

ZERONE [0&1]

C Language Basic

영과일 C언어 기초반 #3 - 문자열

0&1

컴퓨터학부 2022011812

송진우

1. Remind

CODE

```
int i = 0;

while (i < 5) {
    printf("%d\n", i);
    i += 1;
}
```

- i라는 변수가 5보다 작을경우 (0,1,2,3,4)
 - printf로 i의 값을 출력하고
 - i의 값에다가 1을 더하는것을
- 반복하는 코드

OUTPUT

```
0
1
2
3
4
```

- 따라서, 결과는 0, 1, 2, 3, 4로 i가 5가 된 순간부터 반복은 이루어지지 않습니다.

CODE

```
int i;  
for (i = 0; i < 10; i++) {  
    printf("%d\n", i);  
}
```

- i=0을 실행시키기
- i < 10인지 체크하고
- 실행문 부분에 적힌 명령을 실행
- i++을 실행
- i가 0으로 설정되고
 - i가 10보다 작다면
 - printf()함수로 i의 값을 출력하고
 - i에다 1을 추가함
 - i가 10을 초과하는 순간 for문 종료.

OUTPUT

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

- 따라서, i가 10인 결과는 나오지 않습니다.

```
for (i = 0; i < 5; i++) {  
    if (i == 2) break;  
    printf("%d\\n", i);  
}
```

```
0  
1
```

- break는 말 그대로, 반복문을 종료하라는 명령
- 옆 코드에서는 i가 2가 될 때 반복문을 강제로 종료
 - 따라서, printf도 실행되지 않고 프로그램이 종료됨.

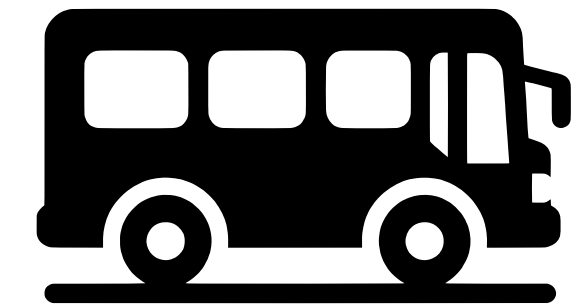
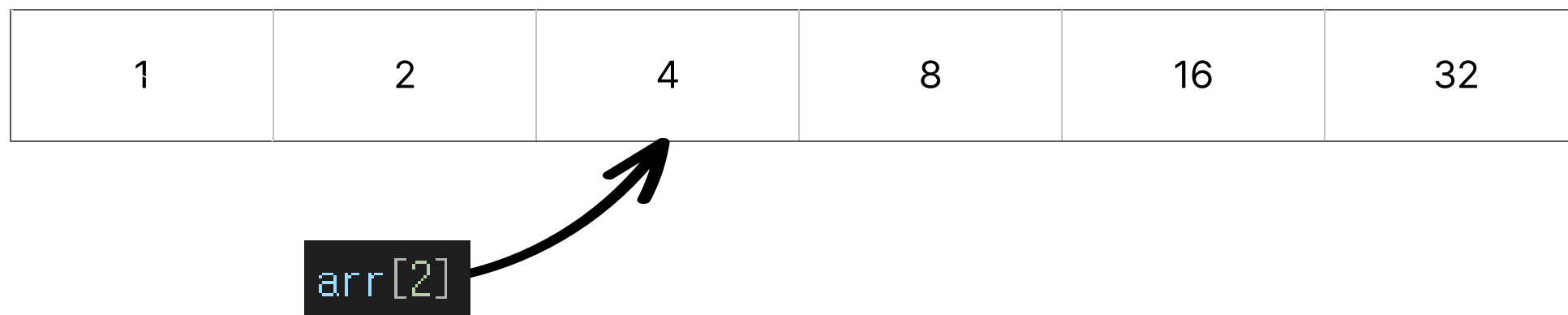
```
for (i = 0; i < 5; i++) {  
    if (i == 2) continue;  
    printf("%d\\n", i);  
}
```

```
0  
1  
3  
4
```

- continue는 말 그대로, 계속해서 반복문을 실행시키라는 명령
 - 단, continue 명령을 받은 순간 진행되던 실행문은 skip!
 - 즉, i가 2가 된 순간 printf명령은 무시하고 다시 증감식을 받아 i는 3이 되고, 3이 출력된 것.

