

1. C언어의 구조

소스코드	설명
예1) <pre> 1 #include<stdio.h> 2 void main() { 3 4 } 5 </pre>	1줄:standard(표준)input/output(입출력)을 사용하기 위해 반드시 include(포함)시켜야 한다. 2줄:C 언어는 main()함수가 몸체이며 실행시에 main()함수부터 실행한다. 앞에 붙는 void는 반환되는 값의 type(자료형)이 없음을 의미한다.
예2) <pre> 1 #include<stdio.h> 2 void main() { 3 printf("Hello World\n"); 4 } 5 </pre>	3줄:printf()함수는 Console(콘솔)화면에 출력하기 위한 명령어이다. printf()안의 " "에 출력할 내용을 넣는다. 명령어(문장)의 마지막에는 ; (세미콜론)로 끝낸다.
예3) <pre> 1 #include<stdio.h> 2 void main() { 3 printf("Hello World\n"); 4 printf("Next Line\n"); 5 } </pre>	3줄:printf()함수안의 " "에 Wn(Wn)는 줄넘김을 의미한다.
예4) <pre> 1 #include<stdio.h> 2 int main() { 3 printf("Hello World\n"); 4 printf("Next Line\n"); 5 return 0; 6 } </pre>	2줄:main()함수 앞의 int는 반환(return)되는 값(0)이 정수형(int)임을 의미한다.

4. 변환포맷

1) 숫자와 문자포맷

숫자포맷

정수포맷	%d (int형의 포맷)
실수포맷	%f (float형의 포맷) %lf (double형의 포맷)

문자포맷

문자포맷	%c (char형의 포맷)
문자열포맷	%s (char형 배열의 포맷)

※ 출력자릿수의 예

명령	출력상태
----	------

①	printf("%d", 4);	4				
②	printf("%5d", 4);					4
③	printf("%05d", 4);	0	0	0	0	4
④	printf("%-5d", 4);	4				
⑤	printf("%5c", 'a');					a
⑥	printf("%-5c", 'a');	a				
⑦	printf("%5s", "AM");				A	M
⑧	printf("%-5s", "AM");	A	M			
⑨	printf("%5.2f", 1.5);		1	.	5	0
⑩	printf("%05.2f", 1.5);	0	1	.	5	0
⑪	printf("%-5.2f", 1.2);	1	.	2	0	

2. 배열

배열을 사용하면 같은 종류의 대량의 데이터를 효율적으로 간편하게 처리할 수 있다.

- 변수 활용한 저장 공간 선언 ; int a0, s1, s2, s3, s4, s5, s6, s7, s8, s9;
- 배열 활용한 저장 공간 선언 ; int a[10];

1. 배열의 사용

- 1차원 배열: 동일한 자료형으로 구성되어진 기억 공간들에 동일한 자료들을 각각 일괄적으로 처리시키기 위하여 사용하는 자료형 예) int M[10];
- 2차원 배열: 행과 열로 구성된 배열을 말하며 2차원 배열은 1차원 배열들의 모임이다.
- 3차원 배열: 면과 행 그리고 열로 구성된 배열을 말하며 3차원 배열은 2차원 배열들의 모임을 말한다.

1차원 배열의 선언방법

자료형 배열명[갯수];	예) int a[5]; <table><tr><td></td><td></td><td></td><td></td><td></td></tr></table> a[0] a[1] a[2] a[3] a[4]										
자료형 배열명[갯수]={값1,값2,값3,..값n};	예) int a[5]={1, 10, 20, 5, 12}; int a[]={1, 10, 20, 5, 12}; char ch[]="korea"; <table><tr><td>1</td><td>10</td><td>20</td><td>5</td><td>12</td></tr></table> a[0] a[1] a[2] a[3] a[4] <table><tr><td>k</td><td>o</td><td>r</td><td>e</td><td>a</td></tr></table> ch[0] ch[1] ch[2] ch[3] ch[4]	1	10	20	5	12	k	o	r	e	a
1	10	20	5	12							
k	o	r	e	a							

