

# Data Structure & Algorithm 자료구조 및 알고리즘

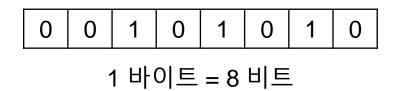
4. 구조체와 포인터 (Structures and Pointers)



### 정보의 표현



- 1 비트(bit, b) = 0 또는 1
- 1 바이트(byte, B) = 8비트



- 1 바이트로 표현할 수 있는 것?
  - 1바이트 = 8비트, 2<sup>8</sup> = 256개의 서로 다른 값
  - -128부터 127까지의 정수
  - 0부터 255까지의 정수
  - 256개의 다른 문자 (소문자 26개, 대문자 26개, 특수기호 등)

### 정보의 표현



- 4 바이트로 표현할 수 있는 것?
  - 4바이트 = 32비트, 2<sup>32</sup>= 4,294,967,296개의 서로 다른 값
  - -2,147,483,648부터 2,147,483,647까지의 정수
  - 0부터 4,294,967,295까지의 정수
  - 0.25 등의 실수
- 8 바이트로 표현할 수 있는 것?
  - 8바이트 = 64비트, 2<sup>64</sup> 개의 서로 다른 값
  - $-2^{63}$  부터  $2^{63}$  -1 까지의 정수
  - 0부터 2<sup>64</sup> 1까지의 정수
  - 0.25 등의 실수 (더욱 정확하게)

## 정보의 표현



- 1 킬로바이트(KB) = 1,000 바이트
- 1 메가바이트(MB) = 1,000 킬로바이트
- 1 기가바이트(GB) = 1,000 메가바이트
- 1 테라바이트(TB) = 1,000 기가바이트
- 1 페타바이트(PB) = 1,000 테라바이트
- 1,000이 약 2<sup>10</sup>이므로,
  - $4 \text{ KB} = 4 * 2^{10} \text{ B}$
  - 16 GB = 16 \*  $2^{30}$  B

### C 언어에서의 데이터



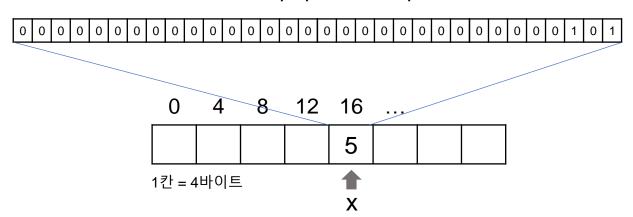
- 타입(Type): 데이터의 형태
  - 정수형: -5, 0, 128 등
  - 실수형: -3.5, 0.005, 28.5 등
  - 문자형: 'A', 'x', '5', '@' 등
- char: 1바이트 (-128 ~ 127)
- int: 4바이트 (-2,147,483,648부터 2,147,483,647)
- float: 4바이트 (단정도, 6-7 유효 숫자)
- double: 8바이트 (배정도, 15-16 유효 숫자)

### 데이터를 담는 변수



• 변수(Variable): 데이터를 담는 이름이 붙여진 메모리 상의 공간

```
1 #include <stdio.h>
2
3 int main() {
4    int x = 5;
5
6    printf("%d\n",x);
7    return 0;
8 }
```



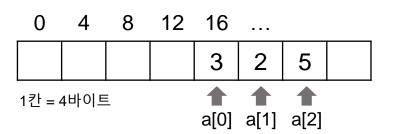
- 16 GB 메모리에는 int 형 변수 몇 개?
  - 16 GB / 4 B = 4 G개 = 4,000,000,000 = 40억개

## 배열



• 배열(Array): 연속적인 메모리 공간에 저장된 데이터

```
1 #include <stdio.h>
2
3 int main() {
4    int a[3] = {3, 2, 5};
5
6    printf("%d\n", a[0] + a[1] + a[2]);
7    return 0;
8 }
9
10
```

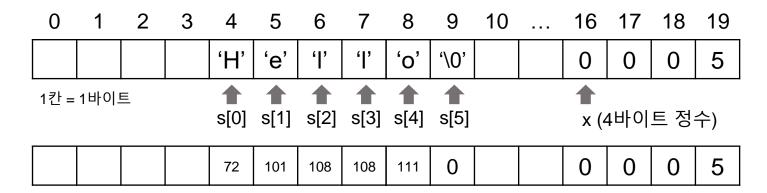


## 배열



• 배열(Array): 연속적인 메모리 공간에 저장된 데이터

```
1 #include <stdio.h>
2
3 int main() {
4    char s[10] = "Hello";
5    int x = 5;
6
7    printf("%s %d\n", s, x);
8    return 0;
9 }
```