배열 [Array]

- 배열은, 다음과 같은 성질을 가집니다.
 - 배열은 한줄로 나열되어있는 데이터들이라고 생각해도 됩니다.
 - 나열되어있는 데이터들은 순서가 지정되어있고, 각 순서에 저장되어있는 데이터를 '원소' 라고 합니다.
 - 각 원소는 '0번' 부터 'N-1번' 까지 번호가 있고, 번호로 각 원소에 접근할 수 있습니다.



```
int arr[1001];
```

자료형 변수이름 [배열크기] = {배열 원소 목록}

배열 선언, 정의 (declaration / definition)

변수이름 [번호]

원소 접근 (Access)

2. 문자열

문자와 문자열

5 브론즈 V

● 27866: 문자와 문자열

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	1024 MB	68116	45559	41808	68.083%

문제

단어 S 와 정수 i 가 주어졌을 때, S 의 i 번째 글자를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 영어 소문자와 대문자로만 이루어진 단어 S 가 주어진다. 단어의 길이는 최대 1000이다.

둘째 줄에 정수 i 가 주어진다. ($1 \leq i \leq |S|$)

출력

S 의 i 번째 글자를 출력한다.

문제에 차근차근 접근

- 우선, 저 문자들을 다 받아내기 위해서 배열을 한번 만들어 보겠습니다.

```
char c[20];
```

```
int i = 0;
while(1){
    scanf("%c", &c[i]);
    i++;
}
```

- 다음으로, 저 여러개의 문자들을 받기 위해서 어떻게 할지 고민을 하다가...
 - while 반복문을 통해서 다 받아와 보는것으로 결정해보겠습니다.

- 그런데 저렇게 하면, 끊임없이 글자들을 읽어오고 반복문이 안끝날텐데... 어떻게 끝내야하지?
 - 한번 저번에 배웠던 break를 활용해서 어떻게 해볼까요.

```
int i = 0;
while(1){
    scanf("%c", &c[i]);
    if(c[i] == '\n') break;
    i++;
}
```

문제에 차근차근 접근

- 그런다음, 숫자를 받아오는 부분을 만들어주고...

```
int num;  
scanf("%d", &num);
```

```
printf("%c", c[num-1]);
```

- 출력해주는 부분도 만들어주고 실행해보면?

- 코드가 깔끔하게 동작되는 것 같네요.

```
char c[1002];  
  
int i = 0;  
while(1){  
    scanf("%c", &c[i]);  
    if(c[i] == '\n') break;  
    i++;  
}  
  
int num;  
scanf("%d", &num);  
  
printf("%c", c[num-1]);
```

- 코드가 잘 동작하는 것 같긴 한데...
 - 저 while 반복문 부분을 아예 빼버릴 수 있다면?
 - 입력이 더 복잡하게 들어온다면?

Sprout

1
5

=== Code Execution Successful ===

문자열

- 간단하게 문자열은 "문자"들의 "열" 으로 문자들을 여러개 나열해둔 문자의 집합입니다.
 - A → 문자 / ABCD → 문자열


```
char c[1002];  
  
c[0] = 'A';  
c[1] = 'B';  
c[2] = 'C';  
c[3] = 'D';  
  
for(int i = 0; i < 4; i++) printf("[%c] ", c[i]);
```



[A] [B] [C] [D]

- 근데, 그냥 문자를 배열에 넣은 것 뿐인데, 왜 특별하게 문자열이라고 명할까?
 - 문자열을 다루는 특수한 특징들이 여러개 있습니다.

```
char c[1002];  
  
c[0] = 'A';  
c[1] = 'B';  
c[2] = 'C';  
c[3] = 'D';  
  
printf("%s", c);
```




ABCD·
=== Code Execution

문자열

- 당연히, 입력도 간편하게 받을 수 있겠죠?

```
int main() {  
  
    char c[1002];  
  
    scanf("%s", c);  
  
    printf("%s", c);  
}
```

- scanf할 때, 문자열에서
&안쓰는것에 주의



```
ABCD  
ABCD  
  
=== Code Execution Successful ===
```

- 입력과 출력을 할 때, 아까 앞부분에서 이상한 모양을 보았습니다.

```
ABCD·
```

```
=== Code Execution
```

- 과연, 저 빨간 점은 왜 나오고, 문제가 생길까요?
 - 이는 문자열의 저장방식이, 단순한 문자의 배열과는 다르기 때문!

