Expressions &
Selection statement

adopted from KNK C Programming : A Modern Approach

배열과 포인터

- 임의의 수(2명 이상)로 짝을 지어서 포인터를 이해할 수 있는 방법을 구상해오기
  - 게임
  - 그림
  - 상자, 줄 등의 오브젝트를 활용한 설명
  - 역활극
  - 그 외 방법

- 선정 팀에게는 상금 5만원

- 2번째 시험 후 실습 수업시간에 제출 및 설명

Expressions

operators.c

```
#include <stdio.h>
int main(){
    float a = 5;
    float b = 10;
    return a%b;
}
```

Compound Operators: compound.c

```
#include <stdio.h>
int main() {
    int i = 10;
    int j = 10;
    int k = 1;

    printf(“%d\n”, I *= j + k);

    i = j = 10;
    printf(“%d\n”, I = I * j +k);

    return 0;
}
```

Inc/Decrement Operators: inc-decrement.c

```
#include <stdio.h>

int main() {
    int i = 1;
    printf("i is %d\n", ++i); // prints "i is 2”
    printf("i is %d\n", i);   // prints "i is 2”

    i = 1;
    printf("i is %d\n", i++); // prints "i is 1”
    printf("i is %d\n", i);   // prints "i is 2”

    return 0;
}
```

Inc/Decrement Operators: inc-decrement1.c

```c
#include <stdio.h>

int main() {
    int i = 1;
    printf("i is %d\n", --i); // prints "i is 0”
    printf("i is %d\n", i);   // prints "i is 0”

    i = 1;
    printf("i is %d\n", i--); // prints "i is 1”
    printf("i is %d\n", i);   // prints "i is 0”

    return 0;
}
```

Inc/Decrement Operators: inc-decrement2.c

```c
#include <stdio.h>

int main() {
    int i = 1;
    int j = 2;
    int k = 0;

    printf("k is %d\n", k = ++i + j++);
//  printf("k is %d\n", k = i++ + j++);

    return 0;
}
```

Order of Subexpression Evaluation

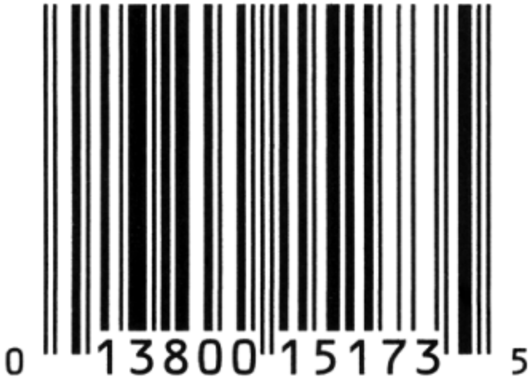
```
#include <stdio.h>

int main() {
    int i = 1;
    int j = 0;
    printf("%d\n", j = i * i++);
//  printf("%d\n", j = i * i++ * i++);

    return 0;
}
```

Program: Computing a UPC Check Digit

- Most goods sold in U.S. and Canadian stores are marked with a Universal Product Code (UPC):



- Meaning of the digits underneath the bar code:

첫 숫자: Type of item
첫 5개 숫자 묶음: Manufacturer
두번째 5개 숫자 묶음: Product (including package size)
마지막 숫자: Check digit, used to help identify an error in the preceding digits (검산용 숫자)

Program: Computing a UPC Check Digit

- Example for UPC 0 13800 15173 5:
- How to compute the check digit:

  Add the first, third, fifth, seventh, ninth, and eleventh digits.

  First sum: 0 + 3 + 0 + 1 + 1 + 3 = 8.

  Add the second, fourth, sixth, eighth, and tenth digits.

  Second sum: 1 + 8 + 0 + 5 + 7 = 21.

  Multiply the first sum by 3 and add it to the second sum.

  total: 3 * (First sum) – second sum = 45

  Subtract 1 from the total.

  result: total – 1 = 44

  Compute the remainder when the adjusted total is divided by 10.

  remainder: result % 10 = 4

  Subtract the remainder from 9.

  Check digit: 9 – remainder = 5

Program: Computing a UPC Check Digit

- The `upc.c` program asks the user to enter the first 11 digits of a UPC, then displays the corresponding check digit:

```
Enter the first (single) digit: 0
Enter first group of five digits: 13800
Enter second group of five digits: 15173
Check digit: 5
```

- The program reads each digit group as five one-digit numbers.
- To read single digits, we'll use `scanf` with the `%1d` conversion specification.

upc.c

```
/* Computes a Universal Product Code check digit */

#include <stdio.h>

int main(void)
{
  int d, i1, i2, i3, i4, i5, j1, j2, j3, j4, j5,
      first_sum, second_sum, total;

  printf("Enter the first (single) digit: ");
  scanf("%1d", &d);
  printf("Enter first group of five digits: ");
  scanf("%1d%1d%1d%1d%1d", &i1, &i2, &i3, &i4, &i5);
  printf("Enter second group of five digits: ");
  scanf("%1d%1d%1d%1d%1d", &j1, &j2, &j3, &j4, &j5);
  first_sum = d + i2 + i4 + j1 + j3 + j5;
  second_sum = i1 + i3 + i5 + j2 + j4;
  total = 3 * first_sum + second_sum;

  printf("Check digit: %d\n", 9 - ((total - 1) % 10));

  return 0;
}
```

Program: Computing a UPC Check Digit

- Question:
  - upc.c 에서 9 – ((total – 1) % 10)을 (10 – (total % 10)) % 10으로 바꿔도 정상적으로 동작할까?