### **ASCII Table**

| 0 0 0 0 32 20 40 [space] 64 40 100 @ 96 1 1 1 1 33 21 41 ! 65 41 101 A 97 2 2 2 2 3 34 22 42 " 66 42 102 B 98 3 3 3 3 3 35 23 43 # 67 43 103 C 99 4 4 4 4 4 36 68 44 104 D 100 5 5 5 5 37 25 45 % 69 45 105 E 101 6 6 6 6 38 26 46 & 70 46 106 F 102 7 7 7 7 39 27 47 ' 71 47 107 G 103 8 8 10 40 28 50 ( 72 48 110 H 104 9 9 11 41 29 51 ) 73 49 111 I 105 10 A 12 42 2A 52 * 74 4A 112 J 106 11 B 13 43 2B 53 + 75 4B 113 K 107 12 C 14 44 2C 54 . 76 4C 114 L 108 13 D 15 45 2D 55 - 77 4D 115 M 109 14 E 16 46 2E 56 . 78 4E 116 N 110 15 F 17 47 2F 57 / 79 4F 117 O 111 16 10 20 48 30 60 0 80 50 120 P 112 17 11 21 49 31 61 1 81 51 121 Q 113 18 12 22 50 32 62 2 82 52 122 R 114 19 13 23 51 33 63 3 83 53 123 S 115 20 14 24 52 34 66 4 4 84 54 124 T 116 21 15 25 53 35 66 5 85 55 125 U 117 22 16 26 54 36 66 6 86 56 126 V 118 23 17 27 55 37 67 7 87 57 127 W 119   | 60<br>61<br>62<br>63<br>64<br>65 | 140<br>141<br>142<br>143<br>144 | a<br>b<br>c |
|---|----------------------------------|---------------------------------|-------------|
| 1       1       1       33       21       41       !       65       41       101       A       97         2       2       2       2       42       "       66       42       102       B       98         3       3       35       23       43       #       67       43       103       C       99         4       4       4       36       24       44       \$       68       44       104       D       100         5       5       5       5       37       25       45       %       69       45       105       E       101         6       6       6       6       38       26       46       &       70       46       106       F       102         7       7       7       39       27       47       '       71       47       107       G       103         8       8       10       40       28       50       (       72       48       110       H       104         9       9       11       41       29       51       )       73       49       111< | 62<br>63<br>64<br>65             | 142<br>143                      | b           |
| 3       3       3       35       23       43       #       67       43       103       C       99         4       4       4       4       4       \$       68       44       104       D       100         5       5       5       5       37       25       45       %       69       45       105       E       101         6       6       6       6       38       26       46       &       70       46       106       F       102         7       7       7       39       27       47       '       71       47       107       G       103         8       8       10       40       28       50       (       72       48       110       H       104         9       9       11       41       29       51       )       73       49       111       I       105         10       A       12       2A       52       *       74       4A       112       J       106         11       B       13       43       28       53       +       75       48        | 63<br>64<br>65                   | 143                             |             |
| 3       3       3       35       23       43       #       67       43       103       C       99         4       4       4       4       4       \$       68       44       104       D       100         5       5       5       5       37       25       45       %       69       45       105       E       101         6       6       6       6       38       26       46       &       70       46       106       F       102         7       7       7       39       27       47       '       71       47       107       G       103         8       8       10       40       28       50       (       72       48       110       H       104         9       9       11       41       29       51       )       73       49       111       I       105         10       A       12       2A       52       *       74       4A       112       J       106         11       B       13       43       28       53       +       75       48        | 64<br>65                         |                                 | C           |
| 5     5     5     5     45     %     69     45     105     E     101       6     6     6     38     26     46     &     70     46     106     F     102       7     7     7     7     39     27     47     '     71     47     107     G     103       8     8     10     40     28     50     (     72     48     110     H     104       9     9     11     41     29     51     )     73     49     111     I     105       10     A     12     42     2A     52     *     74     4A     112     J     106       11     B     13     43     2B     53     +     75     4B     113     K     107       12     C     14     44     2C     54     .     76     4C     114     L     108       13     D     15     45     2D     55     -     77     4D     115     M     109       14     E     16     46     2E     56     .     78     4E     116     N     110   | 65                               | 144                             |             |
| 6 6 6 6 7 7 7 7 8 9 27 47 1 7 1 47 107 G 103 8 8 10 40 28 50 ( 72 48 110 H 104 9 9 11 41 29 51 ) 73 49 111 I 105 10 A 12 42 2A 52 * 74 4A 112 J 106 11 B 13 43 2B 53 + 75 4B 113 K 107 12 C 14 44 2C 54 . 76 4C 114 L 108 13 D 15 45 2D 55 - 77 4D 115 M 109 14 E 16 46 2E 56 . 78 4E 116 N 110 110 111 110 1111 1111 1111 11   |                                  |                                 | d           |
| 7 7 7 7 8 8 8 10 40 28 50 ( 72 48 110 H 104 9 9 11  1   |                                  | 145                             | e           |
| 8 8 10 40 28 50 ( 72 48 110 H 104 9 9 11  | 66                               | 146                             | f           |
