcpy(tmp, city[i]); // tmp에다 i넣기
                strcpy(city[i], city[j]); // i에다 j 넣기
                strcpy(city[j], tmp); // j에다 tmp(i) 넣기
            }
        }
    }
}

int main(){
    char city[Max_index][Length] = {"seoul", "daejeon", "daegu", "kwangju", "inchon", "jeju", "busan"};
    int len= sizeof(city)/sizeof(city[0]);
    printf("\n=== 정렬하기 전 ===\n");
    for(int i = 0; i < len; i++){
        printf("%s ", city[i]);
    }
    selection_sort(city, len); // 선택정렬 함수 호출
    printf("\n\n=== 정렬한 다음 ===\n");
    for(int i = 0; i < len; i++){
        printf("%s ", city[i]);
    }

    printf("\n\n");
    return 0;
}
```

[Result]

```
=== 정렬하기 전 ===
seoul daejeon daegu kwangju inchon jeju busan

=== 정렬한 다음 ===
busan daegu daejeon inchon jeju kwangju seoul
```

8. 다음 [보기]와 같이 사용자로부터 텍스트를 입력받아 텍스트를 모두 대문자로 출력하는 프로그램을 작성하시오. (어떤 라이브러리 함수를 사용해도 좋음)

[보기]

텍스트를 입력하시오 : C language is easy
대문자 출력 : C LANGUAGE IS EASY

[Source Code]

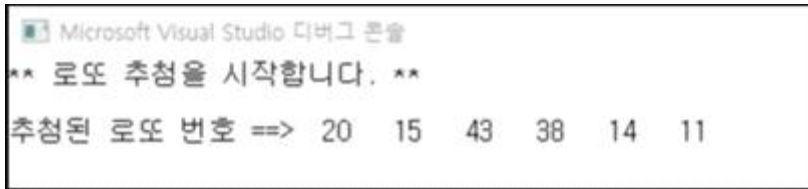
```
#include <stdio.h>
#include <string.h>
int main() {
    char str[] = {};
    printf("\n텍스트를 입력하시오 : ");
    gets(str); // 문자 입력 함수
    printf("대문자 출력 : ");
    while(1) {
        if(strlen(str) == 0) // 문자열 길이 확인
            break; // NULL문자를 만나면 break문을 통해 while문 탈출
        strupr(str); // 대문자로 치환
        puts(str); // 문자 출력 함수
        break;
    }
    printf("\n");
    return 0;
}
```

[Result]

텍스트를 입력하시오 : C language is easy
대문자 출력 : C LANGUAGE IS EASY

9. 로또 숫자 자동 추천 프로그램 작성하기 (1 ~ 45까지 숫자 중에서 숫자 6개를 자동으로 뽑는 프로그램)

[보기]



※ 매번 실행할 때마다 추첨된 로또 번호는 다르게 출력되어야 함

[Source Code]

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define NUM 6
int main() {
    int lotto[NUM];
    int i;
    srand((unsigned)time(NULL));
    printf("\n** 로또 추첨을 시작합니다. **\n추첨된 로또 번호 ==> ");
    for(i = 0; i < NUM; i++) {
        lotto[i] = (rand() % 45) + 1; // 1부터 45까지 난수 생성
        printf("%3d ", lotto[i]); // 생성된 난수 배열의 크기만큼 메모리에 할당받기
    }
    printf("\n\n");
    return 0;
}
```

[Result]

```
PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 10 39 31 10 18 29

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 14 24 22 43 20 33

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 20 11 12 4 39 11

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 23 41 3 29 41 23

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 27 35 2 17 5 35

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 30 28 39 43 15 1

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 33 13 38 23 24 13

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 33 13 38 23 24 13

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 36 7 29 3 26 25

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> ./p9.exe

** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 40 45 28 36 35 36

PS C:\Users\csos\OneDrive - 단국대학교\2019 - 2 DKU 수업자료\고급프로그래밍\3. 과제\vscode_project> █
```