C언어 함수 / 배열

App:ple PI 1차시 | 강의자 최재혁

C언어 1차시 목차

배열

1차원 배열과 2차원 배열,
배열이 어디에 쓰이는 지에
대해 알아봅시다

함수

함수와 매개변수, 재귀함수에
대해 알아보면서
함수에 대해 이해해 봅시다

01. 배열

1차원 배열과 2차원 배열,
배열이 어디에 쓰이는 지에 대해 알아봅시다

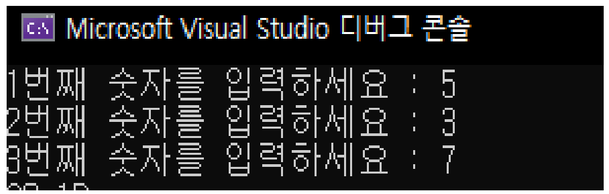
01. 배열은 왜 사용하는가?

예시로 알아보기

C언어로 숫자야구를 구현하려고 하는 재혁이는
문득 이런 의문이 들게 되었다

숫자 야구 게임에는 숫자 3개가 들어가는데
이 3개를 하나의 변수에 담아서 관리할 수 없을까?

랜덤 숫자를 생성하고, 저장 할 공간과
사용자에게 숫자를 입력 받고, 저장 할 공간을
각 하나의 변수에 담는 방법을 찾아보자



하나의
변수로
관리

지금까지 배운 내용으로
왼쪽의 프로그램을 만들 수 있을 지 생각을 해봅시다

↓

하나의 변수명으로 저장?

↓

?????

지금까지 배운 대로 해보려면

각 자리를 각각 하나의 변수에 담아서
저장하는 방법 밖에 없지 않나?

↓

이럴 때 사용할 수 있는 것이 [배열] 이라는 겁니다

02. 1차원 배열은 무엇인가

1차원 배열의 형식

데이터형 배열이름[원소의 개수]

들어갈 수 있는 데이터형(자료형)

형태	이름	크기 (sizeof)
정수형	char	1 byte
	short	2 byte
	Int (기본)	4 byte
	long	4 byte
	long long	8 byte
실수형	float	4 byte
	double (기본)	8 byte
	long double	8 byte 이상

배열의 초기화 방법	
크기를 지정하고 데이터 값을 초기화	int a[5] = { 1, 2, 3, 4, 5 };
크기를 지정 하지 않고 데이터 값을 초기화	int a[ ] = { 1, 2, 3, 4, 5 };
크기를 지정하고 크기보다 적게 초기화	int a[5] = { 1, 2, 3 }; int a[5] = { 1, 2, 3, };

배열은 연속된 기억장소에 동일한 자료형을 갖는
여러 데이터를 저장시키고, 한 변수명을 붙여 관리함

03. 1차원 배열의 저장 방식

<table>
<tr><td colspan="3">배열의 초기화 / 저장 방법</td></tr>
<tr><td>크기를 지정하고
데이터 값을 초기화</td><td>int a[5] = { 1, 2, 3, 4, 5 };</td><td rowspan="2"><table><tr><td>a[0]</td><td>a[1]</td><td>a[2]</td><td>a[3]</td><td>a[4]</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>
크기를 지정 하지 않고 데이터 값을 초기화하는 방법에서는,
컴파일러가 배열의 길이가 자동으로 계산된다</td></tr>
<tr><td>크기를 지정 하지 않고
데이터 값을 초기화</td><td>int a[ ] = { 1, 2, 3, 4, 5 };</td></tr>
<tr><td>크기를 지정하고
크기보다 적게 초기화</td><td>int a[5] = { 1, 2, 3 };
int a[5] = { 1, 2, 3, };</td><td><table><tr><td>a[0]</td><td>a[1]</td><td>a[2]</td><td>a[3]</td><td>a[4]</td></tr><tr><td>1</td><td>2</td><td>3</td><td>0</td><td>0</td></tr></table>
크기보다 적게 초기화 하는 경우,
초기화 되지 않은 칸에는 ‘0’이 들어간다</td></tr>
</table>

