Programming Basics (HW#1)

Data Structure

Im Y. Jung

Problem

다음 조건을 만족하는 C 프로그램을 작성하시오.

– 이름(영어, 이름 성 순), 나이, 성별을 입력받아 이를 출력하는 프로그램. 단 이름의 출력은 성 이름의 순서가 되도록 해야하고, 나이는 10대(10s), 20대(20s), 30대(30s) 등으로 표현되도록 해야함.
– 몇 명의 데이터가 입력될지 모르니, 이름에 stop이 입력되면 프로그램이 끝나도록 만듦.
– 다음 프로그램 구조를 꼭 사용할 것.

```
int main()
{
        struct {
                char fullname[50];  // 전체이름
                char *lastname;  // 성을 가리키는 포인터
                char *firstname;    // 이름을 가리키는 포인터
                int age;
                char sex;
        } student;

        //programming going on
        // 아래에 코드를 채워넣으세요.



    return 0;
}
```

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 to familyname and firstname**
  - **Use the program structure given**

```
struct{
    char fullname[50];
    int age;
    char sex;
    char *lastname;
    char *firstname;
} student;
```

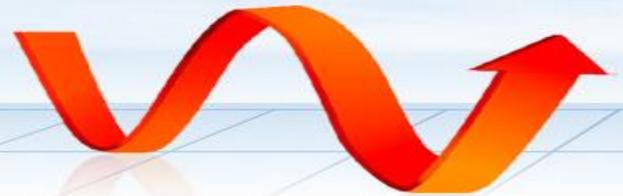
- Execution

```
Provide your personal information:
    Name : sungsoo kim
    Age : 21
    Sex (M/F) : M

Your name is kim sungsoo, you are in 20s, your sex is M.
-------------------------------------------------------
Provide your personal information:
    Name : mijin lee
    Age : 15
    Sex (M/F) : F

Your name is lee mijin, you are in 10s, 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

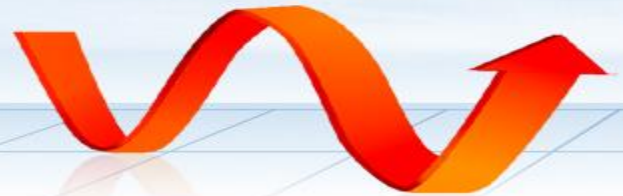
- Program Flow
  - **Algorithm**
  - **Program Structure**



Example – Problem Analysis (1)

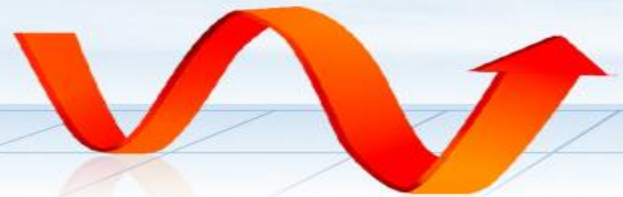
- Input : name(first last), age, sex
- Output : name(last first), age, sex
- Requirements :
  - **If “stop” is entered to name, your program will end.**
  - **If not, your program will go on.**
  - **You should use the given program structure.**

- 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)

- 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**

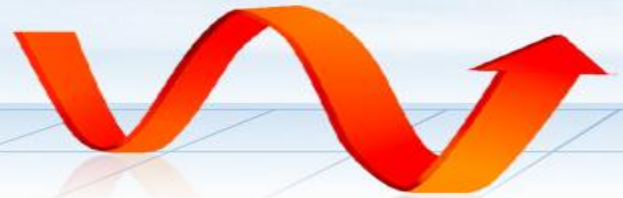


Example – Data/storage Design

```
char name[50];

int age;

char sex;


char *firstname;
char *lastname;
```

- 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**

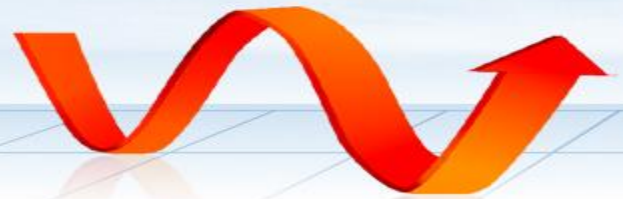


Example – Data/storage Design (2-1)