문자열

- 앞선 코드를 아래와 같이 바꾸면, 빨간 점 없이 나오게 됩니다.

```
char c[1002];

c[0] = 'A';
c[1] = 'B';
c[2] = 'C';
c[3] = 'D';
c[4] = '\0';

printf("%s", c);
```

- 어떤것이 달라졌길래 점이 나오지 않을까요?

ABCD

=== Code Execution Successful ===

- HINT: 이전 코드

```
char c[1002];

c[0] = 'A';
c[1] = 'B';
c[2] = 'C';
c[3] = 'D';

printf("%s", c);
```

문자열

- 아래 코드는, 이전 코드와는 다르게 마지막 부분에 '\0' 이 들어갑니다.

```
char c[1002];

c[0] = 'A';
c[1] = 'B';
c[2] = 'C';
c[3] = 'D';
c[4] = '\0';

printf("%s", c);
```

- 마찬가지로, 문자열로 입력을 받는다면, 입력을 받고 난 후 마지막에 '\0'을 넣어주게 됩니다. **ABCD\0**

```
int main() {

    char c[1002];

    scanf("%s", c);

    printf("%s", c);
}
```

- 그래서, 문자열은 4개의 글자를 넣고 싶으면 최소한 5개의 배열 공간을 잡아주어야 합니다.

- 그렇다면 한번, "I LOVE YOU" 라는 문자열을 저장받고 출력해볼까요?

문자열

- 아마, 원하대로 출력이 안된분들이 많으실 겁니다.
 - 아래 코드를 사용해서 한다면 아래와 같은 결과가 나옵니다.

```
char c[1002];

scanf("%s", c);
printf("%s", c);
```

```
I LOVE YOU
I
=== Code Execution Successful ===
```

- scanf 함수는, 공백을 기준으로 입력을 종료 받기에, 여러번 입력, 혹은 다른 함수를 사용합니다.
 - gets 함수를 사용하면, 줄바꿈을 기준으로 입력받을 수 있습니다.

- fgets 함수는
 - 변수명, 변수 크기, stdin을 넣어주어야 합니다.

```
char c[1002];

fgets(c, sizeof(c), stdin);
printf("%s", c);
```

```
I LOVE YOU
I LOVE YOU

=== Code Execution Successful ===
```

문자열

- 문자열은 또, 여러가지 함수들을 제공합니다.
 - 해당함수들은, 헤더파일 `#include <string.h>` 를 추가해야합니다.

```
#include <stdio.h>
#include <string.h>
```

함수명	기능	사용법 (char A[100] = "HELLO", char B[100] = "HELLO")
strlen()	문자열의 길이 반환	strlen(A) → 5
strcmp()	문자열1이 문자열2보다 사전적으로 앞에있으면 음수, 같으면 0, 뒤에있으면 양수를 반환	strcmp(A, B) = 0
strcpy()	문자열 복사	char C[20]; strcpy(C, A); printf("%s", C) → "HELLO"
strcat()	문자열1에 문자열2 합침	strcat(A, B); printf("%s", A) → "HELLOHELLO"
strstr()	문자열1에 문자열2에 어떻게 포함되어 있는지 반환	strstr("ABCD", "C") → CD (*이해를 돕기위한 설명입니다.)

3. 문제 풀이

- 2754: 학점계산
- 2744: 대소문자 바꾸기
- 11720: 숫자의 합

	char	int	long long	float	double	string
서식 지정자	%c	%d	%lld	%f	%lf	%s
	NOT	![A]	+	더하기	/	나누기
	AND	[A] && [B]	-	빼기	%	나머지
	OR	[A] [B]	*	곱하기	A != B	다르다

함수명	기능	사용법 (char A[100] = "HELLO", char B[100] = "HELLO")
strlen()	문자열의 길이 반환	strlen(A) → 5
strcmp()	문자열1이 문자열2보다 사전적으로 앞에있으면 음수, 같으면 0, 뒤에있으면 양수를 반환	strcmp(A, B) = 0
strcpy()	문자열 복사	char C[20]; strcpy(C, A); printf("%s", C) → "HELLO"
strcat()	문자열1에 문자열2 합침	strcat(A, B); printf("%s", A) → "HELLOHELLO"
strstr()	문자열1에 문자열2에 어떻게 포함되어 있는지 반환	strstr("ABCD", "C") → CD (*이해를 돕기위한 설명입니다.)

