C Programming Language

VLSI Design Lab., Konkuk Univ.

Yong Beom Cho

ybcho@konkuk.ac.kr

LSI Design Lab

Introduction

- C
  - is a general-purpose programming language. But C is a relatively “low level” language.
  - provides a variety of data types.
  - Provides the fundamental control-flow constructions required for well-structured programs: statement grouping, decision making, selecting one of a set of possible cases, looping with the termination test at the top or at the bottom, and early loop exit.
- C Program
  - Whatever its size, consists of functions and variables.
  - A function contains statements that specify the computing operations to be done, and variables store values used during the computation.
  - main will usually call other function to help perform its job, some that you wrote, and others from libraries that are provided for you.

C Language - I

- Comment – explains briefly what the program does.
  - /* and */
  - //
- Declaration – announces the properties of variables.
  - int fahr, celsius;
  - int lower, upper, step;
    - int integers
    - float floating point
    - char character – a single byte
    - short short integer
    - long long integer
    - double double-precision floating point
- Assignment statements
  - lower = 0;
  - upper = 300;

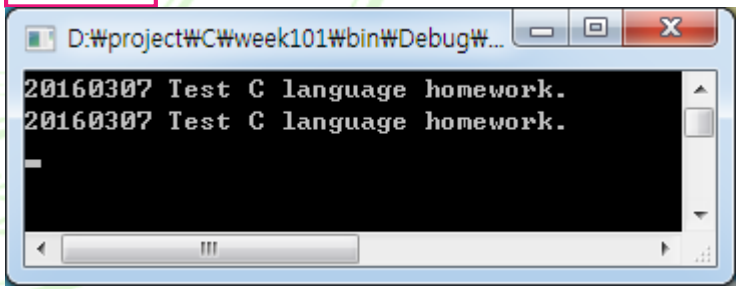
C Language - II

- printf
  - %d : print as decimal integer
  - %6d
  - %f : print as floating point
  - %6f
  - %6.2f
- Character Input and output
  - C = getchar();
  - putchar(c);
  - Copy
  - printf(“”,...);
  - scanf(“%d”,&a);

Source

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main()
5   {
6       int c;
7       c = getchar();
8       while(c != EOF)
9       {
10          putchar(c);
11          c = getchar();
12      }
13      return 0;
14  }
15
```

Result



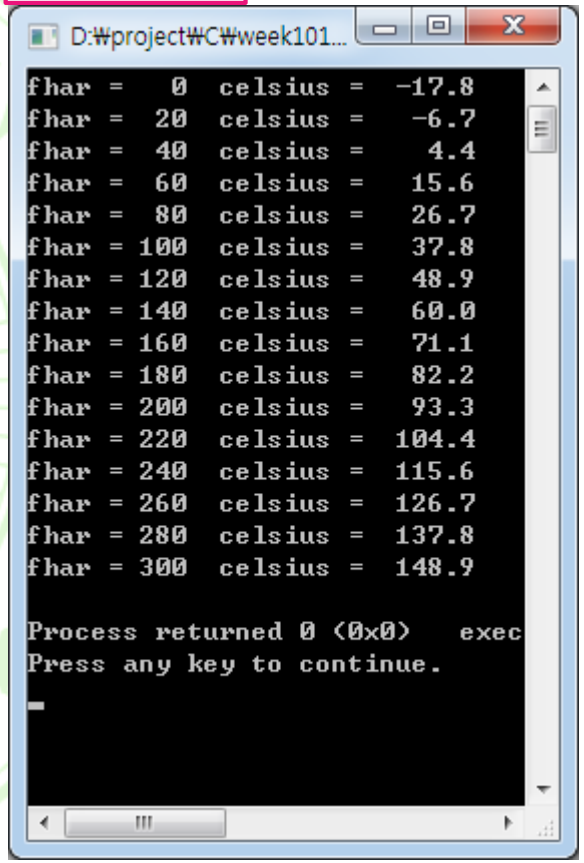
Examples – 온도변환 프로그램

✓ ***File name : ex_01.c***

Source

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main()
5   {
6       float fahr=0, celsius;
7       int lower, upper, step;
8
9       lower = 0;
10      upper = 300;
11      step = 20;
12      while(fahr <= upper)
13      {
14          celsius = (5.0/9.0) * (fahr - 32.0);
15          printf("fhar = %3.0f  celsius = %6.1f\n", fahr, celsius);
16          fahr = fahr + step;
17      }
18      return 0;
19  }
20
```

