

기초 프로그래밍 실습

11월 1주차



학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2016년 11월 4일

8장 실습 문제 1번, 2번

- 1. 월급에 붙는 소득세를 계산하는 함수 `get_tax(int income)`를 작성하고 테스트하여 보자. 과표 구간은 1000만원 이하 8%, 1000만원 초과는 10%로 되어있다고 가정한다. 사용자로부터 소득을 입력 받아서 세금을 계산하는 프로그램을 작성하라.
- 2. 난수(random number)는 컴퓨터를 이용한 문제 해결에서 많이 사용된다. 특히 수학적인 분석이 너무 복잡한 경우에 시뮬레이션을 사용하면 실제로 제품을 제작하지 않고서도 많은 실험을 할 수 있다. Visual Studio의 경우, `rand()`가 한번 호출될 때마다 0에서 32767까지의 정수를 같은 확률로 선택하여 반환한다. `Rand()` 함수를 이용하여 0 또는 1 값을 무작위로 반환하는 함수 `b_rand()`를 작성하고 5번 호출하여 보자.

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

int get_tax (int income) {
    int over = income - 10000000;
    if(over < 0) over = 0;
    int lower = income % 10000000;
    return over * 0.08 + lower * 0.1;
}

int main()
{
    int income;
    printf("수입을 입력하세요 . ");
    scanf("%d", &income);
    printf("당신의 세금은 % d입니다 .", get_tax (income));
}
```

```
----- 문제 3번 실행을 종료합니다 . -----
----- 문제 1번 실행을 시작합니다 . -----
./1.x
수입을 입력하세요 . 230012000
당신의 세금은 17602160입니다 .----- 문제 1번 실행을 종료합니다 . -----
-----
```

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

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

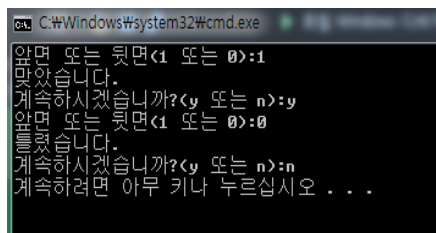
int b_rand() {
    return rand() % 2;
}

int main()
{
    srand(time(NULL));
    for(int i=0; i<5; i++) {
        printf("%d ", b_rand());
    }
}
```

```
gcc 2.c -o 2.x -g -fmax-errors=1 -lm
----- 문제 2번 실행을 시작합니다 . -----
./2.x
1 1 1 1 0 ----- 문제 2번 실행을 종료합니다 . -----
```

8장 실습 문제 3번

- 3. 앞에서 작성한 `b_rand()` 함수를 이용하여 간단한 동전 던지기 게임을 시뮬레이션 하여 보자. 컴퓨터가 동전을 던지고 사용자는 앞뒤를 말한다. 컴퓨터는 `b_rand()`를 이용하여 생성된 난수가 1이면 동전의 앞면으로 간주하고 0이면 동전의 뒷면으로 간주한다.



```
C:\Windows\system32\cmd.exe
앞면 또는 뒷면(1 또는 0):1
맞았습니다.
계속하시겠습니까?(y 또는 n):y
앞면 또는 뒷면(1 또는 0):0
틀렸습니다.
계속하시겠습니까?(y 또는 n):n
계속하려면 아무 키나 누르십시오 . . .
```

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

int b_rand() {
    srand(time(NULL));
    return rand() % 2;
}
```

```

int main()
{
    int side;
    char yesno[2];
here:
    printf(" 앞면 또는 뒷면 <1 또는 0>?");
    scanf("%d", &side);
    if(b_rand() == side) printf(" 맞았습니다 .\n");
    else printf(" 틀렸습니다 .\n");
    printf(" 계속하시겠습니까 ?( y또는 n)");
    scanf("%s", &yesno);
    if(yesno[0] == 'y') goto here;
}

```

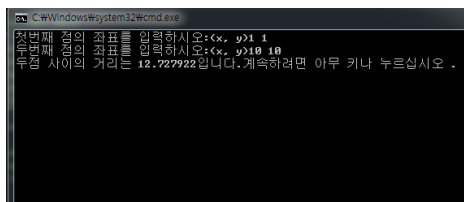
```