| 9 9 11 41 29 51 ) 73 49 111 I 105 10 A 12 42 2A 52 * 74 4A 112 J 106 11 B 13 43 2B 53 + 75 4B 113 K 107 12 C 14 44 2C 54 . 76 4C 114 L 108 13 D 15 45 2D 55 - 77 4D 115 M 109 14 E 16 46 2E 56 . 78 4E 116 N 110 15 F 17 47 2F 57 / 79 4F 117 O 111 16 10 20 48 30 60 0 80 50 120 P 112 17 11 21 49 31 61 1 81 51 121 Q 113 18 12 22 50 32 62 2 82 52 122 R 114 19 13 23 51 33 63 3 83 53 123 S 115 20 14 24 52 34 64 4 84 54 124 T 116 21 15 25 53 35 65 5 85 55 125 U 117 22 16 26 54 36 66 6 86 56 126 V 118   | 67                               | 147                             | g           |
| 10 A 12 42 2A 52 * 74 4A 112 J 106 11 B 13 43 2B 53 + 75 4B 113 K 107 12 C 14 44 2C 54 , 76 4C 114 L 108 13 D 15 45 2D 55 - 77 4D 115 M 109 14 E 16 46 2E 56 . 78 4E 116 N 110 15 F 17 47 2F 57 / 79 4F 117 O 111 16 10 20 48 30 60 0 80 50 120 P 112 17 11 21 49 31 61 1 81 51 121 Q 113 18 12 22 50 32 62 2 82 52 122 R 114 18 12 22 50 32 62 2 82 52 122 R 114 19 13 23 51 33 63 3 83 53 123 S 115 20 14 24 52 34 64 4 84 54 124 T 116 21 15 25 53 35 65 5 85 55 125 U 117 22 16 26 54 36 66 6 86 56 126 V 118   | 68                               | 150                             | h           |
| 11     B     13     43     2B     53     +     75     4B     113     K     107       12     C     14     44     2C     54     ,     76     4C     114     L     108       13     D     15     45     2D     55     -     77     4D     115     M     109       14     E     16     46     2E     56     .     78     4E     116     N     110       15     F     17     47     2F     57     /     79     4F     117     O     111       16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     24     52     34     64     4     84     54     124   | 69                               | 151                             | i           |
| 12     C     14     44     2C     54     ,     76     4C     114     L     108       13     D     15     45     2D     55     -     77     4D     115     M     109       14     E     16     46     2E     56     .     78     4E     116     N     110       15     F     17     47     2F     57     /     79     4F     117     O     111       16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125  | 6A                               | 152                             | j           |
| 13     D     15     45     2D     55     -     77     4D     115     M     109       14     E     16     46     2E     56     .     78     4E     116     N     110       15     F     17     47     2F     57     /     79     4F     117     O     111       16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126   | 6B                               | 153                             | k           |
| 14     E     16     46     2E     56     .     78     4E     116     N     110       15     F     17     47     2F     57     /     79     4F     117     O     111       16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118   | 6C                               | 154                             | I           |
| 15     F     17     47     2F     57     /     79     4F     117     0     111       16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118  | 6D                               | 155                             | m           |
| 16     10     20     48     30     60     0     80     50     120     P     112       17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118   | 6E                               | 156                             | n           |
| 17     11     21     49     31     61     1     81     51     121     Q     113       18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118   | 6F                               | 157                             | 0           |
| 18     12     22     50     32     62     2     82     52     122     R     114       19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118   | 70                               | 160                             | р           |
| 19     13     23     51     33     63     3     83     53     123     S     115       20     14     24     52     34     64     4     84     54     124     T     116       21     15     25     53     35     65     5     85     55     125     U     117       22     16     26     54     36     66     6     86     56     126     V     118   | 71                               | 161                             | q           |
| 20 14 24 52 34 64 4 84 54 124 T 116<br>21 15 25 53 35 65 5 85 55 125 U 117<br>22 16 26 54 36 66 6 86 56 126 V 118   | 72                               | 162                             | r           |
| 21 15 25 53 35 65 5 85 55 125 U 117 22 16 26 54 36 66 6 86 56 126 V 118   | 73                               | 163                             | s           |
| 22 16 26 54 36 66 6 86 56 126 V 118   | 74                               | 164                             | t           |
|   | 75                               | 165                             | u           |
| 23 17 27   155 27 67 7   97 57 127 W   110  | 76                               | 166                             | v           |
| 25 1/ 2/   55 5/ 6/ /   6/ 5/ 12/ W   119   | 77                               | 167                             | w           |
| 24 18 30 56 38 70 8 88 58 130 X 120   | 78                               | 170                             | ×           |
| 25 19 31 57 39 71 9 89 59 131 Y 121   | 79                               | 171                             | У           |
| 26 1A 32   58 3A 72 :   90 5A 132 Z   122   | 7A                               | 172                             | z           |
| 27 1B 33   59 3B 73 ;   91 5B 133 [   123   | 7B                               | 173                             | {           |
| 28 1C 34   60 3C 74 <   92 5C 134 \   124   | 7C                               | 174                             |             |
| 29 1D 35 61 3D 75 = 93 5D 135 ] 125   | 7D                               | 175                             | }           |
| 30 1E 36 62 3E 76 > 94 5E 136 ^ 126   | 7E                               | 176                             | ~           |
| 31 1F 37   63 3F 77 ?   95 5F 137 _   127   | 7F                               | 177                             |             |