Program: Calculating a Broker’s Commission

- When stocks are sold or purchased through a broker, the broker’s commission often depends upon the value of the stocks traded.
- Suppose that a broker charges the amounts shown in the following table:

Transaction size	*Commission rate*
Under $2,500	$30 + 1.7%
$2,500–$6,250	$56 + 0.66%
$6,250–$20,000	$76 + 0.34%
$20,000–$50,000	$100 + 0.22%
$50,000–$500,000	$155 + 0.11%
Over $500,000$255 + 0.09%	

- The minimum charge is $39.

Program: Calculating a Broker’s Commission

- The `broker.c` program asks the user to enter the amount of the trade, then displays the amount of the commission:

```
Enter value of trade: 30000
Commission: $166.00
```

- The heart of the program is a cascaded `if` statement that determines which range the trade falls into.

broker.c

```
/* Calculates a broker's commission */

#include <stdio.h>

int main(void)
{
  float commission, value;

  printf("Enter value of trade: ");
  scanf("%f", &value);

  ...
```

broker.c

```
...
if (value < 2500.00f)
    commission = 30.00f + .017f * value;
else if (value < 6250.00f)
    commission = 56.00f + .0066f * value;
else if (value < 20000.00f)
    commission = 76.00f + .0034f * value;
else if (value < 50000.00f)
    commission = 100.00f + .0022f * value;
else if (value < 500000.00f)
    commission = 155.00f + .0011f * value;
else
    commission = 255.00f + .0009f * value;
...
```

broker.c

```
 ...

 if (commission < 39.00f)
   commission = 39.00f;

 printf("Commission: $%.2f\n", commission);

 return 0;
}
```

Equality and Assignments: eq_assign.c

```
#include <stdio.h>

int main() {
    int i = 1;

    if (i == 0)
        printf("check this\n");

    if ((i = 0))
        printf("print this line\n");

    return 0;
}
```

The “Dangling `else`” Problem: dangling_else.c

```
#include <stdio.h>
int main() {
    int x = 10;
    int y =  5;
    int result;

    if (y != 0)
        if (x != 0)
            result = x / y;
    else
        printf("Error: y is equal to 0\n");

    return 0;
}
```

Dangling `else` 문제를 없애려면?

Conditional Expressions: cond_exp.c

```c
#include <stdio.h>
int main() {
    int i, j, k;
    i = 1;
    j = 2;

    k = i > j ? i : j;        /* k is now 2 */
    printf("%d\n", k);

    k = (i >= 0 ? i : 0) + j; /* k is now 3 */
    printf("%d\n", k);

    return 0;
}
```

The `switch` Statement: switch.c

```c
#include <stdio.h>
int main() {
    int grade = 4;
    switch (grade) {
      case 4:  printf("Excellent");
               break;
      case 3:  printf("Good");
               break;
      case 2:  printf("Average");
               break;
      case 1:  printf("Poor");
               break;
      case 0:  printf("Failing");
               break;
      default: printf("Illegal grade");
               break;
    }
    return 0;
}
```

The `switch` Statement: switch1.c

```c
#include <stdio.h>
int main() {
    int grade = 4;
    switch (grade) {
        case 4:
        case 3:
        case 2:
        case 1:  printf("Passing");
                 break;
        case 0:  printf("Failing");
                 break;
        default: printf("Illegal grade");
                 break;
    }

    return 0;
}
```

Role of break: break.c

```
#include <stdio.h>
int main() {
    int grade = 4;
    switch (grade) {
      case 4:  printf("Excellent\n");
      case 3:  printf("Good\n");
      case 2:  printf("Average\n");
      case 1:  printf("Poor\n");
      case 0:  printf("Failing\n");
      default: printf("Illegal grade\n");
    }
    return 0;
}
```

Program: Printing a Date in Legal Form

- Contracts and other legal documents are often dated in the following way:

```
Dated this __________ day of __________ , 20__ .
```

- The `date.c` program will display a date in this form after the user enters the date in month/day/year form:

```
Enter date (mm/dd/yy): 7/19/14
Dated this 19th day of July, 2014.
```

- The program uses `switch` statements to add “th” (or “st” or “nd” or “rd”) to the day, and to print the month as a word instead of a number.

date.c

```
/* Prints a date in legal form */

#include <stdio.h>

int main(void)
{
  int month, day, year;

  printf("Enter date (mm/dd/yy): ");
  scanf("%d /%d /%d", &month, &day, &year);

  printf("Dated this %d", day);
  switch (day) {
    case 1: case 21: case 31:
      printf("st"); break;
    case 2: case 22:
      printf("nd"); break;
    case 3: case 23:
      printf("rd"); break;
    default: printf("th"); break;
  }
  printf(" day of ");
  switch (month) {
    case 1:  printf("January");   break;
    case 2:  printf("February");  break;
    case 3:  printf("March");     break;
    case 4:  printf("April");     break;
    case 5:  printf("May");       break;
    case 6:  printf("June");      break;
    case 7:  printf("July");      break;
    case 8:  printf("August");    break;
    case 9:  printf("September"); break;
    case 10: printf("October");   break;
    case 11: printf("November");  break;
    case 12: printf("December");  break;
  }

  printf(", 20%.2d.\n", year);
  return 0;
}
```