

CSED101. Programming & Problem solving

Spring, 2017

Programming Assignment #2 (60 points)

이원기(wklee@postech.ac.kr)

■ **Due:** 2017.04.04 23:59

■ **Development Environment:** Windows Visual Studio 2015

■ 제출물

- **C Code files (*.c)**
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- **보고서 파일** (.doc(x) or .hwp) 예) assn2.doc(x) 또는 assn2.hwp
 - AssnReadMe.pdf 를 참조하여 작성할 것.
 - 프로그램 실행 화면을 캡처하여 보고서에 포함시키고 간단히 설명할 것.
 - 명예서약(Honor code): 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
 - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
- 각 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
- 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.)
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

■ Problem 1: 숫자 맞추기 게임 (30점)

(목적)

본 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

(주의사항)

- 파일이름은 확장자를 포함하여 "assn2_1.c"로 할 것
- 보고서는 "assn2.doc" or "assn2.hwp"로 저장 할 것 (보고서는 통합하여 작성)
- 본 과제는 사용자 함수를 정의하고 사용하는 방법을 익히는 문제이므로 사용자 정의 함수를 사용하지 않고, main함수에 모든 기능을 구현할 경우 감점 처리함. (반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다. 그 외에 필요한 함수는 추가적으로 정의해서 사용할 수 있다.)
- 전역변수, 배열 및 goto 문은 사용할 수 없음
- 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

(문제)

컴퓨터가 생성한 임의의 4 자리 숫자(1000~9999)를 사용자가 맞추는 간단한 게임이다.

(설명)

1. 프로그램을 실행하면, 컴퓨터가 임의의 4 자리 숫자를 생성한 후, 아래와 같이 사용자 입력을 기다린다.
 - 컴퓨터가 생성한 임의의 4 자리 숫자는 프로그램을 실행할 때마다 다른 숫자가 생성되어야 한다.
 - 각 자리의 숫자는 중복 되어도 무관하다.
 - 사용자 정의 함수
 - *init_num*: 게임 시작 시, 임의의 4 자리 숫자를 생성해주는 함수



그림1. 프로그램 초기 화면

2. 사용자는 컴퓨터가 생성한 숫자를 맞추기 위해 4 자리 숫자를 추측해서 입력한다.
 - 사용자로부터 정수만 입력 받는 것으로 가정한다. (가령 123c 와 같이 문자와 숫자가 섞인 입력은 고려하지 않는다.)
 - 입력 범위(1000~9999 또는 0) 밖의 정수가 입력되는 경우, Trial 을 하나 증가시키고 다시 사용자 입력을 받도록 한다

(실행예제의 빨간색 밑줄은 사용자 입력을 말한다.)

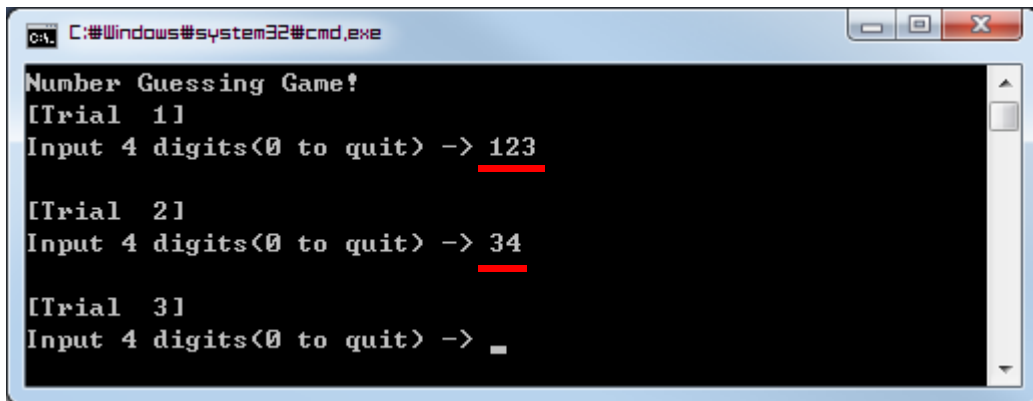


그림 2. 입력 범위 밖의 숫자가 입력된 경우

3. 게임을 위한 규칙은 다음과 같다.

각 자리 숫자에 대하여,

- 사용자가 추측한 숫자와 정답이 일치하면 그 숫자를 출력한다.
- 그렇지 않을 경우에는 입력한 숫자와 정답 숫자 2 개를 제외한 임의의 한 수를 랜덤하게 발생시켜 출력한다.