## sizeof로 크기 가져오기



• sizeof(변수) 또는 sizeof(타입)으로 해당하는 변수/타입이 차지하는 메모리의 크기를 바이트 단위로 가져올 수 있다.

- char a;
- int b;
- double c;
- int d[12];
- char e[10]="abc";

$$sizeof(char) = 1$$

$$sizeof(int) = 4$$

$$sizeof(a) = 1$$

$$sizeof(b) = 4$$

$$sizeof(c) = 8$$

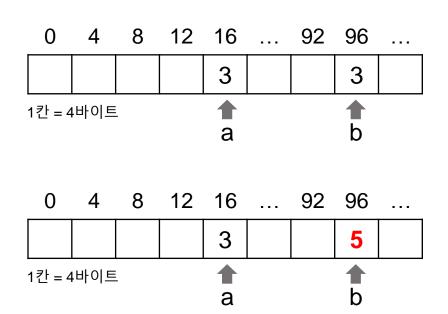
$$sizeof(d) = 48$$

$$sizeof(e) = 10$$

## 변수만 있으면 충분할까?



```
1 #include <stdio.h>
2
3 void set5(int b) {
4         b = 5;
5 }
6
7 int main(){
8         int a = 3;
9
10         set5(a);
11
12         printf("%d\n",a);
13
14         return 0;
15 }
```



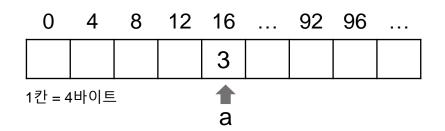
함수 호출이 일어날 때 인자로 넘겨진 변수들은 함수 내부에서 변수의 값이 복사된다. = 값에 의한 호출(call-by-value)

### 포인터의 필요성



• C 언어에서는 함수 인자로 넘겨진 값이 복사가 되고 함수 내부에서 새로 선언된 변수에 할당된다.

