

- 1-1. 함수에 대한 설명 중 잘못된 것을 모두 고르시오. (        )
- ① 함수를 호출하려면 반드시 함수의 선언 또는 정의가 필요하다
  - ② 함수의 정의는 생략할 수 없다
  - ③ 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.

답 : ③, ⑤

- ③ - 함수 선언 시 매개변수의 자료형만 표시하고 이름은 생략할 수 있다.
- ⑤ - 리턴값을 갖지 않는 함수는 void로 지정해야 한다.

- 1-2. 함수의 리턴형과 매개변수에 대한 설명 중 잘못된 것을 모두 고르시오. (        )
- ① 함수의 리턴형은 함수가 처리 결과로 리턴하는 값의 데이터형이다.
  - ② 함수의 리턴형을 여러 개 지정할 수 있다.
  - ③ 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.

답 : ②, ③

- ② - 함수의 리턴형은 단 한 개로 지정해야 한다.
- ③ - 함수의 매개변수는 함수 안에서 선언된 변수가 아닌 전달받은 인자를 받아들이는 변수이다.

- 1-3. 배열의 사용에 대한 설명 중 올바른 것을 모두 고르시오. (        )
- ① 배열의 각 원소에 접근하려면 배열의 인덱스를 사용한다.
  - ② 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.

답: ①, ②

- ① - 배열은 **배열이름[인덱스]** 형식을 가지며 인덱스를 통해 각 원소에 접근할 수 있다.
- ② - 배열의 인덱스는 0부터 전체 원소의 개수보다 1 작은 (n-1) 범위까지이다.

- 1-4. 다음 중 문자배열의 선언 및 초기화가 잘못된 것은? (        )
- ① `char str[5] = {'1', '2', '3', '4', '\0'};`
  - ② `char str[1] = 'A';`
  - ③ `char str[5] = "";`
  - ④ `char str[] = '1234';`

답: ②

② - 문자열 배열의 크기를 설정할 때는 널(\0)값을 포함하여야 한다. `char str[2] = 'A';`

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

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

- ① `s[0] = 'A';`
- ② `s = "dankook";`
- ③ `s[0] = 0;`
- ④ `scanf("%s", s);`

답: ②

② - 이미 문자열이 입력된 배열을 다른 문자열로 대체하려면 배열 한칸에 한글자 씩 할당해주거나 입력을 받아야 한다.

1-6. 포인터에 대한 설명 중 잘못 된 것을 모두 작성하시오. (       )

- ① 포인터는 다른 변수를 가리키는 변수이다.
- ② 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.

답: ②, ④

② - 포인터 변수는 다른 변수의 주소를 저장한다.

④ - 포인터 변수의 크기는 포인터가 가리키는 변수의 크기와 상관없이 항상 같다.

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`
- ② `p + 3`
- ③ `*p`
- ④ `*(p+2)`
- ⑤ `p[5]`

답: ① 0x8004, ② 0x8012, ③ 10, ④ 30, ⑤ 60

- 포인터 변수 p는 배열 arr의 첫원소인 arr[0]의 주소값을 가리키고, p의 i만큼 덧셈은 p가 가리키는 변수인 int형 크기를 i번째 만큼 증가한 주소값을 결과로 가진다.

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

strcmp0, s trtok0, strcat0, strcpy0, strlen0, sprintf0, gest0

- ① \_\_\_\_\_ 함수는 하나의 문자열의 끝에 다른 문자열을 연결한다.
- ② \_\_\_\_\_ 함수는 문자열을 복사한다.
- ③ \_\_\_\_\_ 함수는 문자열에서 토큰을 찾는다
- ④ \_\_\_\_\_ 함수는 표준 입력에서 하나의 문자열을 읽는다.
- ⑤ \_\_\_\_\_ 함수는 문자열을 이루는 문자의 개수를 반환한다.

답: ① strcat(), ② strcpy(), ③ strtok(), ④ gets(), ⑤ strlen()

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

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

- ① str[3] →
- ② \*str →
- ③ \*(str + 1) →

답: ① s, ② F, ③ I

- str은 배열의 첫 원소인 str[0]의 주소값을 가리키고, p의 i만큼 덧셈은 p가 가리키는 변수인 int형 크기를 i번째 만큼 증가한 주소값을 결과로 가진다.

4. 다음 두 문장의 차이점을 설명하시오.

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

답: 포인터 변수 p는 고유의 메모리를 차지하고 있는 변수이지만 배열 a는 선언할 때 그 위치가 고정되어 다른 대상을 가리킬 수 없다.

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

#### 코드

```

1  #include <stdio.h>
2  int main()
3  {
4      int a[10]; // 배열선언
5      int negative=0, positive=0, zero=0; // 음수, 양수, 0의 개수 초기화
6
7      printf("10개의 원소를 입력하세요:\n");
8      for (int i = 0; i < 10; i++) {
9          scanf_s("%d", &a[i]);
10     } //배열 입력
11
12     for (int i = 0; i < 10; i++) {
13         if (a[i] < 0) {
14             negative++; // 배열의 값이 0보다 작으면 음수개수 +1
15         }
16         else if (a[i] > 0) {
17             positive++; // 배열의 값이 0보다 크면 양수개수 +1
18         }
19         else {
20             zero++; // 배열의 값이 0이면 0의개수 +1
21         }
22     }
23
24     printf("입력된 배열: ");
25     for (int i = 0; i < 10; i++) {
26         printf("%d ", a[i]); // 입력된 배열 출력
27     }
28     printf("\n");
29     printf("양수=%d개, 음수=%d개, 0=%d개\n", positive, negative, zero); // 양수, 음수, 0의 개수 출력
30     return 0;
31 }
```