2차원 배열의 선언방법

자료형 배열명[행갯수][열갯수];	
-----------------------	--

	<div>예) int a[2][3];</div> <table><tr><td>a[0][0]</td><td>a[0][1]</td><td>a[0][2]</td></tr><tr><td>a[1][0]</td><td>a[1][1]</td><td>a[1][2]</td></tr></table>	a[0][0]	a[0][1]	a[0][2]	a[1][0]	a[1][1]	a[1][2]
a[0][0]	a[0][1]	a[0][2]					
a[1][0]	a[1][1]	a[1][2]					
<div>자료형 배열명[행갯수][열갯수]= {값1,값2,값3,..값n};</div>	<div>예) int a[2][3]={1,2,3,4,5,6}; int a[][3]={1,2,3,4,5,6};</div> <table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></table>	1	2	3	4	5	6
1	2	3					
4	5	6					

3차원 배열의 선언방법

자료형 배열명[면][행][열];	예) int a[2][3][2]; <table><tr><td>a[0][0][0]</td><td>a[0][0][1]</td><td>a[1][0][0]</td><td>a[1][0][1]</td></tr><tr><td>a[0][1][0]</td><td>a[0][1][1]</td><td>a[1][1][0]</td><td>a[1][1][1]</td></tr><tr><td>a[0][2][0]</td><td>a[0][2][1]</td><td>a[1][2][0]</td><td>a[1][2][1]</td></tr></table>	a[0][0][0]	a[0][0][1]	a[1][0][0]	a[1][0][1]	a[0][1][0]	a[0][1][1]	a[1][1][0]	a[1][1][1]	a[0][2][0]	a[0][2][1]	a[1][2][0]	a[1][2][1]
a[0][0][0]	a[0][0][1]	a[1][0][0]	a[1][0][1]										
a[0][1][0]	a[0][1][1]	a[1][1][0]	a[1][1][1]										
a[0][2][0]	a[0][2][1]	a[1][2][0]	a[1][2][1]										
자료형 배열명[면][행][열]= {값1,값2,값3,..값n};	예) int a[2][3][2]= {1,2,3,4,5,6,7,8,9,10,11,12}; int a[][3][2]= {1,2,3,4,5,6,7,8,9,10,11,12}; <table><tr><td>1</td><td>2</td><td>7</td><td>8</td></tr><tr><td>3</td><td>4</td><td>9</td><td>10</td></tr><tr><td>5</td><td>6</td><td>11</td><td>12</td></tr></table>	1	2	7	8	3	4	9	10	5	6	11	12
1	2	7	8										
3	4	9	10										
5	6	11	12										

1.1. 1차원 배열

실습1) 1차원배열에 저장된 값을 출력하기

1	void main()
2	{
3	int M[5]={10,20,30};
4	M[0]=5;
5	M[3]=40;
6	printf("%d, %d, %d, %d, %d\n", M[0], M[1], M[2], M[3], M[4]);
7	
8	}

실습2) grade배열에 저장된 값들을 출력하는 프로그램

1	void main()
2	{
3	int grade[10]={31,63,62,87,14,25,92,70,75,53};
4	int i;
5	for(i=0;i<10;i++)
6	{
7	printf("%5d\n", grade[i]);
8	}
9	}

실습3) grade배열에 저장된 값들의 평균을 출력하는 프로그램

```
1 void main()
2 {
3     int grade[10]={31,63,62,87,14,25,92,70,75,53};
4     int i, size, hap=0;
5     double avg;
6     size=sizeof(grade)/sizeof(grade[0]);
7     for(i=0;i<size;i++)
8     {
9         hap=hap+grade[i];
10    }
11    avg=hap/size;
12    printf("%lf \n", avg);
13 }
14
```

실습4) a배열에 저장된 값을 b배열로 복사하는 프로그램

```
1 #include<stdio.h>
2 #define SIZE 5
3 void main()
4 {
5     int i;
6     int a[SIZE]={1,2,3,4,5};
7     int b[SIZE];
8     for(i=0;i<SIZE;i++)
9     {
10        b[i]=a[i];
11    }
12 }
13
```

3. 포인터와 포인터 변수

- 포인터란 메모리상의 임의의 위치 값을 말하며 주소, 번지라고도 하며 1바이트 간격으로 부여된 위치 값을 말한다.
- 포인터 변수란 메모리상의 위치 값을 기억시켜 둘 목적으로 사용하는 기억공간을 말한다.

