

CSED101. Programming & Problem solving

Spring, 2014

Programming Assignment #2 (60 points)

김용현 (gkyh0805@postech.ac.kr)

- **Due:** 2014. 04. 15 23:59
- **Development Environment:** Windows Visual Studio 2010
- **제출물**
 - **C code file (assn2.c)**
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙일 것.
 - **보고서 파일** (.doc(x) or .hwp) 예) assn2.doc(x) 또는 assn2.hwp
 - AssnReadMe.pdf 를 참조하여 작성할 것.
 - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.
- **주의사항**
 - 각 문제에 해당하는 요구사항을 반드시 지킬 것.
 - 컴파일 & 실행이 안되면 무조건 0점 처리된다.
 - **하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)**
 - 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
 - **assn2.c**로 제출할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
 - 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

■ Problem: 숫자 게임

(목적)

이 숙제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

(문제)

숫자 게임 프로그램은 암산 게임의 일종으로 제한시간 동안 제공되는 사칙연산(덧셈, 뺄셈, 곱셈, 나눗셈)을 풀고 최고 점수를 달성하는 게임이다. 아래의 설명을 바탕으로 '숫자게임'을 작성해 보도록 하자. 단, 매 실행마다 다른 문제가 생성되도록 작성한다.

전역 변수는 사용할 수 없으며, 아직 수업시간에 다루지 않은 배열, 구조체를 사용하지 않고 프로그램을 작성한다.

(설명)

문제를 해결하기 전, 문제와 함께 제공된 assn2_game.exe를 실행하여 게임을 즐겨보도록(?) 하자. 게임을 실행하면, 빈칸으로 처리 된 수식과 빈칸에 대한 정답 후보 2개가 주어진다. 사용자는 '[', ']' 키를 통해 좌, 우측의 정답을 선택한다. 정답을 선택 한 경우 Score가 1점 추가되고 틀릴 경우 남은 시간이 2초 줄어든다.

프로그램을 실행하면 아래의 그림과 같이 '숫자 게임'이 실행된다.

```
C:\WINDOWS\system32\cmd.exe
-----
[Best Score: 0]
                [Score: 0]
-----
Next   : [ ]   -   4   =   2
Current: 17   - [ ]   =   9
-----
        3   <   =   >   8
-----

[Time : 28.80]
```

수식을 출력하기 위해 수식을 생성하는 방법은 아래와 같다. 먼저 프로그램은 내부적으로 완전한 수식을 생성한다. 수식은 아래와 같이 숫자 3개와 연산자 하나로 구성된다. **단, 아래의 숫자 3개는 모두 자연수라고 가정한다.**

생성된 수식의 예)

$17 - 8 = 9$

- 왼쪽 피연산자 (Left Operand): 17
- 연산자 (Operator): -
- 오른쪽 피연산자 (Right Operand): 8
- 연산결과: 9

➤ 수식 생성 및 문제와 정답 후보 출력 방법

1. 연산자 선택

프로그램에서 사용되는 연산자는 사칙연산자(+, -, *, /) 중 하나가 랜덤하게 선택된다.

2. 완전한 수식 생성 (내부 수식)

선택된 연산자를 이용하여 내부 수식을 생성한다.

생성될 숫자의 범위: 1 ~ (9 + Score)

사용자가 문제를 맞출 때마다 문제의 난이도를 높이기 위해서 랜덤하게 선택될 숫자의 범위를 1씩 증가하도록 한다. 즉, Score는 0으로 시작해서 사용자가 문제를 맞출 때마다 1점씩 증가하고 랜덤하게 선택될 숫자의 범위도 1씩 증가한다.

- 덧셈, 곱셈

왼쪽 피연산자, 오른쪽 피연산자의 숫자를 제시된 범위 안에서 랜덤하게 선택하여 수식 완성

- 뺄셈, 나눗셈

오른쪽 피연산자, 연산결과의 숫자를 제시된 범위 안에서 랜덤하게 선택하여 수식 완성
위의 생성된 수식으로 설명을 하면, 오른쪽 피연산자가 8, 연산 결과가 9로 선택되면, 왼쪽 피연산자의 값은 17의 값이 되도록 수식을 완성한다.

