

Programming Basics

Data Structure

Im Y. Jung



Problem

- Implement a C Program satisfying the following conditions:
 - Input : name(first last), age, sex
 - Output : name(last first), age, sex
 - Use loop and get inputs until “stop” is entered as name
 - Use pointer
 - Use the program structure given

```
int main()
{
    char name[50];
    int age;
    char sex;
    char *namep;

    /*
    Your code ?
    */

    return 0;
}
```

- Execution

Provide your personal information:

Name : sungsoo kim

Age : 21

Sex (M/F) : M

Your name is kim sungsoo, your age is 21, your sex is M.

Provide your personal information:

Name : mijin lee

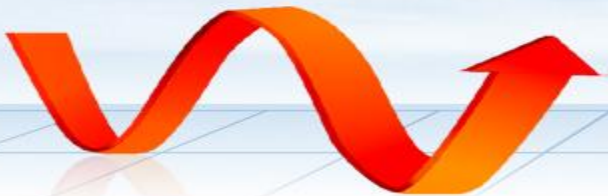
Age : 20

Sex (M/F) : F

Your name is lee mijin, your age is 20, your sex is F.

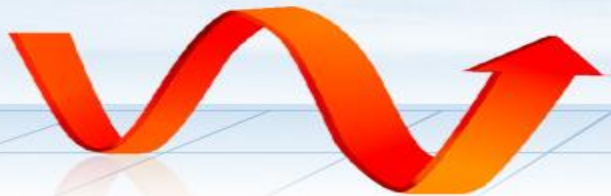
Provide your personal information:

Name : stop



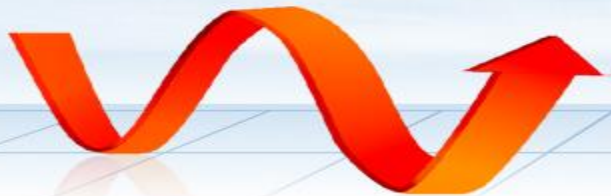
What to know in C programming for Data Structure

- Data/storage Design
 - **Data In/Out**
 - Keyboard/Screen, File/File
 - **Data Type**
 - Primitive
 - long/int/short
 - double/float
 - char
 - Compound
 - Array
 - » One/two/three dimension
 - structure
 - Pointer
- Program Flow
 - **Algorithm**
 - **Program Structure**



Example – Problem Analysis (1)

- Input : name(first last), age, sex
- Output : name(last first), age, sex
- Requirements :
 - If “stop” is entered to name, your program will end.
 - If not, your program will go on.
 - You should use the given program structure.
- What to do
 - **In/Out Design**
 - Keyboard in/Screen out
 - **Get the data from keyboard and store it, then print out the data at screen**
- What to use
 - **Data/storage Design**
 - Name : string
 - Age : int
 - Sex : char



Example – Problem Analysis (2)

- The given structure

```
int main()
{
    char name[50];
    int age;
    char sex;
    char *namep;

    //programming going on
    // insert your code here

    return 0;
}
```

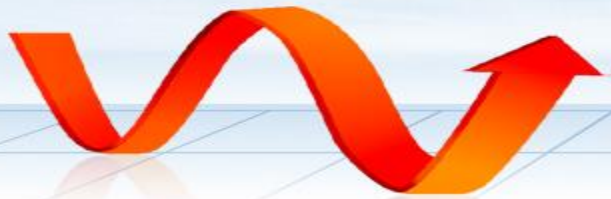
- How to do

- **Program structure**

- Given : one function, main()

- **Algorithm**

- Get name, age, sex continuously until “stop” is entered
 - **Endless loop**, Exit the loop when “stop” is entered
 - Get the various typed data entered
 - » **scanf, gets, getchar**
 - Print out the data entered
 - » When print out, **change the position of first name and last name using pointer**
 - » **printf, puts, putchar**



Example – Data/storage Design

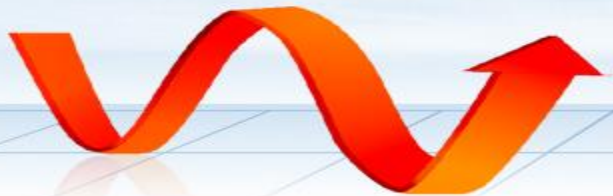
char name[50];

int age;

char sex;

char *namep;

- Name : string
 - Max 49 chars, 50 bytes
- Age : int
 - 4 bytes
- Sex : char
 - 1 byte
- Pointer to char
 - 4 byte (32 bit machine)
 - Store the address of char



Example – Program Flow (1)

```
while(1) {  
    if()  
        break;  
}
```

```
for(;;) {  
    if()  
        break;  
}
```

```
do {  
    if()  
        break;  
} while(1);
```

```
while(1) {  
    Get name from keyboard;  
    if("stop" is entered)  
        break;  
    Get age from keyboard;  
    Get sex from keyboard;  
  
    Analyze name by first name and last name;  
  
    Print out last name, first name, age, sex to screen;  
}
```

