

CSED103: 프로그래밍 입문

April 20, 2009, Monday, 2:35 ~ 4:10am

1. (15점) 다음 질문에 대하여 간결하고 명확하게 대답하시오. (각 3점씩)
 - (a) identifier 란 무엇인가?
 - (b) call by value 란 무엇인가?
 - (c) switch 문과 if ... else if 문의 공통점과 차이점은 무엇인가?
 - (d) function 을 사용할 때 function declaration 을 해야 하는 이유는 무엇인가?
 - (e) divide and conquer 를 실생활에 적용할 수 있는 예는 어떤 것이 있는가?

6' < 2' 9'

2. (9점) 아래 프로그램에서 컴파일할 때 에러가 발생하는 부분을 지적하고 그 이유를 설명하시오.

```
1  #include <stdio.h>
2
3  double cal_average(int homework, int mid, int final);
4
5  int main()
6  {
7      int homework, mid, final;
8      double wAverage;
9      scanf("%d%d%d", &homework, &mid, &final);
10     waverager = cal_average(homework, mid, final);
11     if (waverager > 65)
12         printf("Passed\n");
13         printf("Average: %f\n", waverager);
14     else
15         printf("Failed\n");
16     return 0;
17 }
18
19 double cal_average(int homework, int mid, int final)
20 {
21     return(homework + weight1*mid + weight2*final);
22 }
```


3. (8점) 아래의 프로그램을 실행하였을 때, 화면에 출력되는 결과를 작성하시오. 참고로 main()에서 정의된 변수 x, y, z의 주소값은 각각 100, 104, 108이라 가정한다.

<pre>#include <stdio.h> void swapA(int x, int y) { int tmp; tmp = x; x = y; y = tmp; } void swapB(int *x, int *y) { int tmp; tmp = *x; *x = *y; *y = tmp; }</pre>	<pre>void getData(int *x, int *y) { printf("Enter the integers (x, y): "); scanf("%d %d", x, y); printf("x : %d, y : %d\n", x, y); } void main() { int x; int y; int z; getData(&x, &y); z = x * y; printf("x : %d, y : %d, z : %d\n", x, y, z); swapA(x, y); printf("x : %d, y : %d, z : %d\n", x, y, z); swapB(&y, &z); printf("x : %d, y : %d, z : %d\n", x, y, z); }</pre>
---	---

출력 결과

<pre>Enter the integers (x, y): 5 10 x: 5, y: 10 x: 5 y: 10 z: 50 swap A printf(5 10 50) printf(x, y, z)</pre>
--

4. (10점) 아래의 프로그램을 실행하였을 때, 화면에 출력되는 결과를 작성하시오.

```
#include <stdio.h>

int bar(int n){
    printf("bar: %d\n", n);
    switch (n){
        case 0: return 3;
        case 1: return 1;
        default: return bar(n-1) + 5;
    }
}

int foo(int n) {
    int x;

    if (n==0)
        return 0;
    x = foo(n-1);
    printf("foo: %d %d\n", n, x);
    if (bar(n-1) > x)
        return bar(n-1);
    return x;
}

int main(void) {
    printf("main: %d\n", foo(3));
    return 0;
}
```

foo: 1, 0
 if (bar: 0 > 3) \Rightarrow 3
 bar: 0
 foo: 2, 3
 bar: 1 \Rightarrow 3
 foo: 3, 3
 bar: 2
 bar: 1
 bar: 2
 bar: 1
 return 6

1. foo(0)
 2. foo(1)
 3. foo(2)
 4. bar: 1

foo: 1, 0
 foo: 2, 3
 foo: 3, 3
 bar: 2
 bar: 1
 bar: 2
 bar: 1

4. (10점) 아래의 프로그램을 실행하였을 때, 화면에 출력되는 결과를 작성하시오.

```
#include <stdio.h>

int bar(int n) {
    printf("bar: %d\n", n);
    switch (n) {
        case 0: return 3;
        case 1: return 1;
        default: return bar(n-1) + 5;
    }
}

int foo(int n) {
    int x;

    if (n==0)
        return 0;

    x = foo(n-1);
    printf("foo: %d %d\n", n, x);
    if ( bar(n-1) > x )
        return bar(n-1);
    return x;
}

int main(void) {
    printf("main: %d\n", foo(3));
    return 0;
}
```

5. (9점) 아래 프로그램을 완성하시오.

```

/*****
Test a function that rounds double values.
*****/

```

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

```
double round(double number, double level); /* prototype */
```

```
main()
```

```
{
```

```
    double input;
```

```
    double level;
```

```
    double result;
```

```
    printf("Round a number to a specified level.\n");
```

```
    printf("Please enter the number: ");
```

```
    scanf("%lf", &input);
```

```
    printf("Please enter the rounding level: ");
```

```
    scanf("%lf", &level);
```

```
    [ (a) ] printf("%f rounds to %f\n", input, result);
```

```
}
```

```

/*****

```

```
round() : Round a number to a specified level.
```

```
For example, if the number is 6784.9, and the level
```

```
is 10.0, the function will return the result 6780.0.
```

```
*****/
```

```
double round(double number, double level) /* Header */
```

```
{
```

```
    double rounded;
```

```
    if (number >= 0.0)
```

```
        rounded = [ (b) ];
```

```
    else
```

```
        rounded = [ (c) ];
```

```
    [ (d) ] return rounded;
```

```
}
```

6784.9 / Level

6784 / 10

정수 자리 일

6784.9

정수 자리 일

(6784 mod 10)

4

10.0

6784 / Level

6. (13점) 다음 프로그램은 1부터 100까지의 정수 중, 무작위로 임의의 수를 선택한 후, 표준입력으로부터 선택된 수만큼의 정수(integer) 데이터를 얻고, 이에 대한 기하평균(geometric mean)을 구하는 프로그램이다. n 개의 데이터에 대한 기하평균을 구하는 식은

$$\sqrt[n]{x_1 \times x_2 \times \dots \times x_n}$$

이며, 표준입력에서 각 정수 데이터는 ';'에 의해 구별된다. (Appendix 참고)

- 1) main()이 아래와 같이 정의되어 있을 때, 주석에 맞게 빈칸을 채우시오.