3. 숨겨질 숫자의 위치 선택

왼쪽 피연산자, 오른쪽 피연산자, 연산결과 중에서 숨겨질 숫자의 위치를 랜덤하게 선택하도록 한다. 숨겨질 숫자가 사용자가 맞춰야 할 정답에 해당한다.

4. 사용자가 선택할 정답 1개의 후보 생성

오답인 숫자를 랜덤하게 생성한다. 생성될 숫자의 범위는 위와 동일하다.

선택된 오답이 정답과 같은 값이 나오지 않도록 한다.

5. 정답을 포함하여 2개의 정답 후보의 출력 순서를 랜덤하게 결정한다.

위의 과정이 끝나면, 아래와 같이 화면에 빈칸 있는 수식과 함께 2개의 정답 후보가 출력이 된다.

화면 출력 예)

$17 - [] = 9$
$3 <=> 8$

사용자는 '[,]' 키를 통해 좌, 우측의 정답을 선택한다. 정답을 선택 한 경우 Score가 **1점** 추가되고, 오답을 선택한 경우 남은 시간이 **2초** 줄어든다. 그리고, 다음 문제가 출력된다.

사용자의 정답 입력 중, '[,]' 키 외의 입력은 없다고 가정한다.

게임 중 남은 시간의 실시간 출력

- [Time : xx.xx]: 밀리 초까지 출력되게 작성하며 키 입력으로 인해 시간 출력이 멈추지 않도록 한다.
- 남은 시간 계산을 위해 아래의 'Tips'에서 설명하고 있는 **clock()** 함수를 이용하도록 한다.
- 남은 시간을 실시간을 출력하기 위해 형식문자열 'Wr'을 출력문에 사용하도록 한다.
'Wr'을 출력할 경우 출력화면의 커서가 해당 라인의 맨 앞으로 이동하게 된다. 이를 이용해 남은 시간을 실시간으로 출력할 수 있다. 아래의 사용예제를 참고하시오.
- 사용 예제

```
printf("Hello World");  
printf("WrKorea");
```

⇒ 실행결과: Korea World

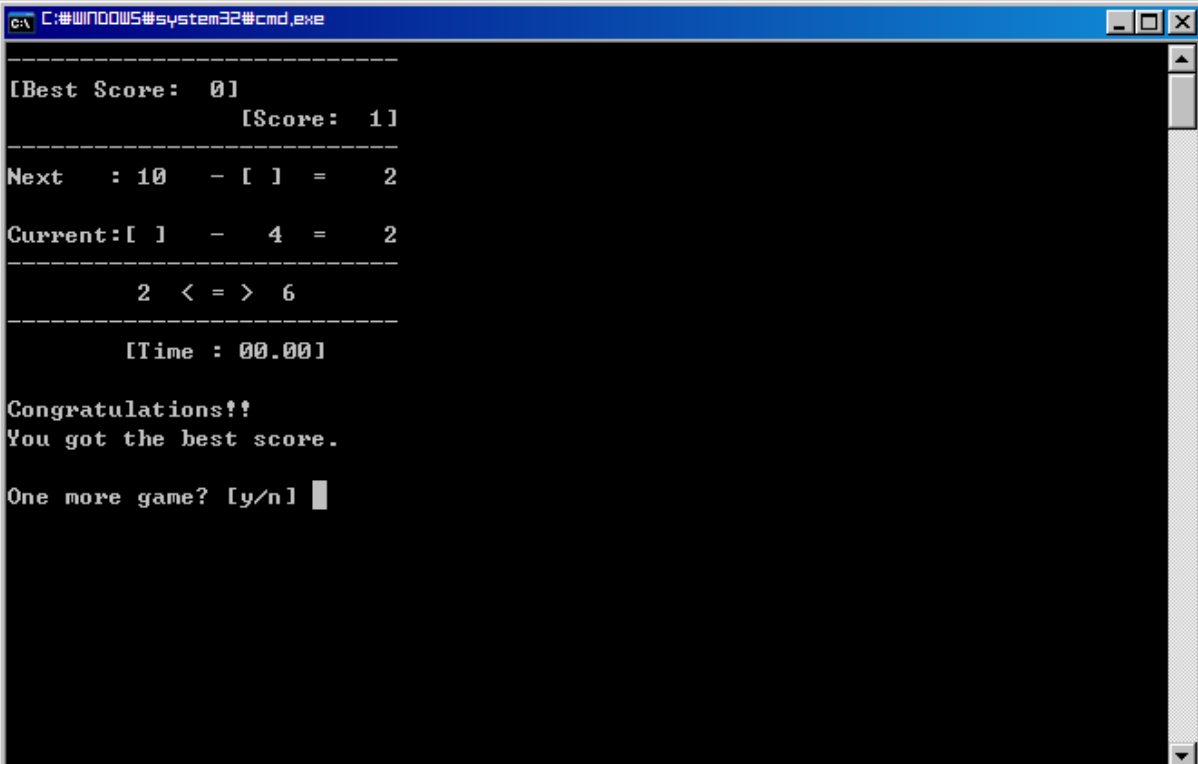
(주의사항) 시간 출력을 위해 전체화면을 지우는 system("cls")를 사용하지 말 것. (이를 사용하게 되면 전체화면이 깜빡이게 됨.)