- Get name, age, sex continuously until “stop” is entered
 - Endless loop, Exit the loop when “stop” is entered
 - Analyze name by first and last
 - Print out the data entered



Example – Program Flow (2)

- Get name from keyboard;
 - Get age from keyboard;
 - Get sex from keyboard;
 - Print out last name, first name, age, sex to screen;
- Get the various typed data entered
 - **scanf, gets, getchar**
 - Print out the data entered
 - **printf, puts, putchar**

C API Reference

<https://en.cppreference.com/w/c/io/fscanf>

<https://en.cppreference.com/w/c/io/fprintf>

<https://en.cppreference.com/w/c/io/getchar>



scanf(1)

- Notice
 - **Space in scanf format**
 - "skip all (0 or more) whitespace" from the current position on.
 - **Since most scanf format specifier will already skip all whitespace before attempting to read anything, space is not used in scanf format most of the time.**
 - The specifiers that do not ignore whitespace are a [, c, C, and n.
 - **To use fflush() ?**
 - C Standard specifies the behavior of fflush(stdin) is undefined.

```
fflush(stdin);
scanf("%c" &c);
```

• Example (char)

```
int main()
{
    char ch;
    scanf("%c", &ch);
    printf("%c", ch);
    scanf("%c", &ch);
    printf("%c", ch);

    return 0;
}
```

```
int main()
{
    char ch;
    scanf("%c", &ch);
    printf("%d", ch);
    printf("\n");
    scanf("%c", &ch);
    printf("%d", ch);

    return 0;
}
```

```
int main()
{
    char ch;
    scanf(" %c", &ch);
    printf("%d", ch);
    printf("\n");
    scanf(" %c", &ch);
    printf("%d", ch);

    return 0;
}
```

• Example (int)

```
int main()
{
    int v1, v2, v3;
    scanf(" %d %d %d", &v1, &v2, &v3);
    printf("Out1: %d %d %d", v1, v2, v3);
    printf("\n");
    scanf("%d%d%d", &v1, &v2, &v3);
    printf("Out2: %d %d %d", v1, v2, v3);

    return 0;
}
```

scanf (2)

- Notice
 - %s : get a string until not a space or return
 - %[pattern]s : take input while it is a char in pattern
 - %[^pattern]s : take input while it is not a char in pattern
 - scanf ("%[^\\n]s", name);
 - cf. scanf (" %[^\\n]s", name);
 - If there is a char left such as '\\n' in the previous input, it should be ignored.

```
int main()
{
    char str[30];

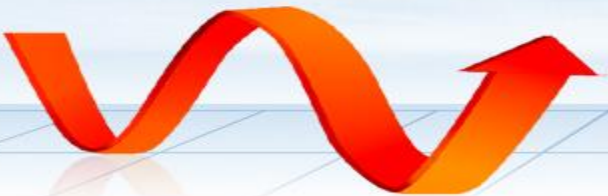
    printf("Enter a string: ");
    scanf("%[abc]s", str);

    printf("The string is: %s\\n",str);
    return 0;
}
```

```
int main()
{
    char str[30];

    printf("Enter a string: ");
    scanf("%[^\\abc]s", str);

    printf("The string is: %s\\n",str);
    return 0;
}
```



scanf(3)

- Notice
 - **scanf (" %[^\n]s", name);**
 - If there is a char left such as '\n' in the previous input, it should be ignored.

```
#include <stdio.h>

int main()
{
    int age;
    char name[30];
    char temp;

    printf("Enter age: ");
    scanf("%d",&age);
    printf("Enter name: ");
    scanf("%c",&temp); // temp statement to clear buffer
    scanf("%[^\n]",name);

    printf("Name is: %s, age is: %d\n",name,age);
    return 0;
}
```

```
int main () {
    int age;
    char name[30];

    printf("Enter age: ");
    scanf("%d",&age);
    printf("Enter name: ");
    scanf("%[^\n]",name);

    printf("Name is: %d, age is: %d\n",name[0],age);
    return 0;
}
```

```
int main()
{
    int age;
    char name[30];

    printf("Enter age: ");
    scanf("%d",&age);
    printf("Enter name: ");
    scanf(" %[^\n]s",name);

    printf("Name is: %s, age is: %d\n",name,age);
    return 0;
}
```

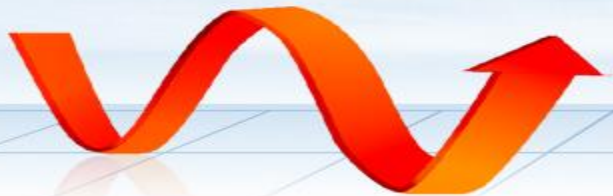
fgets

- reads the complete string with spaces and also add a new line character after the string input.

```
int main()
{
    int age;
    char name[30];
    char temp;

    printf("Enter age: ");
    scanf("%d",&age);
    printf("Enter name: ");
    scanf("%c",&temp); // temp statement to clear buffer
    fgets(name,30,stdin);

    printf("Name is: %s, age is: %d\n",name,age);
    return 0;
}
```



Example – Program Flow (3)