 따라서, 함수 내부에서 외부 변수를 수정하려면 값을 넘기는 것이 아니라 변수의 메모리 주소를 넘겨야 한다.

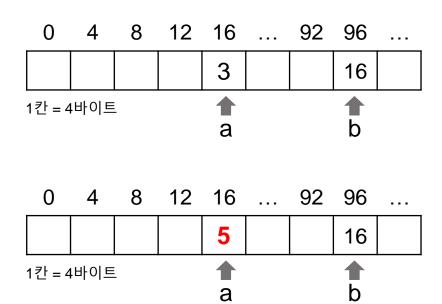


printf("%d", a); => 3 printf("%p", &a); => 16

### 포인터 예제



```
#include <stdio.h>
  void set5(int *b) {
       *b = 5;
   int main(){
       int a = 3;
       set5(&a);
11
       printf("%d\n",a);
13
14
       return 0;
```



함수 호출이 일어날 때 인자로 넘겨진 변수들은 함수 내부에서 **여전히** 변수의 값이 복사된다. 그러나, 복사되는 값이 메모리 주소 값이고 그 메모리 주소 값을 참조하여 a의 값을 변경하였다. = 참조에 의한 호출(call-by-reference)

### 포인터 예제



```
#include <stdio.h>
3 int main() {
     int x = 5;
     int *y;
     y = &x;
     printf("%d\n", x); // 5
     printf("%p\n", y); // 0x7ffd2dfcaef4
     *y = 1;
     printf("%d\n", x); // 1
     printf("%d\n", *y); // 1
     return 0;
```

### 포인터 변수의 크기



• int 형은 4바이트를 차지했다. 따라서,  $2^{32}$ 개의 서로 다른 정수를 표현할 수 있었다.

- int \*형은 몇 바이트를 차지할까?
  - = 컴퓨터에는 서로 다른 주소 값이 몇 개가 있을까?
- 32비트 운영 체제는 32비트로 주소 값을 표현
  - $2^{32}B=4*2^{30}B=4GB$
- 64비트 운영 체제는 64비트로 주소 값을 표현
  - $2^{64}B=16*2^{60}B=16EB$

설치된 메모리(RAM): 16.0GB

시스템 종류: 64비트 운영 체제, x64 기반 프로세서

### 포인터 변수의 크기



## 포인터 관련 연산자



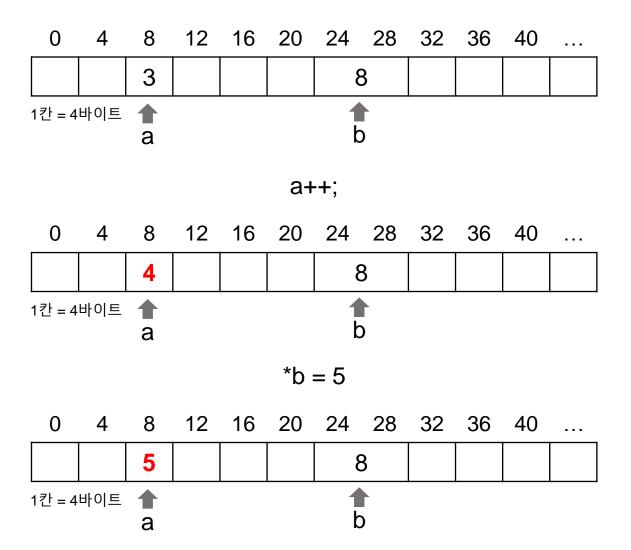
|                    | 0 | 4 | 8 | 12     | 16 | 20 | 24 | 28 | 32 | 36 | 40 |  |
|--------------------|---|---|---|--------|----|----|----|----|----|----|----|--|
|                    |   |   | 3 |        |    |    | 8  | 3  |    |    |    |  |
| 1칸 = 4바이트 <b>a</b> |   |   |   | 1<br>t | )  |    |    |    |    |    |    |  |

#### x 자리에 a또는 b가 들어갈 때

|             | x         | &x         | *x     | x+1                  |
|-------------|-----------|------------|--------|----------------------|
| int a = 3;  | 3         | 8 (a의 주소)  | 컴파일 에러 | 4                    |
| int *b = &a | 8 (a의 주소) | 24 (b의 주소) | 3      | 12 (8 + sizeof(int)) |