```
struct {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
} student;
```

- *cf.*

```
struct student {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
};
```



Example – Data/storage Design (2-2)

```
struct {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
    } student;
```

student.name
student.age
student.sex
student.firstname
student.lastname

- *cf.*

```
struct student {
        char name[50];
        int age;
        char sex;
        char *namep;
    };
```

(struct) student std;
std.name
std.age
std.sex
std.firstname
std.lastname

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)

- **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. (ex) %s, %d**
  - The specifiers that do not ignore whitespace are a **[, c, C,** and **n**.
- **Space in scanf format**
  - "skip all (0 or more) whitespace" from the current position on.
- **To use fflush() ?**
  - C Standard specifies the behavior of fflush(stdin) is undefined.
- **getchar()**

```
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;
}
```

```
a
a
```

```
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;
}
```

```
a
97
a
97
```

- 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;
}
```

```
1
2
3
Out1: 1 2 3
4 5 6
Out2: 4 5 6
```

scanf (2)

- **%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.

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

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

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

```
Enter a string: abababdabcab
The string is: ababab
```

```
Enter a string: ab cabcabab
The string is: ab
```

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

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

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

```
Enter a string: edfcdefdef
The string is: edf
```

scanf (3)

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

```c
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;
}
```

```
Enter age: 23
Enter name: Name is: -16, age is: 23
```

```c
#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;
}
```

```
Enter age: 23
Enter name: Gildong Hong
Name is: Gildong Hong, age is: 23
```

```c
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;
}
```

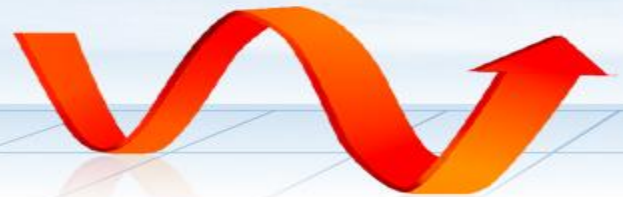
```
Enter age: 23
Enter name: Gildong Kim
Name is: Gildong Kim, age is: 23
```

fgets

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

```c
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;
}
```

```
Enter age: 23
Enter name: Gildong kim
Name is: Gildong kim
, age is: 23
```



Example – Program Flow (3)

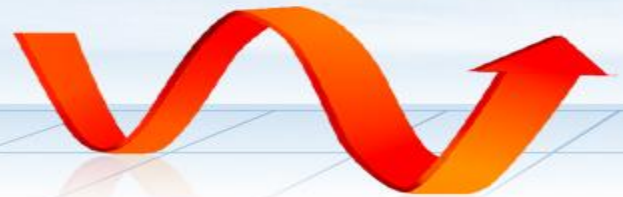
- **if(**"stop" is entered**)**

```c
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);
}
```

```
str2 is less than str1
```

- 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

```c
#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);
}
```

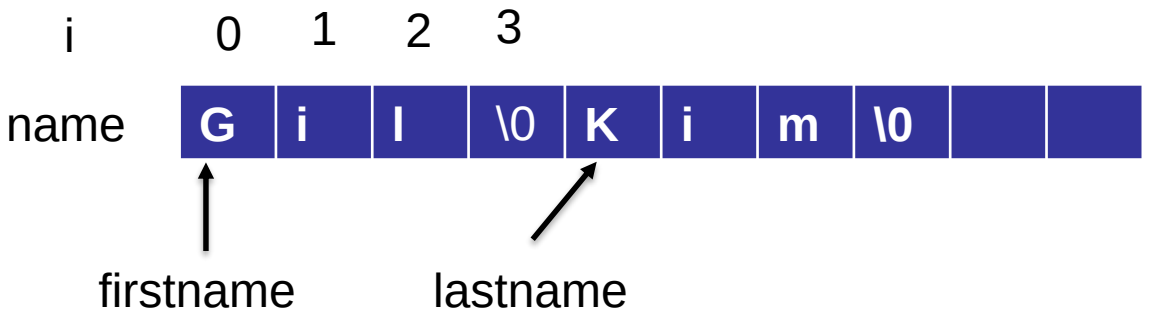
```
token : -
str : -

token : This,
str : -

token : a
str : -

token : sample
str : -

token : string.
str : -
```

```c
#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);
}
```