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

//표준입력에서 num 개의 정수 데이터의 기하평균을 구하는 함수.
double calGeometricMean( int num );

int main()
{
    int    dataCount;    //data 의 수
    double geoMean;      //geometric mean 값.
    // random 수 초기화 (time())를 이용한 초기화
    (a) (2 점)
    // 1 부터 100 에서 random 수 선택하여 dataCount 에 저장.
    (b) (2 점)
    printf("# of data : %d\n", dataCount);

    //기하평균계산
    geoMean = calGeometricMean( dataCount );
    printf( "Geometric Mean : %g\n", geoMean);
    return 1;
}
```

2) 위의 프로그램이 실행 가능하도록 기하평균을 구하기 위한 `calGeometricMean()`을 작성하시오. 참고로 위의 프로그램을 실행한 예는 다음과 같다.

```
# of data : 3
```

```
34;
```

```
59;
```

```
92;
```

```
Geometric Mean : 56.9342
```

```
double calGeometricMean( int num ) // (9 점)
```

```
{
```

(c)

```
}
```


7. (16점) 다음의 프로그램은 텍스트 파일 안에 있는 알파벳, 숫자, 공백, 구두점(punctuation, '!' 등) 문자의 개수를 분석하는 프로그램이다. 아래와 같이 main()함수가 정의되었을 경우, txtCount()을 정의하고 구현하시오. (Appendix 참고)

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

```
#include <ctype.h>
```

(a)

(2점)

```
int main( void )
```

```
{
```

```
    int totalCount = 0;    //전체 character의 개수
```

```
    int alphaCount = 0;    //알파벳의 개수
```

```
    int digitCount = 0;    //숫자의 개수
```

```
    int spaceCount = 0;    //스페이스의 개수
```

```
    int punctCount = 0;    //특수문자의 개수
```

```
    FILE *fp;              //파일 포인터
```

```
    if( !( fp = fopen("document.txt", "r")) ){ return 0; }
```

```
    //call txtCount()-----
```

(b)

(2점)

```
    //-----
```

```
    printf("Total number of characters : %d\n", totalCount);
```

```
    printf("The number of alphabets : %d\n", alphaCount);
```

```
    printf("The number of digits : %d\n", digitCount);
```

```
    printf("The number of spaces : %d\n", spaceCount);
```

```
    printf("The number of punctuations : %d\n", punctCount);
```

```
    fclose( fp );
```

```
    return 1;
```

```
}
```

(c) (12점)

2014

Pascal's triangle에서 n 번째 row의 r ($0 \leq r \leq n$) 번째 숫자는 $\binom{n}{r}$, 즉 이항계수(binomial coefficient)이다. 이항계수는 다음과 같이 정의된다.

$$\binom{n}{0} = 1, \binom{n}{n} = 1, \binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}, \text{ for } 1 \leq r \leq n-1,$$

아래는 Pascal's triangle의 예제이다. (row의 개수가 5일 때, $n = 0, 1, 2, \dots, 4$)

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

이 프로그램은 row의 개수를 표준입력으로부터 읽어 들여서, row의 개수만큼 Pascal's triangle을 표준출력으로 인쇄한다.

프로그램을 작성할 때, 되도록 수업시간에 배운 기능만을 사용하고, 프로그램의 효율성에 대하여는 신경 쓰지 않아도 된다.

Appendix

Routine	Description	Usage
File stream I/O functions		
fopen	Open stream	FILE *fopen(const char *filename, const char *mode);
fclose	Close stream	int fclose(FILE *stream);
feof	Test for end of file on stream	int feof(FILE *stream);
fgetc	Read character from stream	int fgetc(FILE *stream);
fputc	Write a character to a stream	int fputc(int c, FILE *stream);
fgets	Read string from stream	char *fgets(char *string, int n, FILE *stream);
fputs	Write string to stream	int fputs(const char *string, FILE *stream);
fscanf	Read formatted data from stream	int fscanf(FILE *stream, const char *format [, argument]...);
fprintf	Write formatted data to stream	int fprintf(FILE *stream, const char *format [, argument]...);
fseek	Move file position to given location	int fseek(FILE *stream, long offset, int origin);
fread	Read unformatted data from stream	size_t fread(void *buffer, size_t size, size_t count, FILE *stream);
fwrite	Write unformatted data to stream	size_t fwrite(const void *buffer, size_t size, size_t count, FILE *stream);
rewind	Move file position to beginning of stream	void rewind(FILE *stream);
Character Classification functions		
isalpha	Test for alphabetic	int isalpha(int c);
isspace	Test for white-space	int isspace(int c);
isdigit	Test for decimal digit	int isspace (int c); <i>isdigit</i>
ispunct	Test for punctuation	int ispunct(int c);
Math functions		
sqrt	Find square root	double sqrt(double x);
pow	Calculate value raised to a power	double pow(double x, double y);
log	Calculate natural logarithm	double log(double x);
log10	Calculated base-10 logarithm	double log10(double x);