(1) 포인터의 표기

& : 시작주소 예) &a : a의 시작주소

* : 포인터 예) *a : 포인터 변수 a

포인터 변수

1) 포인터 변수 선언	자료형 <u>*변수명1</u> ; 자료형 변수명2;	char *ap; float *cp; int *bp; int b;	<div>주소 변수 기억장소</div> <div> </div>
2) 포인터 변수에 주소 저장	<u>변수명1=&변수명2</u> ;	bp=&b;	
3) 포인터 변수를 통하여 상수값 저장	<u>*변수명1=상수값</u> ;	*bp=100;	
1 void main() 2 { 3 char *ap; 4 int *ap; 5 float *cp; 6 int a; 7 ap=&a; 8 *ap=10; 9 } 10	3줄:char형의 기억공간의 시작주소를 저장하는 주소변수 5줄:int형의 기억공간의 시작주소를 저장하는 주소변수 6줄:float형의 기억공간의 시작주소를 저장하는 주소변수 8줄:a의 주소를 ap포인터 변수에 저장해 둔다 9줄:ap포인터변수에 저장된 “주소를 참조”한 저장 공간에 10을 저장 포인터 변수 앞에 *이 붙으면 포인터 변수에 담긴 “주소를 참조”한 저장 공간이다.		

실습1) 포인터 변수의 사용 예

1 void main() 2 { 3 int a=10, b=20; 4 int *ptr1, *ptr2; 5 6 ptr1=&a; 7 ptr2=&b; 8 printf("a=%d, %uWn", a, &a); 9 printf("b=%d, %uWn", b, &b); 10 printf("ptr1의 주소 %uWn", &ptr1); 11 printf("ptr2의 주소 %uWn", &ptr2); 12 printf("%dWn", *ptr1); 13 printf("%dWn", *ptr2); 14 15 } 16	3줄:a,b변수는 일반변수이며 상수를 저장 4줄:ptr1, ptr2변수는 포인트 변수이며 주소를 저장. 4줄:int *prt1=&a; 포인터 변수의 선언과 할당을 동시에 가능 6줄:ptr1변수에 a의 주소를 저장 7줄:ptr2변수에 b의 주소를 저장 12줄:ptr1변수에 저장된 주소에 해당하는 기억공간	<div>변수명 메모리 주소</div> <div> </div>
---	--	--

실습2) 포인터 변수를 이용한 값 저장

1 void mai() { 2 int a, *ap 3 int b, *bp	
--	--

실습3) 포인터 변수를 이용한 연산

1 void main(){ 2 int a,b,c,d; 3 int *ap, *bp;	
---	--

4	ap=&a;	4	ap=&a;
5	bp=&b;	5	bp=&b;
6	*ap=10;	6	*ap=5;
7	*bp=20;	7	*bp=3;
8	printf("%d Wn", *ap);	8	c=*ap+*bp;
9	printf("%d Wn", *bp);	9	d=a+b;
10	}	10	printf("c=%dWn", c);
11		11	printf("d=%dWn", d);
12		12	printf("*ap=%uWn", ap);
13		13	printf("%dWn", *ap);
14		14	printf("*ap=%uWn", &a);
15		15	}
16		16	
결과 또는 설명		결과 또는 설명	

실습4) 값전달 방식의 swap() 함수

```

1 void swap(int a, int b)
2 {
3     int temp;
4     temp=a;
5     a=b;
6     b=temp;
7 }
8 void main()
9 {
10    int a=10, b=20;
11    printf("a=%d, b=%dWn", a, b);
12    swap(a, b);
13    printf("a=%d, b=%dWn", a, b);
14 }
15

```

결과 또는 설명

실습5) 주소전달 방식의 swap() 함수

```

1 void swap(int *a, int *b)
2 {
3     int temp;
4     temp=*a;
5     *a=*b;
6     *b=temp;
7 }
8 void main()
9 {
10    int a=10, b=20;
11    printf("a=%d, b=%dWn", a, b);
12    swap(&a, &b);
13    printf("a=%d, b=%dWn", a, b);
14 }
15
16

```

결과 또는 설명

포인터와 배열의 관계 (배열명자체가 주소임 즉 a 가 주소)

```
int a[10]={10,20,30,40,50,60,70,80,90,100};  
printf("%d\n", a[0]);    //10  
printf("%d\n", a[1]);    //20  
printf("%d\n", a[2]);    //30  
printf("%d\n", a[3]);    //40  
printf("%d\n", *(a+0));  //또는 *a 같음 //10  
printf("%d\n", *(a+1));  //20  
printf("%d\n", *(a+2));  //30  
printf("%d\n", *(a+3));  //40  
printf("%d\n", *(a+3)+2); //42
```