----- 문제 3번 실행을 시작합니다 . -----
./3.x
앞면 또는 뒷면 <1 또는 0>?1
맞았습니다 .
계속하시겠습니까?(y또는 n)y
앞면 또는 뒷면 <1 또는 0>?0
맞았습니다 .
계속하시겠습니까?(y또는 n)y
앞면 또는 뒷면 <1 또는 0>?0
틀렸습니다 .
계속하시겠습니까?(y또는 n)n
----- 문제 3번 실행을 종료합니다 . -----

```

8장 실습 문제 4번

- 4. 두 점 사이의 거리를 계산하는 함수를 작성하여 보자. 2차원 공간에서 두 점 (x1,y1)와 (x2,y2) 사이의 거리를 계산하는 `get_distance(double x1, double y1, double x2, double y2)`를 작성하라. 다음과 같은 두 점 사이의 거리를 계산하는공식을 사용하라. 제공근은 `sqrt()`라이브러리 함수를 사용하라



```

C:\Windows\system32\cmd.exe
첫번째 점의 좌표를 입력하십시오:<x, y>1 1
두번째 점의 좌표를 입력하십시오:<x, y>10 10
두점 사이의 거리는 12.727922입니다. 계속하려면 아무 키나 누르십시오 . . .

```

```

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

```

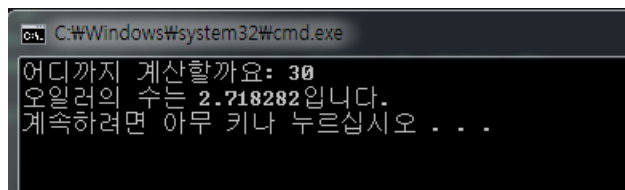
```
double distance(double x1, double y1, double x2, double y2) {
    return sqrt ((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

int main()
{
    int x1, y1, x2, y2;
    printf(" 첫번째 점의 좌표를 입력하세요 .(x, y)");
    scanf("%d %d", &x1, &y1);
    printf(" 두번째 점의 좌표를 입력하세요 .(x, y)");
    scanf("%d %d", &x2, &y2);
    printf(" 두 점 사이의 거리는 % lf입니다 .\n", distance(x1, y1, x2, y2));
}
```

```
----- 문제 4번 실행을 시작합니다 . -----
./4.x
첫번째 점의 좌표를 입력하세요 .(x, y)1 1
두번째 점의 좌표를 입력하세요 .(x, y)10 10
두 점 사이의 거리는 12.727922입니다 .
----- 문제 4번 실행을 종료합니다 . -----
```

8장 실습 문제 5번

- 5. 오일러 수 e 는 자연로그의 밑수로 사용된다. 이 값은 다음과 같은 식에 의하여 근사치를 구할 수 있다. 본문에 있는 팩토리얼 값을 계산하는 함수 `factorial()`을 호출하여서 오일러 수를 계산하는 프로그램을 작성하라.



```
C:\Windows\system32\cmd.exe
어디까지 계산할까요: 30
오일러의 수는 2.718282입니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
#include<stdio.h>

int factorial (int n) {
    if(n == 1) return 1;
    return n * factorial (n-1);
}
```

```

}

int main()
{
    int n;
    printf(" 몇 번째까지 계산할까요 ?");
    scanf("%d", &n);
    double e = 0;
    for(int i=1; i<=n; i++) e += 1.0 / factorial (i);
    printf(" 오일러수는 % lf입니다 .\n", e+1);
}

```

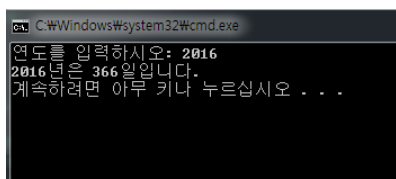
```

zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
----- 문제 5번 실행을 시작합니다 . -----
./5.x
몇 번째까지 계산할까요?30
오일러수는 2.718282입니다 .
----- 문제 5번 실행을 종료합니다 . -----

```

8장 실습 문제 6번

- 6.우리는 앞에서 윤년을 구하는 알고리즘을 학습하였다. 이것을 함수 `is_leap(int year)` 함수로 작성하고 이 함수를 사용하여 1년이 몇 일인지를 출력하는 프로그램을 작성하여 보자.
 - 윤년은 4의 배수지만 100의 배수는 제외하고 400의 배수는 무조건 추가하면 구할 수 있다. 윤년이면 366이고 평년이면 365일이 된다.