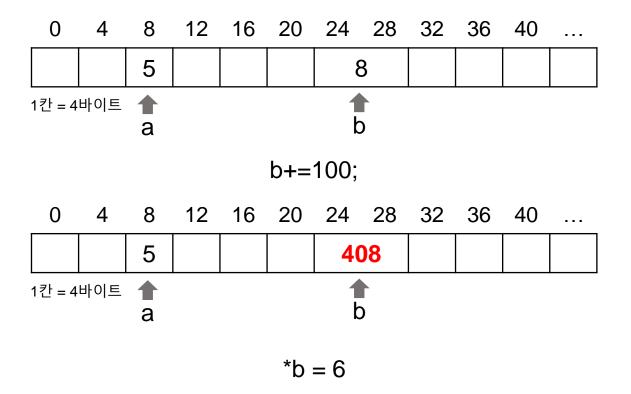
### 포인터 관련 연산자





### 포인터 관련 연산자





Segmentation Fault!

## 동적 할당

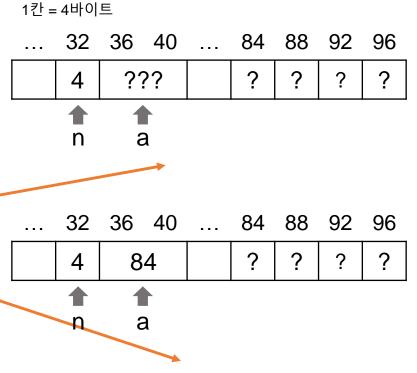


- int a[100];은 길이가 100인 int형 배열을 생성.
- 배열의 크기가 얼마나 필요한지 컴파일 타임에 알 수 없다면? = 런타임에서야 알 수 있다면?
  - int n;
  - scanf("%d", &n);
  - int a[n];
- malloc(<필요한 바이트 수>)를 이용하여 동적으로 메모리에 공간을 할당 받는다.

## 동적 할당 예제



```
1 #include <stdio.h>
2 #include <stdib.h>
3
4 int main() {
5    int n = 4;
6    int *a;
7
8    a = (int *)malloc(sizeof(int) * n);
9
10    for(int i = 0; i < n; i++) a[i] = i + 1;
11
12    for(int i = 0; i < n; i++) printf("%d\n" a[i]);
13
14    free(a);
15
16    return 0;
17 }</pre>
```



### 포인터 관련 주의사항



- malloc을 수행해서 돌아오는 결과 값은 새로 생성된 메모리 공간의 주소 값이다.
  - 반드시 초기화하고 사용할 것 (쓰레기 값)
  - 잃어버리지 말고 free를 실행하여 반환할 것
  - free로 반환한 이후에는 다시 접근하지 말 것
- 포인터를 이용하면 메모리의 값을 직접 변경할 수 있으므로 프로그램 외부 메모리에 접근하지 않도록 주의할 것.
- int \*a, b;
  - int \*a; int \*b; (X)
  - int \*a; int b; (O)

## 구조체 (Structure)



- 정수, 실수, 문자열 등의 단일 자료 형이 아니라 복합적인 자료를 포함하는 자료형
  - 구조체 안에도 구조체가 포함될 수 있음
- 학생: 이름(char), 입학 년도(int), 평점(float)
- 연관된 데이터를 묶어서 쉽게 관리할 수 있다.
  - 대입 또는 함수에서 리턴할 때

### 구조체 예제



```
| #include <stdio.h>
2 #include <string.h>
4 struct student {
     char name [20];
     int year;
                                               세미콜론에 주의
     float GPA;
10 int main() {
11
     struct student s;
                                               문자열은 바로 대입할 수
                                               없으므로 strcpy함수를
     strcpy(s.name, "김 알고 리 증 ");
     s.year = 2019;
                                               사용하여 복사
     s.GPA = 3.0;
     printf("%s %d %.3f\n", s.name, s.year, s.GPA);
                                                .연산자를 이용하여 멤버
     return 0;
                                                변수에 접근
```