04. 1차원 배열을 출력하는 방법

배열의 초기화 / 저장 방법	
크기를 지정하고 데이터 값을 초기화	int a[5] = { 1, 2, 3, 4, 5 };
크기를 지정 하지 않고 데이터 값을 초기화	int a[ ] = { 1, 2, 3, 4, 5 };
크기를 지정하고 크기보다 적게 초기화	int a[5] = { 1, 2, 3 }; int a[5] = { 1, 2, 3, };

배열을 출력하려면 각각의 인덱스에 접근 해야 한다
이 때, 각각의 인덱스는 a[0] = 1 같은 것을 의미함

↓

각 인덱스에 접근하여
하나씩 출력 해줘야 하므로

반복문(for)를 이용하여 출력할 수 있음
(printf로 출력할 수도 있으나, 보통은 반복문 사용)

배열의 출력 예시

```
int a[5] = { 1,2,3,4,5 };

for (int i = 0; i < 5; i++) {
    printf("a[%d] = %d \n", i, a[i]);
}
```

a[접근 할 인덱스] 로 접근 가능함
-> i가 0부터 4까지 반복하므로 전부 출력됨

< 실행 결과 >

```
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
a[4] = 5
```

05. 행렬 이해하기

<별 헤는 밤>

별 하나에 추억(追憶)과
별 하나에 사랑과
별 하나에 쓸쓸함과
별 하나에 동경(憧憬)과
별 하나에 시(詩)와
별 하나에 어머니, 어머니

윤동주의 <별 헤는 밤> 시 중
중간에 나오는 부분을 가져왔습니다

시의 한 줄 한 줄을 뭐라고 하더라?
-> 행이라고 합니다

즉, 가져온 시는 6행의 시

	1열	2열	3열	4열	5열
1행					
2행					

가로 줄이 행 이였다면, 남은 세로 줄은 당연히 열이 되겠죠?

그러므로 위의 표는
2행 5열의 크기를 가진 표가 되는 겁니다.

이것을 왜 배우고 있냐고요?
우리는 지금까지 1행만 있는 1차원 배열을 배웠다면,
다음 페이지에서 2행 이상인 2차원 배열을 배워볼 겁니다.

06. 2차원 배열은 무엇인가

2차원 배열의 형식

= 가로 줄 = 세로 줄

```
데이터형  배열이름[행 원소의 개수][열 원소의 개수]
```

들어갈 수 있는 데이터형(자료형)

형태	이름	크기 (sizeof)
정수형	char	1 byte
	short	2 byte
	Int (기본)	4 byte
	long	4 byte
	long long	8 byte
실수형	float	4 byte
	double (기본)	8 byte
	long double	8 byte 이상

배열의 초기화 방법	
크기를 지정하고 데이터 값을 초기화	int a[2][3] = { { 1, 2, 3 }, { 4, 5, 6 } }; int a[2][3] = { 1, 2, 3, 4, 5, 6 };
크기를 지정 하지 않고 데이터 값을 초기화	int a[ ][3] = { 1, 2, 3, 4, 5, 6 };
크기를 지정하고 크기보다 적게 초기화	int a[2][3] = { 1, 2, 3 }; int a[2][3] = { 1, 2, 3, };