```

C:\Windows\system32\cmd.exe
연도를 입력하십시오: 2016
2016년은 366일입니다.
계속하려면 아무 키나 누르십시오 . . .

```

```

#include<stdio.h>

int is_reap (int y) {
    if(y % 400 == 0) return 1;
    if(y % 100 == 0) return 0;
    if(y % 4 == 0) return 1;
    return 0;
}

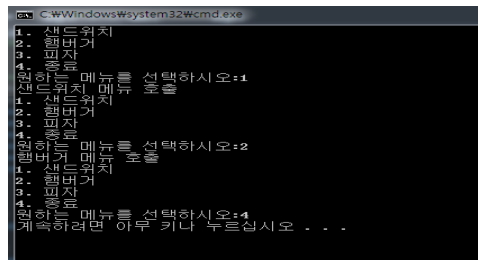
```

```
int main()
{
    int y;
    printf(" 연도를      입력하시오 .");
    scanf("%d", &y);
    printf("%d년은 %d일입니다 .", y, 365+is_reap(y));
}
```

```
----- 문제 4번 실행을 종료합니다 . -----
----- 문제 6번 실행을 시작합니다 . -----
./6.x
연도를 입력하시오 .2016
2016년은 366일입니다 .----- 문제 6번 실행을 종료합니다 . -----
-----
```

8장 실습 문제 7번

- 7. 간단한 메뉴 시스템을 작성하여 보자. 다음과 같은 메뉴를 선택하고 사용자로부터 입력을 받아서 입력이 올바른지를 검사한다. 메뉴를 실제로 실행할 필요는 없다. 다만 메뉴의 이름을 화면에 출력하고 다시 반복한다.
 - (a) 메뉴를 화면에 출력하고 원하는 메뉴를 선택하라는 메시지를 내보내는 함수 `print_menu()`를 작성하라
 - 사용자로부터 하나의 정수를 읽은 후에 메뉴 번호의 상한과 하한을 넘지 않았는지를 검사하는 함수 `check_menu_number()`를 작성한다. 만약 사용자로부터 입력된 정수가 상한이나 하한을 벗어나면 `print_manu()`를 호출하여 화면에 메뉴를 표시한다. 범위 안에 드는 정수가 입력되면 메뉴를 처리하는 함수를 호출한다.
 - 각각의 메뉴를 처리하는 함수를 작성한다. 메뉴에 해당되는 함수가 호출되면 메뉴가 호출되었다는 메시지만 출력하고 종료한다.



```
C:\Windows\system32\cmd.exe
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하시오:1
1. 샌드위치 메뉴 호출
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하시오:2
2. 햄버거 메뉴 호출
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하시오:3
3. 피자 메뉴 호출
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하시오:4
4. 종료 메뉴 호출
원하는 메뉴를 선택하시오:4
계속하려면 아무 키나 누르십시오 . . .
```

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

```
char menus[4][20] = {" 샌드위치 ", " 햄버거 ", " 피자 ", " 종료 "};
```

```
void print_menu() {
    for(int i=1; i<5; i++) printf("%d. %s\n", i, menus[i-1]);
    printf(" 원하는 메뉴를      선택하시오 .");
}
```

```
void call_menu(int n) {
    printf("%s 메뉴      호출 \n", menus[n]);
}
```

```
int check_menu_number(int n) {
```

```

    if(n > 0 && n < 4) return 1;
    else return 0;
}

int main()
{
    int n = -1;
    while(n != 4) {
        if(check_menu_number(n)) call_menu(n-1);
        print_menu();
        scanf("%d", &n);
    }
}

```

```

3. 피자
4. 종료
원하는 메뉴를 선택하십시오 .1
샌드위치 메뉴 호출
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하십시오 .2
햄버거 메뉴 호출
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하십시오 .6
1. 샌드위치
2. 햄버거
3. 피자
4. 종료
원하는 메뉴를 선택하십시오 .4
----- 문제 7번 실행을 종료합니다 . -----

```



연결 해제 - 오프라인 상태
무선 네트워크