

Introduction to software design

2012-1

Final

2012.06.13 16:00-18:00

Student ID:_____

Name:_____

0. 표지에 이름과 학번을 적으시오. (6)

1. 변수 x, y가 integer type이라 가정하고 다음 빈 칸에 x와 y의 계산 결과 값을 적으시오. (5)

$x = (3 + 7) * 6;$	$x =$	
$x = (12 + 6) / 2 * 3;$	$x =$	
$x = 3 * (8 / 4) + 2;$	$x =$	
$y = x = (2 + 3)/4;$	$x =$	$y =$
$y = 3 + 2 * (x = 7 / 2);$	$x =$	$y =$

2. 각각의 문장을 condition으로 표현하시오. (4)

x는 5보다 크다.	$x > 5$
x는 5가 아니다.	
x는 20보다 크거나 같다.	
x는 1과 9를 포함하는 두 수 사이의 값이다.	
x는 1과 9를 포함하는 두 수 사이의 값이 아니다.	

3. 다음 While-loop를 for-loop으로 바꿔 표현하시오. (10)

```
// sweetie1.c -- a counting loop

#include <stdio.h>

int main(void)
{
    const int NUMBER = 22;
    int count = 1;

    while (count <= NUMBER)
    {
        printf("Be my Valentine!\n");
        count++;
    }
    return 0;
}
```

for-loop code

```
// sweetie1.c -- a counting loop

#include <stdio.h>

int main(void)
{
    const int NUMBER = 22;
    int count;

    for (  ;  ;  )
        printf("Be my Valentine!\n");

    return 0;
}
```

4. 다음과 같은 결과를 출력하는 프로그램의 빈 칸을 채우시오. (10)

출력 결과

```
Pick an integer from 1 to 100. I will try to guess it.  
Respond with a y if my guess is right and with  
an n if it is wrong.  
Uh...is your number 1?  
n  
Well, then, is it 2?  
no  
Well, then, is it 3?  
no sir  
Well, then, is it 4?  
forget it  
Sorry, I understand only y or n.  
n  
Well, then, is it 5?  
y  
I knew I could do it!
```

```
#include <stdio.h>  
  
int main(void) {  
    int guess = 1;  
    int response;  
  
    printf("Pick an integer from 1 to 100. I will try to guess ");  
    printf("it.\nRespond with a y if my guess is right and with");  
    printf("\nan n if it is wrong.\n");  
    printf("Uh...is your number %d?\n", guess);  
  
    while ((response = getchar()) != 'y') {  
        if (  )  
            printf("Well, then, is it %d?\n", ++guess);  
        else  
            printf("Sorry, I understand only y or n.\n");  
        while (  )  
            continue;  
    }  
  
    printf("I knew I could do it!\n");  
    return 0;  
}
```


(2)

```
#include<stdio.h>

int main(void) {
    int i,j,n;
    int set[100][100];
    n = 6;
    for(i = 0; i <= n; i++) {
        for(j = 0; j <= n; j++) {
            set[i][j] = 0;
        }
    }
    set[0][0] = 1;
    for(i = 1; i <= n; i++)
        for(j = 1; j <= i; j++)
            set[i][j] = set[i - 1][j] + set[i - 1][j - 1];

    printf(" Index");
    for(i = 1; i <= n; i++)
        printf("%3d", i);
    printf("\n");

    for(i = 1; i <= n; i++) {
        printf("%4d  ", i);
        for(j = 1; j <= n; j++)
            printf("%3d",set[i][j]);
        printf("\n");
    }
    return 0;
}
```

출력결과

Index	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

6. 다음과 같은 결과를 출력하는 프로그램의 빈 칸을 채우시오. (10)

Enter an uppercase letter: D

```
      A
    A B A
  A B C B A
A B C D C B A
```

Enter an uppercase letter: E

```
      A
    A B A
  A B C B A
A B C D C B A
A B C D E D C B A
```

Enter an uppercase letter: G

```
      A
    A B A
  A B C B A
A B C D C B A
  A B C D E D C B A
A B C D E F E D C B A
A B C D E F G F E D C B A
```



```

#include <stdio.h>
void pyramid(char letter);
int main(void) {
    char letter;

    printf("Enter an uppercase letter: ");
    scanf("%c", &letter);

    pyramid(letter);

    return 0;
}
void pyramid(char letter)
{
    int i, j;
    for(i = 'A'; i <= letter; i++)
    {
        for(j = 0 ; j <=  ; j++)
            printf(" ");
        for(j = 0 ; j <=  ; j++)
            printf("%c", j+'A');
        for(j =  ; j >= 0; j--)
            printf("%c", j+'A');

        printf("\n");
    }
}