배열은 연속된 기억장소에 동일한 자료형을 갖는 여러 데이터를 저장시키고, 한 변수명을 붙여 관리함

07. 2차원 배열의 저장 방식

<table>
<tr><td colspan="3">배열의 초기화 / 저장 방법</td></tr>
<tr><td>크기를 지정하고
데이터 값을 초기화</td><td>int a[2][3] = { { 1, 2, 3 }, { 4, 5, 6 } };
int a[2][3] = { 1, 2, 3, 4, 5, 6 };</td><td rowspan="2">
<table>
<tr><th></th><th>a[0]</th><th>a[1]</th><th>a[2]</th></tr>
<tr><td>a[0]</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>a[1]</td><td>4</td><td>5</td><td>6</td></tr>
</table>
크기를 지정 하지 않고 데이터 값을 초기화하는 방법에서는,
컴파일러가 배열의 길이가 자동으로 계산된다
</td></tr>
<tr><td>크기를 지정 하지 않고
데이터 값을 초기화</td><td>int a[ ][3] = { 1, 2, 3, 4, 5, 6 };</td></tr>
<tr><td>크기를 지정하고
크기보다 적게 초기화</td><td>int a[2][3] = { 1, 2, 3, 4 };
int a[2][3] = { 1, 2, 3, 4, };</td><td>
<table>
<tr><th></th><th>a[0]</th><th>a[1]</th><th>a[2]</th></tr>
<tr><td>a[0]</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>a[1]</td><td>4</td><td>0</td><td>0</td></tr>
</table>
크기보다 적게 초기화 하는 경우,
초기화 되지 않은 칸에는 ‘0’이 들어간다
</td></tr>
</table>

08. 2차원 배열을 출력하는 방법

배열의 초기화 / 저장 방법	
크기를 지정하고 데이터 값을 초기화	int a[2][3] = { { 1, 2, 3 }, { 4, 5, 6 } }; int a[2][3] = { 1, 2, 3, 4, 5, 6 };
크기를 지정 하지 않고 데이터 값을 초기화	int a[ ][3] = { 1, 2, 3, 4, 5, 6 };
크기를 지정하고 크기보다 적게 초기화	int a[2][3] = { 1, 2, 3, 4 }; int a[2][3] = { 1, 2, 3, 4, };

각 인덱스에 접근하여 하나씩 출력 해줘야 하므로

반복문(for)를 이용하여 출력할 수 있음
(printf로 출력할 수도 있으나, 보통은 반복문 사용)

배열의 출력 예시

```
int a[2][3] = { 1, 2, 3, 4, 5, 6 };

for (int i = 0; i < 2; i++) {   행을 반복하는 부분
   for (int j = 0; j < 3; j++) {    열을 반복하는 부분
      printf("%d ", a[i][j]);
   }                    배열명[행][열]로 접근이 가능함
   printf("\n");
}
```

< 실행 결과 >

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

02. 함수

함수와 매개변수, 재귀함수에 대해 알아보면서
함수에 대해 이해해 봅시다

01. 함수의 종류

printf, scanf와 같은 명령문들은
#include <stdio.h>로 받아와서, 사용이 가능한 함수이다.

ㄴ 당연한 소리 잖아?

ㄴ 사실은 당연한 소리가 아닐지도?

printf, scanf와 같은 함수들은,
stdio.h (헤더파일)에 내장된 함수이고,
이 함수들은 미리 만들어 놔 컴파일러에서
제공하는 "라이브러리 함수" 라고 합니다

라이브러리 함수	프로그래머들이 미리 만들어 놓고, 컴파일러에서 제공하는 함수 (<stdio.h> 헤더 파일에 존재하는 printf, scanf 함수)
사용자 정의 함수	사용자가 직접 정의하여 만들 수 있는 함수

02. 그럼 함수는 왜 쓰는데?

지금까지 모든 예제였던, 프로그램을 만들어볼 때
코드를 다 int main() 함수에 때려 넣었습니다

(문제 1 : 유지보수와 가독성)
만약 코드가 천 줄만 넘어가도 알아보기 힘듦 (어떤 기능인지 판별이 불가함)

: 전체 코드를 적절히 모듈(기능 별로 코드를 묶어)로 나눠 체계적이고, 유지보수가 쉬워진다

(문제 2 : 재활용성)
같은 코드를 또 다시 작성 해야 함

: 기존에 함수를 작성 해 놓았다면, 함수를 호출하기만 해도 사용 가능 (중복 X, 재사용 가능)

03. 함수의 형식과 선언 방법

함수의 형식

```
데이터형  함수명(매개변수){

    함수 내부 코드;
    return 결과 값;

}
```

```
#include <stdio.h>

int sum(int a, int b);

int main() {
    int a = 3, b = 2;
    sum(a, b);
}

int sum(int a, int b){
    return a + b;
}
```

함수의 원형 선언

함수의 호출

함수의 정의

정의 부분이 main() 함수 위에 있다면
함수의 원형 선언 생략 가능

함수의 정의 부분이 뒤에 있을 시,
함수의 원형 선언 생략 불가

함수의 결과 값의 데이터형(자료형)