- **if("stop" is entered)**

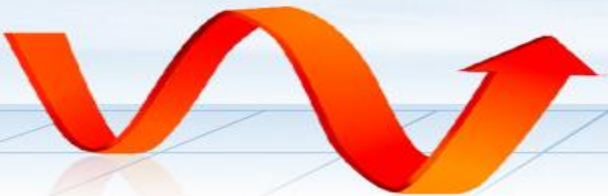
```
int main () {
    char str1[15];
    char str2[15];
    int ret;

    strcpy(str1, "abcdef");
    strcpy(str2, "ABCDEF");
    ret = strcmp(str1, str2);

    if(ret < 0) {
        printf("str1 is less than str2");
    } else if(ret > 0) {
        printf("str2 is less than str1");
    } else {
        printf("str1 is equal to str2");
    }

    return(0);
}
```

- String comparison
 - **strcmp** : #include <string.h>
- C API Reference
 - https://www.tutorialspoint.com/c_standard_library/c_function_strcmp.htm



Example – Program Flow (4)

- Analyze name by first name and last name;

- String tokenize
 - strtok** : **#include <string.h>**

C API Reference

https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm

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

int main () {
    char str[80] = "- This, a sample string.";
    const char s[5] = " ";
    char *token;

    /* get the first token */
    token = strtok(str, s);
    /* walk through other tokens */
    while( token != NULL ) {
        printf( "\ntoken : %s\n", token );
        printf( "str : %s\n", str );

        token = strtok(NULL, s);
    }

    return(0);
}
```

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

int main () {
    char str[80] = "- This, a sample string.";
    const char s[5] = " ,.-|";
    char *token;

    /* get the first token */
    token = strtok(str, s);
    /* walk through other tokens */
    while( token != NULL ) {
        printf( "\ntoken : %s\n", token );
        printf( "str : %s\n", str );

        token = strtok(NULL, s);
    }

    return(0);
}
```



Samples

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

int main () {
    char name[50];
    int age;
    char sex;
    char* namep;
    int i;

    while (1)
    {
        printf("Provide your personal information :\n");
        printf("NAME : ");
        scanf(" %[^\\n]s", name);

        if (!strcmp(name, "stop"))
            break;

        printf("AGE : ");
        scanf(" %d", &age);
        printf("SEX(M/F) : ");
        scanf(" %c", &sex);
        for(i=0, namep = name; name[i] != ' '; i++)
            namep++;
        namep++;
        name[i] = '\\0';

        printf("Your name is %s %s", namep, name);
        printf(" your age is %d, your sex is %c\\n", age, sex);
    }
    return 0;
}
```

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

int main () {
    char name[50];
    int age;
    char sex;
    char* namep;
    int i;

    while (1)
    {
        printf("Provide your personal information :\n");
        printf("NAME : ");
        scanf("%s", name);

        if (!strcmp(name, "stop"))
            break;

        scanf("%s", name+strlen(name)+1);
        name[strlen(name)] = ' ';

        printf("AGE : ");
        scanf(" %d", &age);
        printf("SEX(M/F) : ");
        scanf(" %c", &sex);
        for(i=0, namep = name; name[i] != ' '; i++)
            namep++;
        namep++;
        name[i] = '\\0';

        printf("Your name is %s %s", namep, name);
        printf(" your age is %d, your sex is %c\\n", age, sex);
    }
    return 0;
}
```

Samples

```
#include<stdio.h>
#include<string.h>
int main(void) {
    char name[50];
    int age;
    char sex;
    char* namep;
    while (1) {
        printf("Provide your personal information:\n");
        printf("Name:");
        gets(name);
        int check = strcmp(name, "stop"); // 문자열이 stop이면 0을 반환
        if (check == 0) {
            break;
        }
        namep = strtok(name, " ");
        namep = strtok(NULL, " ");
        // 성과 이름 나누기
        printf("Age:");
        scanf("%d", &age);
        while (getchar() != '\n'); // 버퍼 지우기
        printf("Sex(M/F):");
        sex = getchar();
        while (getchar() != '\n'); // 버퍼 지우기
        printf("Your name is %s %s, your age is %d, your sex is %c.\n\n", namep, name, age, sex);
    }
    return 0;
}
```

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

int main(){
    char name[50];
    int age;
    char sex;
    char *namep;
    char *sArr[2]; // 임시 저장소 생성

    while(1){
        printf("Provide your personal information\n");
        printf("Name :");
        scanf(" %[^\\n]", name); // 엔터키 말고 전부 입력 받음
        if(strcmp(name, "stop")==0) // stop이 나오면 중지시킴
            break;

        printf("Age :");
        scanf(" %d", &age);
        printf("Sex(M/F) :");
        scanf(" %c", &sex);

        namep = strtok(name, " "); // 공백란 이전 문자열 뽑아내기
        sArr[0] = namep;
        namep = strtok(NULL, " "); // 공백란 이후 문자열 뽑아내기
        sArr[1] = namep;
        printf("Your name is %s %s, your age is %d, your sex is %c.\n", sArr[1], sArr[0], age, sex);
    }
    return 0;
}
```