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main() {
5
6     char G[5];
7     scanf("%s", G);
8
9     double grade = 0;
10    if(strcmp(G, "A+") == 0) grade = 4.3;
11    else if(strcmp(G, "A0") == 0) grade = 4.0;
12    else if(strcmp(G, "A-") == 0) grade = 3.7;
13    else if(strcmp(G, "B+") == 0) grade = 3.3;
14    else if(strcmp(G, "B0") == 0) grade = 3.0;
15    else if(strcmp(G, "B-") == 0) grade = 2.7;
16    else if(strcmp(G, "C+") == 0) grade = 2.3;
17    else if(strcmp(G, "C0") == 0) grade = 2.0;
18    else if(strcmp(G, "C-") == 0) grade = 1.7;
19    else if(strcmp(G, "D+") == 0) grade = 1.3;
20    else if(strcmp(G, "D0") == 0) grade = 1.0;
21    else if(strcmp(G, "D-") == 0) grade = 0.7;
22    else if(strcmp(G, "F") == 0) grade = 0.0;
23
24    printf("%.1lf", grade);
25
26    return 0;
27 }
```

- 사용자로부터 우선 N과 X를 받아와서
- N의 크기는 최대 10000이므로, A 배열의 크기의 크기도 10001으로 설정해둡니다.
 - A[N+1] 과 같이 변수가 들어간 선언은 허용하지 않는 컴파일러도 있습니다.
- 다음으로, N번 루프를 돌며, A에 값들을 저장합니다.
- 그런 뒤, X와 A의 i번째 원소를 비교하면서 i를 N-1까지 이동시키며 반복하여 조건에 따라 A의 i번째 원소를 출력해줍니다.


```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5
6      char string[101];
7      scanf("%s", string);
8
9      for(int i = 0; i < strlen(string); i++){
10         if(string[i] <= 90) string[i] = string[i]+32;
11         else string[i] = string[i]-32;
12     }
13
14     printf("%s", string);
15
16     return 0;
17 }
```

- 최대 100자의 문자열이 들어오니, 문자열 배열의 사이즈를 101로만 들어줍니다.
- 그런 뒤, string에 입력을 받아오고 받아온 문자열을 반복문으로 하나씩 접근하며 조사합니다.
 - 만약에, 아스키코드 기준으로...
 - string[i] 가 90 이하다. 즉, 대문자라면 32를 더해서 소문자로 만듭니다.
 - 그게 아니면 소문자라는 뜻이니, 32를 빼서 대문자로 만듭니다
- 그렇게 대소문자를 바꾼 뒤, 출력을 해서 문제 풀이를 종료합니다.

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5
6      int N;
7      scanf("%d", &N);
8
9      char string[101];
10     scanf("%s", string);
11
12     int answer = 0;
13     for(int i = 0; i < N; i++){
14         int tmp = string[i] - '0';
15         answer += tmp;
16     }
17
18     printf("%d", answer);
19 }
```

- 우선, N개의 숫자가 입력으로 주어지니, N을 우선적으로 읽어들이입니다.
- 그런 뒤, 문자열 변수를 하나 선언하고 공백없이 주어진 숫자들을 문자의 형태로 받아옵니다.
- 그리고 난 뒤, 모든 수를 더해줄 변수 answer를 선언합니다.
- 다음으로, string의 0번 index부터 N-1번 index까지 반복을 돌리면서 하나씩 더해줍니다.
 - 이 때, 문자열에서 숫자의 값은 숫자로 변환했을 때, 48+숫자 입니다. 따라서 숫자에서 48을 뺀 값 만큼 tmp에 저장하고, 해당 tmp 값을 answer에 더해줍니다.
- 마지막으로 answer를 출력하여 답변을 출력합니다.