Result



Examples – Data types and sizes

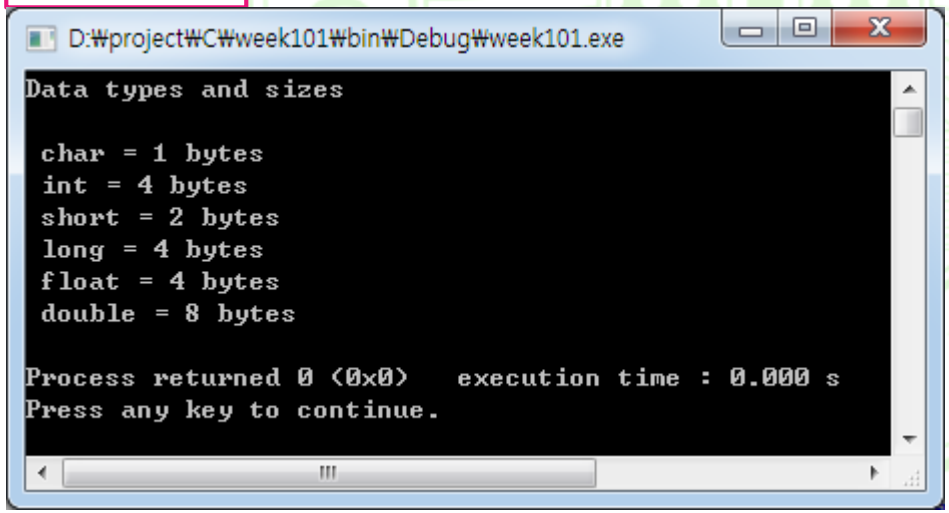
✓ ***File name : ex_02.c***

Source

```c
#include <stdio.h>
#include <stdlib.h>
/* Data types and sizes */
int main()
{
    printf("Data types and sizes \n\n");
    printf(" char = %d bytes \n int = %d bytes\n",sizeof(char), sizeof(int));
    printf(" short = %d bytes \n long = %d bytes\n",sizeof(short), sizeof(long));
    printf(" float = %d bytes \n double = %d bytes\n",sizeof(float), sizeof(double));
    return 0;
}

