



## C기초 문제지(2)

### 문제 C21-0001

10 크기의 정수 배열을 만들고,  
5에서 9 사이의 값으로 10개의 빈 공간을 채우고(랜덤),  
아래와 같이 배열의 내용을 출력하는 프로그램을 작성하시오

- 입출력 예

입력1	출력1
없음	rand arr[5~9] : 5 7 8 7 9 6 8 5 7 8

```
d:\CloudStation_CodeHub\VisualStudioCode\ch01>cd "d:\CloudStation_CodeHub\ch01_03t && "d:\CloudStation_CodeHub\VisualStudioCode\ch01\"ch01_03t
rand arr[5~9] : 7 8 7 7 6 7 5 7 5
d:\CloudStation_CodeHub\VisualStudioCode\ch01>
```

### 문제 C21-0002

0~9 사이의 10개 임의의 값을 생성하고,  
짝수는 A 배열에 홀수는 B 배열에 각각 분리해서 담아 각각 출력하는 코드를 작성해보자.

- 입출력 예

```
d:\CloudStation_CodeHub\VisualStudioCode\ch02>cd "d:\CloudStation_CodeHub\VisualStudioCode\ch02\"ch02_01_clustering
2 7 2 1 5 3 0 5 2 5
arrOdd: 7 1 5 3 5 5
arrEven: 2 2 0 2
```

입력1	출력1
없음	6 1 2 0 3 3 0 9 6 6 arrEven : 6 2 0 0 6 6 arrOdd : 1 3 3 9

## 문제 C21-0003

아래 코드는 6,3 두개의 숫자를 배열의 첫번째 두번째 위치에 저장하고, 두 위치의 값을 SWAP 해서 출력값을 비교하는 프로그램의 일부이다. 프로그램을 완성하시오.

- 코드 작성 부분을 채우시오.

```
#include <stdio.h> void main() { int arr[10] = {6,3}; printf("before :"); for(int i=0; i<2; i++) printf("%d ", arr[i]); printf("\n"); //////////////// 코드 작성 ///////////
//////////////////////////// printf("after :"); for(int i=0; i<2; i++) printf("%d ", arr[i]); }
```

- 입출력 예

```
if ($?) { .\test }
before : 6 before : 3
after : 3 after : 6
PS C:\test> cd "c:\test\" ; if (?) { gcc test.c -o test } ;
if (?) { .\test }
before :6 3
after :3 6
PS C:\test>
```

## 문제 C21-0004

아래 코드는 배열의 값을 왼쪽과 오른쪽으로 shift 하는 프로그램의 일부이다.

- 조건
  - 왼쪽으로 한번, 오른쪽으로 두번 shift 하는 프로그램을 완성하시오.
  - 코드 작성 부분을 채우시오.

```
#include <stdio.h> #include <stdlib.h> #include <time.h> void main() { int
arr[10] = {1,2,3,4,5}; printf("%d \n", sizeof(arr)); //물리적메모리크기 printf("%d
\n",sizeof(arr)/sizeof(int)); //저장공간갯수 int arr_cnt = 5; //배열의 사용크기를 알려주어야
한다. printf("초기값 :"); for(int i=0; i<arr_cnt; i++) printf("%d ", arr[i]);
printf("\n"); //////////////// 코드 작성 ///////////////////////////////// printf("left shift
:"); for(int i=0; i<arr_cnt; i++) printf("%d ", arr[i]); printf("\n"); //////////////// 코드
작성 ///////////////////////////////// printf("right shift :"); for(int i=0;
i<arr_cnt; i++) printf("%d ", arr[i]); printf("\n"); for(int i=arr_cnt-1; i>0; i-
) { int tmp; tmp = arr[i]; arr[i] = arr[i-1]; arr[i-1] = tmp; } printf("right
shift :"); for(int i=0; i<arr_cnt; i++) printf("%d ", arr[i]); printf("\n"); }
```

shift 첫번째 예제

- 입출력 예

```
int arr[10] = {1,2,3,4,5};

문제 출력 디버그 콘솔 터미널 2: Code
```

```
초기값 :1 2 3 4 5
left shift :2 3 4 5 1
right shift :1 2 3 4 5
right shift :5 1 2 3 4

d:\CloudStation_CodeHub\VisualStudioCode\ch05>
```

## 문제 C22-0001

다음 프로그램의 결과를 작성하시오.

```
void main() { char a=3,b=6; printf("%d %d\n", a>>1,b>>1); // (1) printf("%d %d\n", a<<1,b<<1); // (2) printf("%d %d\n", a&1,a&b); // (3) printf("%d %d\n", a&&1,a&&b); // (4) printf("%d %d\n", ~a,!a); // (5) }
```

- 정답

번호	정답
(1)	1 3
(2)	6 12
(3)	1 2
(4)	1 1
(5)	-4 0

## 문제 C22-0002

아래 프로그램의 출력결과를 작성해보자.

- 조건
  - (7) 번은 sarr 배열의 원소 개수 5가 출력하도록 sizeof 연산자를 이용해서 프로그램하시오.

```
#include <stdio.h> void main() { short s = 3; int i = 4; float f = 5.f; char c = 'A'; short sarr[] = {'H','e','l','l','o'}; printf("%d\n" , sizeof(s)); // (1) printf("%d\n" , sizeof(i)); // (2) printf("%d\n" , sizeof(f)); // (3) printf("%d\n" , sizeof(-5.0)); // (4) printf("%d\n" , sizeof(5)); // (5) printf("%d\n" , sizeof(sarr)); // (6) _____ // (7) }
```

(1)	(2)	(3)	(4)	(5)	(6)
2	4	4	8	4	10

(7)

```
char sarr1[] = {'H','e','l','l','o'};  
printf("%d\n" , sizeof(sarr1)); // (7)
```

## 문제 C22-0003

다음 프로그램의 결과와 메모리 맵을 그리시오.

- 조건
  - 메모리맵은 주소, 이름, 값 3가지값을 모두 그리시오.

```
#include <stdio.h> void main() { char arr[] = "Hello"; arr[2] = '0'; arr[3] = '\0'; arr[4] = '0'; printf("%s", arr); }
```

제작 제작 신 1

추가: "13"

제작 제작 신 1

추가: "61"

제작 제작 신 1

추가: "12"

제작 제작 신 1

추가: "11"

제작 제작 신 1

추가: "-4"

## 문제 C23-0001

1바이트 메모리 영역을 할당하고,

ASCII 테이블을 참고해서 2진수 01000011을 메모리에 저장하는 5가지 방법을 각각 작성하고, 모두 같은 값이 출력되는 프로그램을 작성하시오.

- 조건
  - 키보드 입력 없음
  - 5가지 방법을 모두 작성하시오.
  - 메모리 입력(변수값설정) 과 출력(변수값 참조, print() 이용)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>Ø</b>	96	60	140	&#96;	<b>`</b>
1	1 001	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2 002	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3 003	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4 004	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>:</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5 005	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6 006	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7 007	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8 010	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9 011	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A 012	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B 013	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C 014	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D 015	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E 016	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F 017	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10 020	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11 021	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12 022	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13 023	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14 024	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15 025	025	<b>NAK</b> (negative acknowledgement)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16 026	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17 027	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18 030	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19 031	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A 032	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B 033	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>:</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C 034	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D 035	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E 036	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F 037	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: www.LookupTables.com

제우 제우 신 1  
추가: "ch"