게임 종료

제한 시간이 경과하면 게임이 끝난다. 이 게임에서의 제한 시간은 30초라고 가정한다.

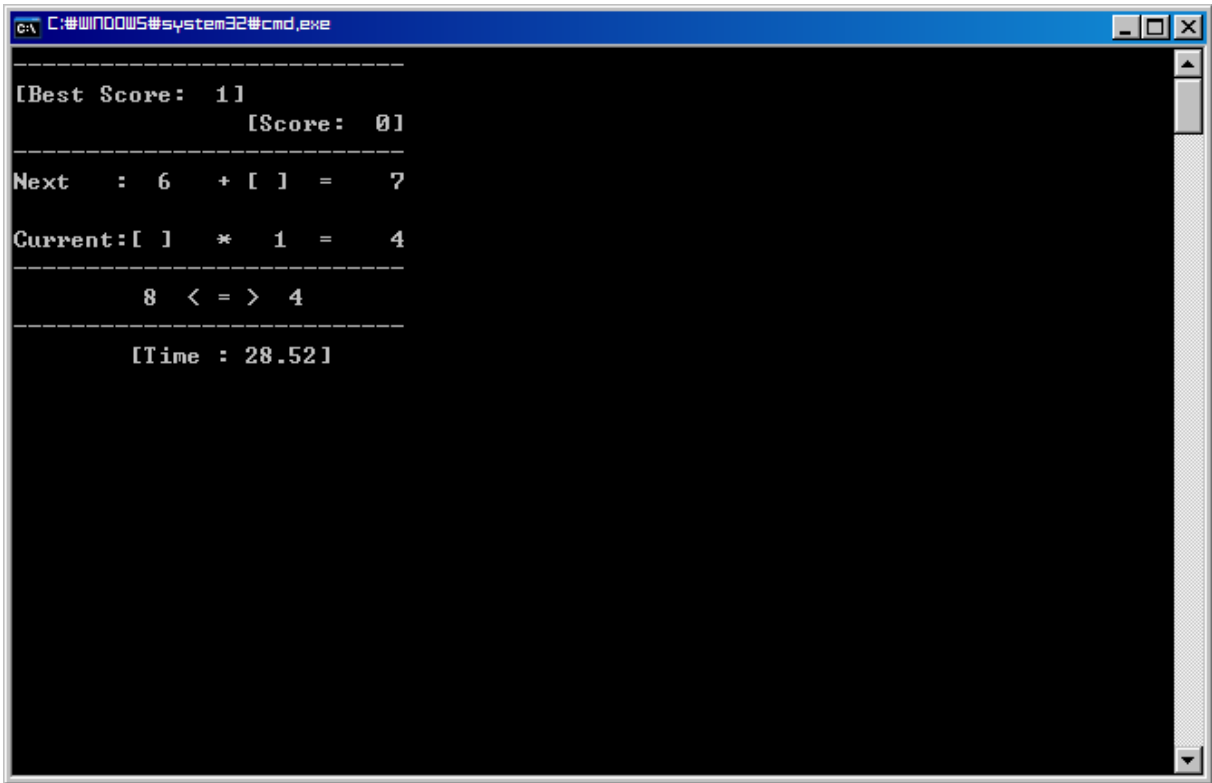
게임이 끝나면, 현재 점수와 Best Score를 비교해서 현재 점수가 Best Score 보다 더 크면 "Congratulations!! You got the best score."를 메시지를 출력한다. (그렇지 않은 경우 메시지가 출력되지 않는다.) 그 후에, "One more game? [y/n]" 메시지가 출력되고 사용자의 입력을 기다린다. 사용자가 'y'를 선택하면 다시 게임이 시작되고 Best Score가 갱신되며, 'n'을 선택하면 게임이 종료된다. ('y', 'n'을 제외한 다른 문자 입력에 대해서는 고려하지 않아도 된다.)