형태	이름
정수형	char
	short
	Int (기본)
	long
	long long
실수형	float
	double (기본)
	long double

결과 값과, 데이터형은
반드시 일치 해야 한다

04. 매개변수 (call by value)

함수의 형식

```
#include <stdio.h>

int sum(int a, int b);          // 함수의 원형 선언

int main() {
    int a = 3, b = 2;
    sum(a, b);                  // 함수의 호출
    printf("%d", b);
}

int sum(int a, int b){          // 함수의 정의
    b++;
    return a + b;
}
```

Main 함수에 존재하는 b 값은 2이다.
그렇다면, sum 함수를 호출 하고 나서, Main 함수의 b 값이 변하였을까?

```
int main() {
    int a = 3, b = 2;
    printf("%d", sum(a, b)); // 6
    printf("%d", b); // 2
}
```

```
int sum(int a, int b){
    b++;
    return a + b;
}
```

초기 선언된 b 값 = 2
함수 호출 후 b 값 = 2

실 매개변수

sum 함수 안에서의 b 값 = 3

형식 매개변수

→

값만 복사됨
(call by value)

05. 함수의 유형

함수의 형식

```
데이터형  함수명(매개변수){

    함수 내부 코드;
    return 결과 값;

}
```

함수의 유형에 따라 반환 값, 매개변수를 적어줄 수도 있고, 적지 않을 수도 있다

	반환 값이 있는 경우 (return 값이 존재)	반환 값이 없는 경우 (return 값이 존재 X)

매개변수가 있는 경우	int sum (int a, int b){
	  return a+b;
	}
		void sumPrint (int result){
		  printf("%d", result);
		}

매개변수가 없는 경우	int inputNum (void){
	  int a;
	  scanf("%d", &a);
	  return a;
	}
		void welcomePrint(void){
		  printf("출력!");
		}

06. 재귀 함수

재귀호출(recursive call)은, 함수 내부에서 함수가 자기 자신을 다시 호출하는 것을 의미함

계속 호출할 시, 무한으로 호출 됨 : 끝나는 조건 필요

숫자를 입력했을 때, 입력한 수부터
1까지 1씩 빼서 출력하는 함수를 만들고 싶어

ㄴ 1씩 뺀다는 것은, 다시 호출할 때 필요한 규칙임

ㄴ 1까지라는 것은, 끝나는 조건에 해당함

조건에 따라 재귀함수를 만들어보자면?

재귀 호출 예시

```c
#define _CRT_SECURE_NO_WARNINGS
#include <Stdio.h>

void recursiveCall(int num);

int main() {
    int num;
    scanf("%d", &num);

    recursiveCall(num);
}

void recursiveCall(int num) {
    printf("%d", num);
    if (num > 1)  recursiveCall(num - 1);
}
```

< 실행 결과 >

```
5
54321
```

01. 숫자야구의 RULE

게임의 진행자는 3자리의 수를 생각합니다. [ex. 106]

게임의 참여자들은, 진행자가 생각한 3자리 수를 맞추는데,
이 때 진행자는 **Strike(자리와 숫자를 모두 맞춤)**
Ball(숫자만 맞춤)을 알려줍니다.
[만약, 150이라고 했다면 1S 1B이 됨]

10턴 안에 맞추면 맞춘 참여자가 이기는 것이고,
맞추지 못하면 진행자가 이기게 되는 게임

#END. 과제

1차시에 배운 내용을 토대로,
과제를 안내 받고 과제를 수행 해봅시다

오늘의 과제를 알아봅시다

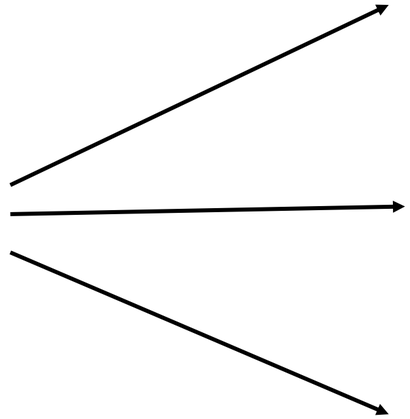
01. 과제 안내 (숫자야구)

숫자야구 게임 만들기
(과제 제출 기한 : ~04/)

기능이 일부 구현 되지 않아도 되니
자신이 할 수 있는 최대한으로 시도 해서 제출해주세요

숫자야구가 뭔데?

