

Coding Review (I2P 2019)

Standard Inputs of C

Review of common inputs: `scanf`, `getchar`, `gets` (`fgets`).

Basic formats

1. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int a, b;
    scanf("%d%d", a, b);
    printf("%d %d", a, b);
    return 0;
}
```

Input:

12 34

Answer: (Write your answer here!)

2. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int a, b;
    scanf("%1d%2d", &a, &b);
    printf("%d %d", a, b);
    return 0;
}
```

Input:

1234 5678

Answer: (Write your answer here!)

3. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int a, b;
    scanf("(%d,%d)", &a, &b);
    printf("%d %d", a, b);
    return 0;
}
```

Input:

(12,34)

Answer: (Write your answer here!) What if the input becomes:

(12, 34)

Answer: (Write your answer here!)

4. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    float f;
    double d;
    scanf("%f%f", &f, &d);
    printf("%f %f", f, d);
    return 0;
}
```

Answer: (Write your answer here!)

Usages of `scanf("%c", ...)` and `getchar`

1. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
```

```

int i;
char a[6];
for (i = 0; i < 6; i++)
    scanf("%c", &a[i]);
for (i = 0; i < 6; i++)
    printf("%d ", (int)a[i]);
return 0;
}

```

Input:

```

A B
C

```

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!)

2. What is the result of the following code?

```

#include <stdio.h>

int main(void) {
    int i;
    char a[6];
    for (i = 0; i < 6; i++)
        scanf(" %c", &a[i]);
    for (i = 0; i < 6; i++)
        printf("%d ", (int)a[i]);
    return 0;
}

```

Input:

```

AB C
DEF G

```

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!)

3. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int i;
    char a[6];
    for (i = 0; i < 6; i++)
        a[i] = getchar();
    for (i = 0; i < 6; i++)
        printf("%d ", (int)a[i]);
    return 0;
}
```

Input:

```
AB C
DEF G
```

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!)

Usages of `scanf("%s", ...)`

1. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int i;
    char a[5] = {-1, -1, -1, -1, -1};
    scanf("%s", a);
    for (i = 0; i < 5; i++)
        printf("%d ", (int)a[i]);
    return 0;
}
```

Input:

```
ABC DE
FG
```

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!)

2. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int i;
    char a[3];
    scanf("%s", a);
    for (i = 0; i < 3; i++)
        printf("%d ", (int)a[i]);
    return 0;
}
```

Input:

ABC

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!)

gets vs. fgets

1. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int i;
    char s[6] = {-1, -1, -1, -1, -1, -1};
    fgets(s, 6, stdin);
    for (i = 0; i < 6; i++)
        printf("%d ", s[i]);
    return 0;
}
```

Input:

ABCDEF

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!) What if the input becomes:

ABC

Answer: (Write your answer here!)

2. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int i;
    char s[6] = {-1, -1, -1, -1, -1, -1};
    gets(s);
    for (i = 0; i < 6; i++)
        printf("%d ", s[i]);
    return 0;
}
```

Input:

ABC

Hint: 'A' is 65, ' ' is 32, '\n' is 10.

Answer: (Write your answer here!) What if the input becomes:

ABCDEF

Answer: (Write your answer here!)

scanf's return value

1. What is the result of the following code?

```
#include <stdio.h>

int main(void) {
    int a, b, ret;
    a = b = -1;
    ret = scanf("%d%d", &a, &b);
}
```

```
printf("%d %d %d", ret, a, b);  
return 0;  
}
```

Input:

```
100 abc 1000
```

Answer: (Write your answer here!)

2. What is the result of the following code?

```
#include <stdio.h>  
  
int main(void) {  
    int a, b, ret;  
    a = b = -1;  
    ret = scanf("(%d,%d)", &a, &b);  
    printf("%d %d %d", ret, a, b);  
    return 0;  
}
```

Input:

```
( 1 , 2 )
```

Answer: (Write your answer here!)

End of File EOF

EOF is defined to be `-1` in most cases. When the input ends, using `scanf`, `getchar` result in EOF.

For `gets` and `fgets`, they returns `NULL`, which is `0` in most cases.

To simulate EOF, press `Ctrl+Z` and `Enter` on Windows; press `Ctrl+D` and `Enter` on MacOS or Linux.

For some common usages:

```
while (scanf("%d", &x) != EOF) {  
    // Do something.  
}  
while (scanf("%d%d", &a, &b) == 2) {  
    // Do something.  
}
```

```

while ((c = getchar()) != EOF) {
    // Do something.
}
// Change 'MAX_STRLEN' to a constant (max string length including null-terminator)
while (fgets(s, MAX_STRLEN, stdin) != NULL) {
    // Do something.
}

```

Case Study

There's often multiple ways to read a input.

If we want to read in each of the digits of a 5-digit number:

12345

And store them into 5 variables `a`, `b`, `c`, `d`, `e` respectively.

1. Math

```

int n;
scanf("%d", &n);
a = n / 10000 % 10;
b = n / 1000 % 10;
c = n / 100 % 10;
d = n / 10 % 10;
e = n / 1 % 10;

```

2. Scan 1-digit at a time

```

scanf("%1d%1d%1d%1d%1d", &a, &b, &c, &d, &e);

```

3. Scan by char

```

a = getchar() - '0';
b = getchar() - '0';
c = getchar() - '0';
d = getchar() - '0';
e = getchar() - '0';

```

4. Scan by string

```


```



```
char str[6];  
scanf("%s", &str);  
a = str[0] - '0';  
b = str[1] - '0';  
c = str[2] - '0';  
d = str[3] - '0';  
e = str[4] - '0';
```

5. And more...

For the next assignment, we'll review standard outputs of C.

If there's any typo, please discuss on iLMS or email j3soon@gapp.nthu.edu.tw, I appreciate your help.