```

7. 다음 빈 칸을 채우시오. (10)

```
/* order.c -- precedence in pointer operations */

#include <stdio.h>

int data[2] = {200, 300};
int moredata[2] = {150, 50};
int main(void)
{
    int * p1, * p2, * p3;

    p1 = p2 = data;
    p3 = moredata;

    printf("    *p1 = %d,    *p2 = %d,    *p3 = %d\n",
           *p1      ,    *p2      ,    *p3);

    printf(" *++p1 = %d, *p2++ = %d, (*p3)++ = %d\n",
           *++p1      , *p2++      , (*p3)++);

    printf("    *p1 = %d,    *p2 = %d,    *p3 = %d\n",
           *p1      ,    *p2      ,    *p3);

    return 0;
}
```

출력결과

*p1 = *p2 = *p3 =

*++p1 = *p2++ = (*p3)++ =

*p1 = *p2 = *p3 =

8. 다음 빈 칸을 채우시오. (25)

(1) 다음은 ‘삽입정렬’을 수행하는 프로그램이다. 아래와 같이 새로운 수를 하나씩 입력하면 이미 정렬된 배열에 알맞게 삽입하고 출력하는 것을 반복한다. 이 삽입 정렬을 수행하는 프로그램의 빈 칸을 채우시오.

입출력 예제

```
2
2
4
2 4
6
2 4 6
5
2 4 5 6
10
2 4 5 6 10
1
1 2 4 5 6 10
3
1 2 3 4 5 6 10
7
1 2 3 4 5 6 7 10
13
1 2 3 4 5 6 7 10 13
9
1 2 3 4 5 6 7 9 10 13
11
1 2 3 4 5 6 7 9 10 11 13
-1
```

```

#include <stdio.h>
#define MAXSIZE 10000

int main(void) {
    int i, loc, currentSize = 0;
    int inputNum;
    int num[MAXSIZE];

    while (scanf("%d", &inputNum) == 1) {
        if (inputNum == -1)
            break;
        else if(currentSize == MAXSIZE) {
            printf("ERROR: Memory is full.");
            break;
        }
        else {
            ;
            while (  &&  ){
                num[loc] = num[loc-1];
                
            }
            num[loc] = ;

            for (i = 0; i < currentSize; i++)
                printf("%d ", num[i]);
            printf("\n");
        }
    }
    return 0;
}

```

(2) 아래는 8-(1)의 코드의 일부분이다. 여기서 if 문을 제거해도 같은 동작을 수행하도록 while 조건 부분을 수정하라.

```
while (scanf("%d", &inputNum) == 1) {  
    if (inputNum == -1)  
        break;  
    ...  
}
```



```
while(  ){  
    ....  
}
```

9. 다음과 같은 선언이 주어졌을 때, 각 statement가 올바른지 O, 그렇지 않으면 X로 표시하십시오. (10)

(1)

```
float rootbeer[10], things[10][5], *pf, value = 2.2;
int i = 3;
```

rootbeer[2] = value;	
----------------------	--

things[4][4] = rootbeer[3];	
-----------------------------	--

things[5] = rootbeer;	
-----------------------	--

pf = value;	
-------------	--

pf = rootbeer;	
----------------	--

(2)

```
double rates[5] = {88.99, 100.12, 59.45, 183.11, 340.5};
const double locked[4] = {0.08, 0.075, 0.0725, 0.07};
const double * pd = rates;
```

pd[2] = 222.22;	
-----------------	--

*pd = 29.89;	
--------------	--

pd++;	
-------	--

pd = &rates[3];	
-----------------	--

pd = locked;	
--------------	--

10. 다음의 가운데 정렬을 하는 프로그램의 빈 칸을 채우시오. (10)
(단, strlen()함수를 사용한 일반적인 답이 아닌 경우 오답처리)

출력결과

[illegible]

※Hint: `size_t strlen(const char *str)` 함수는 인자로 String을 받아 그 String의 길이를 리턴하는 함수이다.

예: `strlen("abcdefg") == 7`

```
strlen("12345 67890") == 11
```

```

#include <stdio.h>
#include <string.h>
#define NAME "GIGATHINK, INC."
#define ADDRESS "101 Megabuck Plaza"
#define PLACE "Megapolis, CA 94904"
#define WIDTH 40
#define SPACE ' '

void show_n_char(char ch, int num);

int main(void){
    int spaces;

    show_n_char('*', WIDTH);
    putchar('\n');
    show_n_char(SPACE, 12);
    printf("%s\n", NAME);
    spaces = 
    show_n_char(SPACE, spaces);
    printf("%s\n", ADDRESS);
    show_n_char(SPACE,  );
    printf("%s\n", PLACE);
    show_n_char('*', WIDTH);
    putchar('\n');
    return 0;
}

void show_n_char(char ch, int num) {
    int count;
    for (count = 1; count <= num; count++)
        putchar(ch);
}