#### 실행 결과

Microsoft Visual Studio 디버그 콘솔

```

10개의 원소를 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 숫자중에서 입력 받은 수 만큼을 생성하고 그 중 가장 큰수와 가장 작은 수를 출력하는 프로그램을 작성하시오.(srand()함수를 사용하여, 매번 실행할 때 다른 임의의 수가 발생할 수 있도록 작성해야 함)

#### 코드

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int main()
5  {
6      int n, random;
7      int min = 100, max = 1; // 변수 min 에 최댓값을, 변수 max에 최솟값을 저장
8      printf("하나의 수를 입력하세요: ");
9      scanf_s("%d", &n); // 하나의 수 입력
10
11     srand(time(NULL)); // 씨드를 시간으로 설정
12     for (int i = 0; i < n; i++) {
13         random = rand() % 100 + 1; // 난수 생성
14         printf("랜덤으로 생성된 수: %d\n", random); // 난수 출력
15         if (random > max) {
16             max = random; // 발생한 난수가 max값 보다 크면 max에 난수 값 저장
17         }
18         if (random < min) {
19             min = random; // 발생한 난수가 min값 보다 작으면 min에 난수 값 저장
20         }
21     }
22
23     printf("가장 큰 수: %d\n", max); // 가장 큰 수 max 출력
24     printf("가장 작은 수: %d\n", min); // 가장 작은 수 min 출력
25     return 0;
26 }

```

#### 실행 결과

Microsoft Visual Studio 디버그 콘솔

```

하나의 수를 입력하세요: 5
랜덤으로 생성된 수: 74
랜덤으로 생성된 수: 95
랜덤으로 생성된 수: 25
랜덤으로 생성된 수: 8
랜덤으로 생성된 수: 78
가장 큰 수: 95
가장 작은 수: 8

```

Microsoft Visual Studio 디버그 콘솔

```

하나의 수를 입력하세요: 10
랜덤으로 생성된 수: 79
랜덤으로 생성된 수: 17
랜덤으로 생성된 수: 56
랜덤으로 생성된 수: 1
랜덤으로 생성된 수: 4
랜덤으로 생성된 수: 25
랜덤으로 생성된 수: 34
랜덤으로 생성된 수: 41
랜덤으로 생성된 수: 23
랜덤으로 생성된 수: 93
가장 큰 수: 93
가장 작은 수: 1

```

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 정렬 방법이다. (strcmp, strcpy 함수를 사용)

[보기]

```
{"seoul", "daejeon", "daegu", "kwangju", "incheon", "jeju", "busan"}
```

코드

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      char str[7][8] = { "seoul", "daejeon", "daegu", "kwangju", "incheon", "jeju", "busan" }; //이차원 배열 선언
7      char temp[8];
8
9      printf("=== 정렬하기 전 ===\n");
10     for (int i = 0; i < 7; i++) {
11         printf("%s ", str[i]); // 정렬 전 문자열 배열 출력
12     }
13     printf("\n\n");
14
15     for (int i = 0; i < 6; i++) {
16         int index = i; // 가장 작은 문자의 위치를 저장하는 index 를 i로 초기화
17
18         for (int j = i + 1; j < 7; j++) {
19             if (strcmp(str[index], str[j]) > 0) {
20
21                 index = j; // str[j]가 str[index]보다 작으면 index에 j를 저장
22             }
23         }
24         strcpy(temp, str[i]);
25         strcpy(str[i], str[index]);
26         strcpy(str[index], temp);
27     } // 선택 정렬을 이용하여 문자열을 오름차순으로 정렬
28
29     printf("=== 정렬한 다음 ===\n");
30     for (int i = 0; i < 7; i++) {
31         printf("%s ", str[i]); // 정렬 후 문자열 배열 출력
32     }
33
34     return 0;
35 }
```

실행 결과

 Microsoft Visual Studio 디버그 콘솔

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

=== 정렬한 다음 ===
busan daegu daejeon incheon 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

코드

```
1  #include <stdio.h>
2  int main()
3  {
4      char text[100];
5      printf("텍스트를 입력하시오 : ");
6      gets(text); // 배열에 문자열 입력
7
8      for (int i = 0; text[i] ; i++) {
9          if (text[i] >= 'a' && text[i] <= 'z') {
10             text[i] = text[i] - 32; // 아스키코드 값을 이용해 소문자를 대문자로 변환
11         }
12     } // 문자열이 끝날 때까지 반복
13
14     printf("대문자 출력: %s", text); // 변환된 문자열 출력
15     return 0;
16
17 }
```

실행결과

 Microsoft Visual Studio 디버그 콘솔

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)

### 코드

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      int lotto[6];
8      srand(time(NULL));
9
10     printf("** 로또 추첨을 시작합니다. **\n\n");
11     printf("추첨된 로또 번호 ==> ");
12
13     for (int i = 0; i < 6; i++) {
14         lotto[i] = rand() % 45 + 1; // 1~45 난수를 생성하여 배열에 저장
15
16         for (int j = 0; j < i; j++) {
17             if (lotto[i] == lotto[j]) {
18                 i--; // 생성된 난수가 이전에 저장된 난수와 중복될 경우 반복 되돌림.
19             }
20         }
21     }
22
23     for (int i = 0; i < 6; i++) {
24         printf("%d ", lotto[i]); // 배열에 저장된 난수 출력
25     }
26     return 0;
27 }
```

### 실행 결과



Microsoft Visual Studio 디버그 콘솔

```
** 로또 추첨을 시작합니다. **
추첨된 로또 번호 ==> 36 14 6 23 21 35
```