아래는 제한 시간이 경과해서 게임이 종료 됐을 때의 출력화면에 해당한다.



```
C:\#WINDOWS\system32#cmd.exe  
-----  
[Best Score: 0]  
                [Score: 1]  
-----  
Next   : 10   - [ ] = 2  
Current:[ ]   - 4   = 2  
-----  
                2 < = > 6  
-----  
                [Time : 00.00]  
  
Congratulations!!  
You got the best score.  
  
One more game? [y/n] █
```

아래는 위의 화면에서 사용자 입력을 'y'로 받은 경우에 해당한다. Best Score가 갱신되고 다시 게임이 시작된다.



➤ 사용자 정의 함수

반드시 아래의 함수를 정의해서 사용해야 하며, 아래 함수 외의 필요한 함수는 정의해서 사용할 수 있다.

- **getLetfTime**: 남은 시간을 리턴하는 함수
- **printEquation**: 수식을 출력하는 함수
 수식 출력시 **아래의 출력문을 사용 (아래의 출력문을 복사해서 사용하세요.)**
 - 왼쪽 피연산자 빈칸 출력 : `printf("[] %3c %3d = %3dWn", ...);`
 - 오른쪽 피연산자 빈칸 출력: `printf("%3d %3c [] = %3dWn", ...);`
 - 연산결과 빈칸 출력 : `printf("%3d %3c %3d = []Wn", ...);`
- **getOperatorToChar**: 수식을 구성하는 연산자는 +, *, -, / 순서대로 0, 1, 2, 3의 정수(int)에 대응하여 저장 및 사용 한다. 이 때 연산자 번호를 함수의 입력으로 전달 받아 그 번호에 대응되는 문자(char) 즉, '+', '*', '-', '/' 중 하나를 리턴 하는 함수이다. 수식 출력 등에 사용할 수 있다.

➤ 그 외의 출력문(복사해서 사용하세요.)

```
printf("-----Wn");
```

➤ 문제의 출력 형식은 채점을 위해 위쪽의 실행 예시와 최대한 비슷하게 작성해 주세요.

➤ **Tips**

• **남은 시간 계산**

clock_t clock(void): 프로그램 실행 후 경과된 시간을 리턴하는 함수

- 이 함수는 시스템으로부터 현재 시간을 millisecond 단위로 가져오며 1초는 1000 밀리세컨드가 된다. 경과된 시간은 clock()함수를 이용하여 현재 시간과 이전 시간의 차로 계산한다.
- <time.h>에 포함되어 있음
- clock_t 형은 long 형으로 time.h 에 정의되어 있다.

• **화면 지우기**

int system(char*): 명령프롬프트(cmd)상에 정의되어 있는 명령어(dir, cls 등)를 실행하는 함수

- <stdlib.h>에 포함되어 있음
- 사용 예제

```
#include <stdlib.h>

int main()
{
    printf("Hello World\n");
    system("cls"); // 화면에 Hello World 출력 후, 화면을 지움
    ...
}
```

• **대기 없이 키보드 입력**

scanf를 사용하여 키보드 입력을 받게 되면 입력 라인에서 프로그램이 멈추게 된다. 본 프로그램은 게임의 형식을 취하고 있기 때문에 실시간 입력이 가능해야 한다. 아래의 두 함수를 응용하여 키보드 입력을 받을 수 있다.

int kbhit(void): 키 입력을 체크하는 함수

- 키 입력이 있으면 0 이외의 값을, 아니면 0을 리턴 한다.
- <conio.h>에 포함되어 있음

int getch(void): 화면에 출력 없이 키보드로부터 하나의 문자를 입력 받는 함수

- 엔터를 누르지 않고도 키보드 입력을 받을 수 있고, 누른 키가 화면에 출력되지 않기 때문에 게임을 엔터키 없이 진행할 수 있다.
- 누른 키보드의 값을 리턴한다.
- <conio.h>에 포함되어 있음