```

Result



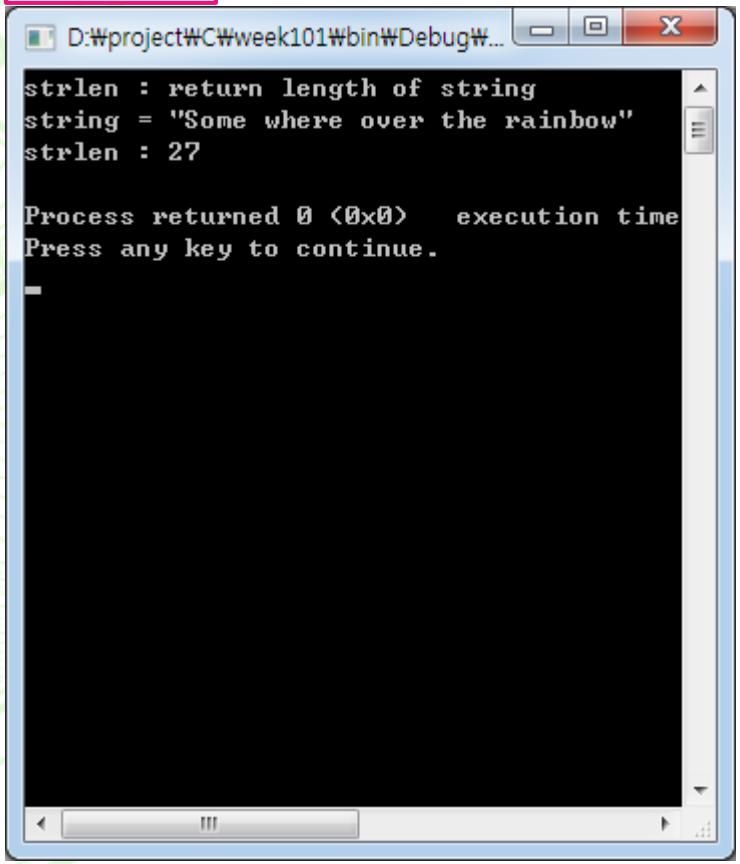
Examples – return length of string

- ✓ ***File name : ex_03.c***

Source

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   /* strlen : return length of s */
4   int strlen(unsigned char str[]);
5   int main()
6   {
7       int len;
8       unsigned char *str = "Some where over the rainbow";
9
10      printf("strlen : return length of string\n");
11      printf("string = \"%s\"\n", str);
12      len = strlen(str);
13      printf("strlen : %d\n", len);
14      return 0;
15  }
16  int strlen(unsigned char str[])
17  {
18      int i=0;
19      while(str[i] != '\0')
20      {
21          ++i;
22      }
23      return i;
24  }
25
```

Result



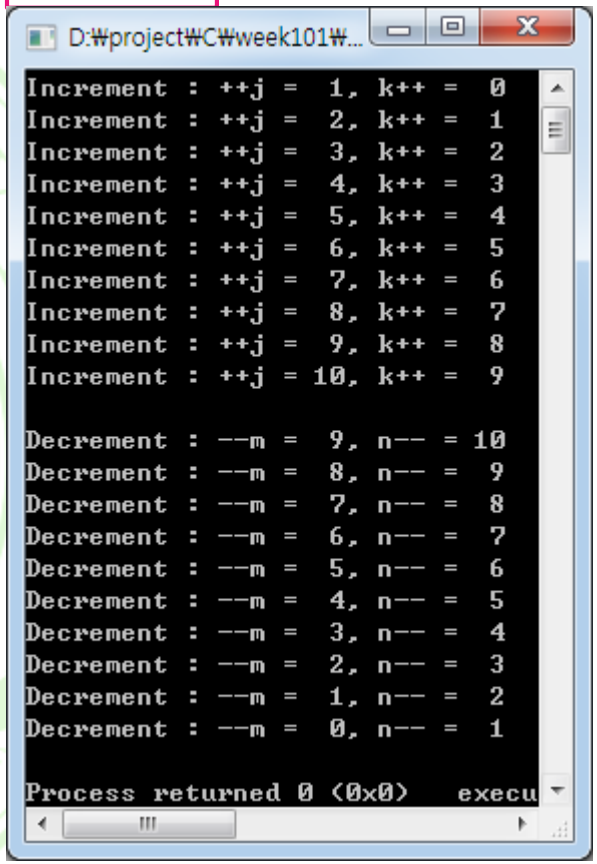
Examples – 증가/감소 연산자

✓ ***File name : ex_04.c***

Source

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main()
5   {
6       int i;
7       int j=0,k=0;
8       int m=10, n=10;
9
10      for(i=0;i<10;i++)
11      {
12          printf("Increment : ++j = %2d, k++ = %2d \n",++j,k++);
13      }
14
15      printf("\n");
16
17      for(i=0;i<10;i++)
18      {
19          printf("Decrement : --m = %2d, n-- = %2d \n",--m,n--);
20      }
21      return 0;
22  }
23
```

Resource



Control Flow

Control Flow - I

- Statements and Blocks
  - Statement

```
1 x=0;
2 i++;
3 printf(...);
```

  - Block

    { }

- If-Else

```
If (expression)
    Statement1
else
    Statement2
```

- Else-if

```
if (expression)
        Statement1
else if (expression)
        Statement2
else if (expression)
        Statement3
else if (expression)
        Statement4
else
        Statement5
```

Examples – (else-if)