랜덤으로 정해진 숫자를
921로 가정할 때



사용자가 숫자를 “568” 이라고 하면,
하나도 맞지 않으므로 0S 0B 출력

사용자가 숫자를 “268” 이라고 하면,
숫자가 하나 맞았으나, 위치가 다르므로 0S 1B 출력
(숫자의 자리가 중요하다는 뜻이겠죠?)

사용자가 숫자를 “935” 이라고 하면,
숫자가 하나 맞았고, 정해진 위치가 같으므로 1S 출력

02. 과제 수행 요구 사항

숫자야구 게임 만들기
(과제 제출 기한 : ~04/)

기능이 일부 구현 되지 않아도 되니
자신이 할 수 있는 최대한으로 시도 해서 제출해주세요

과제 요구 사항

01. 사용자가 프로그램을 실행시켰을 때, (화면 기본 구성)

★ 사용자가 맞출 랜덤 숫자 3자리를 뽑는다

★ "숫자를 입력하세요 : "라고 출력한 뒤, 사용자에게 입력을 받을 수 있도록 한다 (따로 스캔해도 되고, 같이 스캔해도 됨)

02. 사용자가 입력했을 때,

★ 입력을 받은 뒤, 사용자의 숫자에 따라 [ ]S [ ]B의 형식으로 코드를 내보낸다

★ 3S인 경우(위치와 숫자 모두 동일) "게임에서 승리하였습니다, 숫자 : ___" 를 내보낸다 (이 때, 숫자는 랜덤 생성 숫자를 출력)

★ 3S이 아닌 경우, 사용자가 다시 숫자를 맞출 수 있도록 한다

★ 오늘 수업이 함수와 배열 이었으므로, 최소 함수 1개, 배열 1개를 충족하도록 한다 ★

03. 과제 제출 방법 안내

숫자야구 게임 만들기
(과제 제출 기한 : ~04/)

기능이 일부 구현 되지 않아도 되니
자신이 할 수 있는 최대한으로 시도 해서 제출해주세요

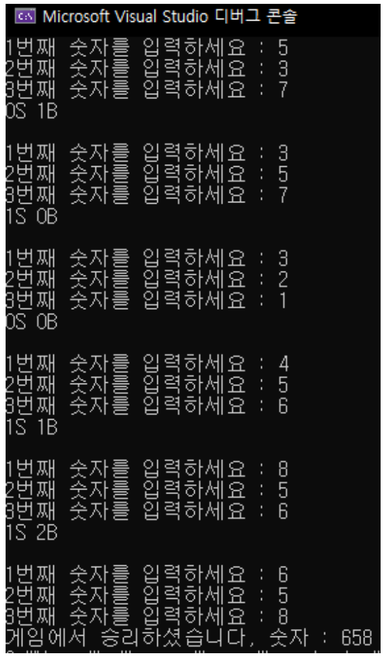
과제 제출 방법

아래 안내 된 구글 폼 주소로 들어가서 제출 !!
제출할 때, 간단한 설문지가 있으니 설문 조사에 꼭 참여해줄 것

과제 파일은 어떻게 내도 상관 없으나 소스코드는 반드시 내야 함
(한글 파일, .c 파일, 워드, 메모장, ppt 모두 상관 X)

구글 폼 링크 :: https://forms.gle/XyhxSCY5ZQ6EfuG26

과제에 대한 질문이 있을 경우
디스코드 : 재혁 #2293, 인스타 : @jaehyeok3017,
애플파이 12,13기 단체 카톡을 이용하여 질문을 하시면 됩니다.



C언어 1차시 끝 :)

위 페이지에서 제시하였던
제출 기한까지 반드시 제출해주세요 !