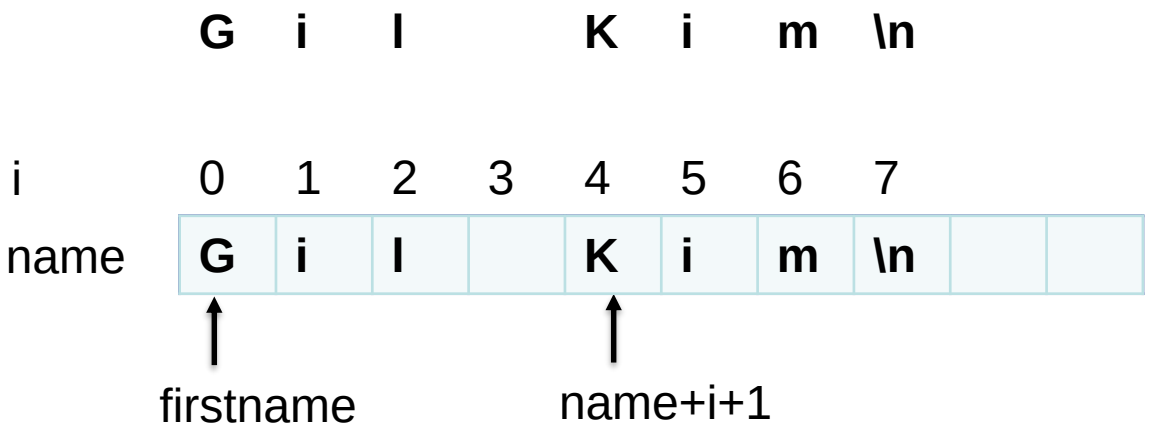
```
token : This
str : - This

token : a
str : - This

token : sample
str : - This

token : string
str : - This
```

Samples

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

int main()
{
    int i;

    struct {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
    } student;

    while(1){
        printf("\nProvide your personal information : \n");
        printf("Name : ");
        scanf(" %[^\n]s", student.name);

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

        printf("Age : ");
        scanf("%d", &student.age);

        printf("Sex(M/F) : ");
        scanf(" %c", &student.sex);

        for(i=0, student.firstname = student.name; student.name[i] != ' '; i++);
        student.name[i] = '\0';
        student.lastname = student.name+i+1 ;

        printf("Your name is %s %s,", student.lastname, student.firstname);
        printf(" your age is %ds, your sex is %c\n", (int)(student.age/10)*10, student.sex);
    }

    return 0;
}
```



Samples




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

int main()
{
    int i;

    struct {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
    } student;

    while(1){
        printf("\nProvide your personal information : \n");
        printf("Name : ");
        scanf(" %[^\n]s", student.name);

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

        printf("Age : ");
        scanf("%d", &student.age);

        printf("Sex(M/F) : ");
        scanf(" %c", &student.sex);

        for(i=0, student.firstname = student.name; student.name[i] != ' '; i++);
        student.name[i] = '\0';
        student.lastname = student.name+i+1 ;

        printf("Your name is %s %s,", student.lastname, student.firstname);
        printf(" your age is %ds, your sex is %c\n", (int)(student.age/10)*10, student.sex);
    }

    return 0;
}
```

Samples – HW#1

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

int main()
{
    int i;

    struct {
        char name[50];
        int age;
        char sex;
        char *firstname;
        char *lastname;
    } student;

    while(1){
        printf("\nProvide your personal information : \n");
        printf("Name : ");
        scanf(" %[^\n]s", student.name);

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

        printf("Age : ");
        scanf("%d", &student.age);

        printf("Sex(M/F) : ");
        scanf(" %c", &student.sex);

        // 성, 이름 분리
        student.firstname = strtok(student.name, " ");   // 공백이전 문자열 추출
        student.lastname = strtok(NULL, " ");            // 공백이후 다음 문자열 추출

        printf("Your name is %s %s,", student.lastname, student.firstname);
        printf(" your age is %ds, your sex is %c\n", (int)(student.age/10)*10, student.sex);
    }

    return 0;
}
```

- Using strtok