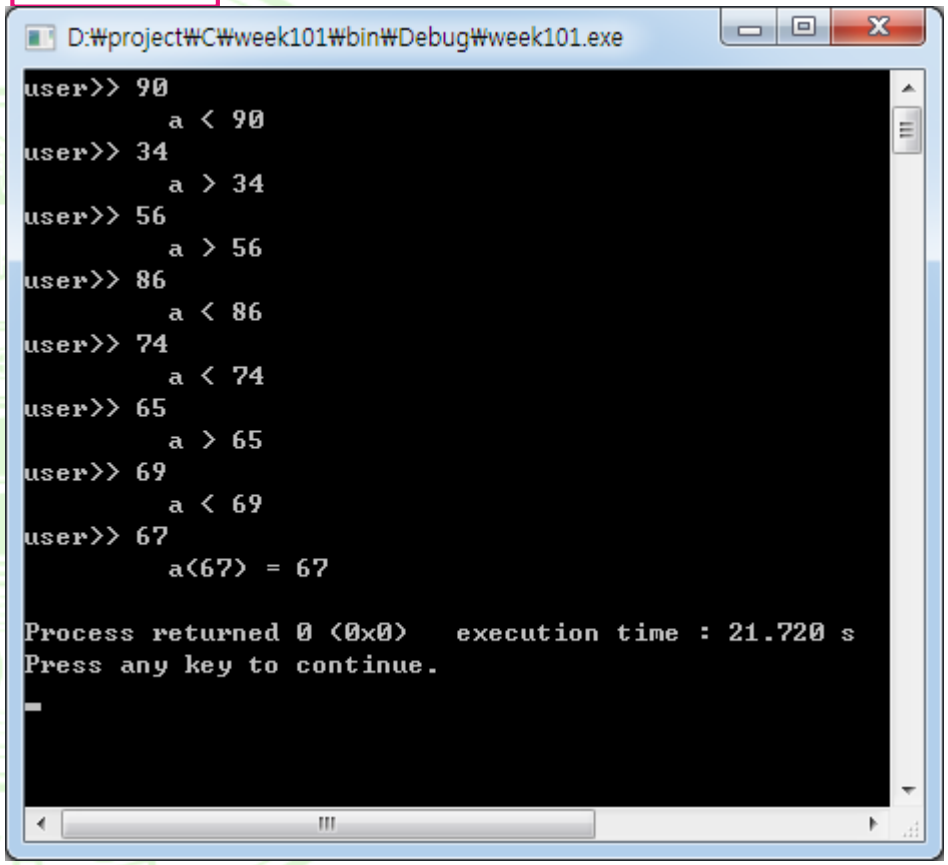
✓ ***File name : ex_06.c***

Source

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
    int a, b;
    srand((unsigned)time(NULL));
    a = rand()%100 + 1;
    while(1)
    {
        printf("user>> ");
        scanf("%d",&b);
        if(a == b)
        {
            printf("\t a(%d) = %d \n", a, b);
            break;
        } else if (a > b) {
            printf("\t a > %d \n",b);
        } else if (a < b) {
            printf("\t a < %d \n",b);
        }
    }
    return 0;
}
```

Result



Control Flow - II

- switch~case

```
switch (expression) {
    case const-expr:
                statement1
    case const-expr:
                statement2
    default: statement3
}
```

Examples – (switch~case)

✓ ***File name : ex_07.c***

Source

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   int sumAB(int a, int b);
4   int mulAB(int a, int b);
5   int main()
6   {
7       int a, b, n, result;
8       while(1)
9       {
10          printf("1:ADD, 2:MUL, 3: quit\n");
11          scanf("%d",&n);
12          if(n==3)
13              break;
14          printf("a : ");
15          scanf("%d", &a);
16          printf("b : ");
17          scanf("%d", &b);
18          switch(n) {
19              case 1 : result = sumAB(a,b);
20                  printf("%d + %d = %d\n\n",a,b,result);
21                  break;
22              case 2 : result = mulAB(a,b);
23                  printf("%d * %d = %d\n\n",a,b,result);
24                  break;
25              default: break;
26          }
27      }
28      return 0;
29  }
30  int sumAB(int a, int b)
31  {
32      return a+b;
33  }
34  int mulAB(int a, int b)
35  {
36      return a*b;
37  }
```

Result