## typedef



- 항상 struct \_student로 타입을 적기는 번거롭다.
- typedef struct \_student student; 로 별칭을 붙이자.

```
#include <stdio.h>
2 #include <string.h>
4 struct _student {
     char name [20];
     int year;
     float GPA;
8 };
.0 typedef struct student student;
.2 int main() {
      student s;
      strcpy(s.name, "김 알 고 리 즘 ");
      s.year = 2019;
     s.GPA = 3.0;
      printf("%s %d %.3f\n", s.name, s.year, s.GPA);
      return 0;
```

struct \_student에 student라는 별칭을 붙임

간결하게 사용

## typedef



• 구조체 선언과 typedef를 한번에 끝내기

```
1 #include <stdio.h>
2 #include <string.h>
3
4 typedef struct _student {
5     char name[20];
6     int year;
7     float GPA;
8 } student;
```

## 구조체, 전처리기, typedef 예제

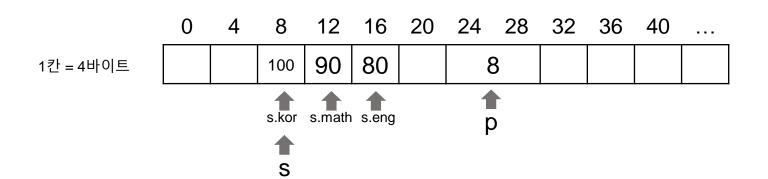


```
1 // '참'을 표현하기 위한 매크로 정의
#define TRUE
                   // '거짓'을 표현하기 위한 매크로 정의
#define FALSE
              0
#define LIST LEN
              100
typedef int LData;
typedef struct ArrayList // 배열기반 리스트를 정의한 구조체
  LData arr[LIST_LEN]; // 리스트의 저장소인 배열
  int numOfData; // 저장된 데이터의 수
  int curPosition; // 데이터 참조위치를 기록
} ArrayList;
typedef ArrayList List;
```

## 포인터 + 구조체



```
1 #include <stdio.h>
2
3 typedef struct _score {
4    int kor;
5    int math;
6    int eng;
7 } score;
8
9 int main() {
10    score s = {100, 90, 80};
11    score *p = &s;
12
13    printf("%d\n", (*p).kor); // 100
14    printf("%d\n", p->kor); // 100
15
16    return 0;
17 }
```



### 포인터 + 구조체 예제

1

#define TRUE

// '참'을 표현하기 위한 매크로 정의



```
#define FALSE
                   // '거짓'을 표현하기 위한 매크로 정의
#define LIST LEN
             100
typedef int LData;
typedef struct __ArrayList
                     // 배열기반 리스트를 정의한 구조체
  LData arr[LIST LEN];
                     // 리스트의 저장소인 배열
  int numOfData;
                   // 저장된 데이터의 수
  int curPosition;
                      // 데이터 참조위치를 기록
} ArrayList;
                      void LInsert(List * plist, LData data)
typedef ArrayList List;
                          if(plist->numOfData > LIST_LEN) // 더 이상 저장할 공간이 없다면
                              puts("저장이 불가능합니다.");
                              return;
                          plist->arr[plist->numOfData] = data; // 데이터 저장
                          (plist->numOfData)++; // 저장된 데이터의 수 증가
```

## 요약



- 포인터를 이용하여 변수의 주소 값을 가져와 접근할 수 있다.
  - call-by-value vs call-by-reference
- malloc(bytes)으로 원하는 크기의 메모리를 동적으로 할당할 수 있다.
  - 이 때, 결과 값은 할당된 메모리의 주소
  - 잘 가지고 있다가 free()를 통해 해제해 주어야 한다.
- 구조체를 이용하여 기본 자료 형을 묶은 복합 자료형을 만들 수 있다.

## 요약



- &, 앰퍼샌드(ampersand), &(변수)로 사용
  - 변수의 주소 값을 가져옴 (&a)
- \*, 스타(star), \*(포인터 변수)로 사용
  - 포인터 변수에 담긴 주소에 접근 (\*p)
- ., 닷(dot), (구조체 변수).(멤버 변수)로 사용
  - 구조체 변수에서 해당 멤버 변수에 접근 (student.name)
- ->, 애로우(arrow), (구조체 포인터 변수)->(멤버 변수)로 사용
  - 구조체 포인터 변수가 가리키는 구조체에 가서 멤버 변수에 접근 (s->name)