```


11. 다음 if-elseif문을 switch-case문으로 바꿔 표현하시오. (15)

if-elseif문

```
#include<stdio.h>

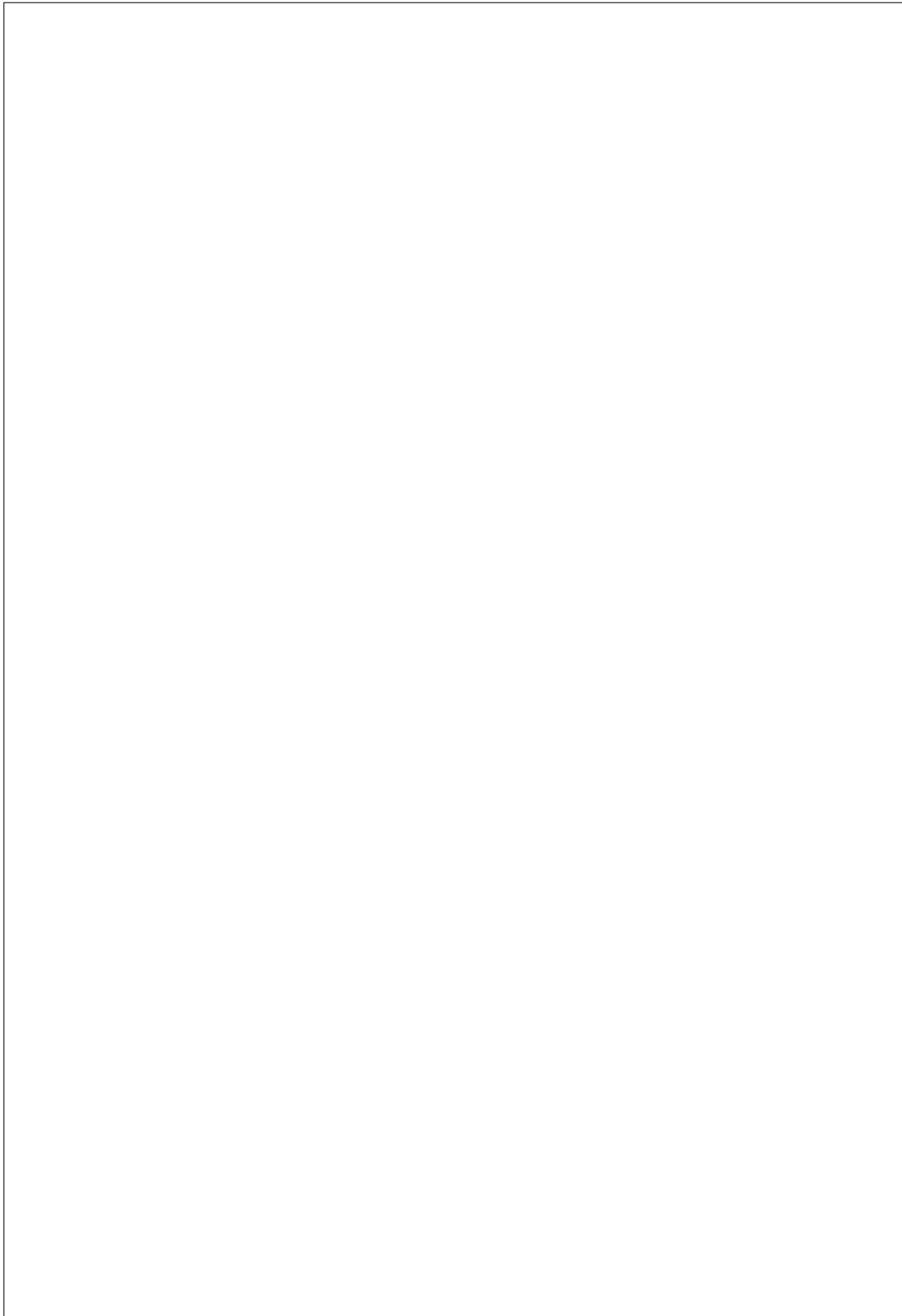
int main(void)
{
    char code;
    scanf("%c",&code);

    if(code == 'A' || code == 'a') printf("Your Score : 90-100\n");
    else if(code == 'B' || code == 'b') printf("Your Score : 80-89\n");
    else if(code == 'C' || code == 'c') printf("Your Score : 70-79\n");
    else if(code == 'D' || code == 'd') printf("Your Score : 60-69\n");
    else if(code == 'F' || code == 'f') printf("Your Score : - 60\n");
    else printf("Not known..\n");

    return 0;
}
```

Switch-case문

```
int main(void)
{
    char code;
    scanf("%c",&code);
```



```
    return 0;
```

```
}
```

12. 입력이 **abcdefg** 일 때, 다음과 같은 출력이 나오도록 빈 칸을 채우시오. (입력 문자열의 길이가 **1000**을 넘지 않는다.) (15)

출력결과

```
gabcdef
fgabcde
efgabcd
defgabc
cdefgab
bcdefga
abcdefg
```

```
#include<stdio.h>
#include<string.h>
int main(void) {
    
    return 0;
}
```