설명을 위해, 컴퓨터가 생성한 숫자가 8424 라고 가정하자.

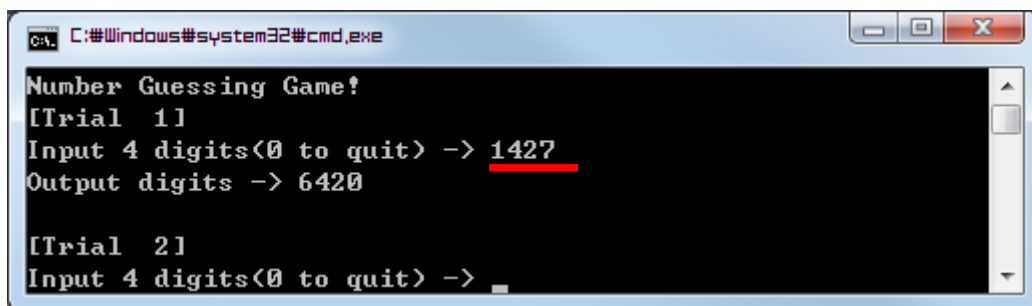


그림 3. 사용자가 추측한 숫자와 정답이 일치하지 않는 경우

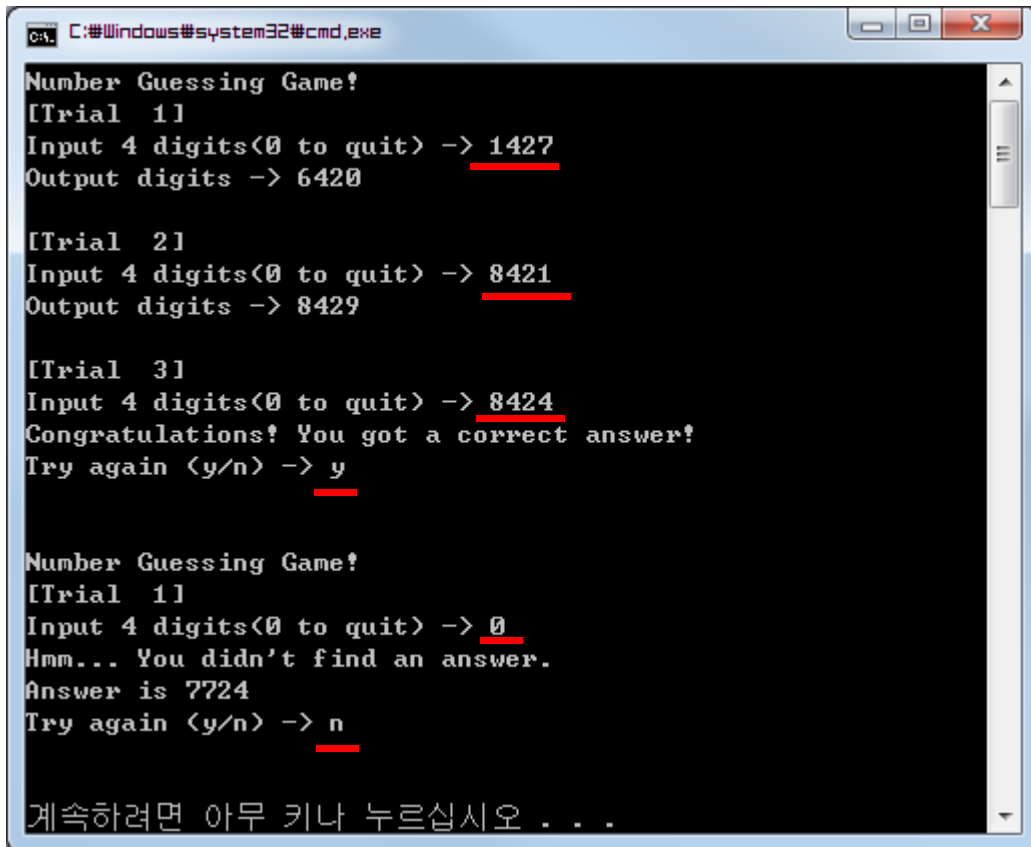
위 예제에서 사용자가 1427 을 입력했고, 규칙에 따라 첫 자리부터 비교하면, 사용자의 첫 자리는 1 이고 정답의 첫 자리는 8 로 일치하지 않으므로 1 과 8 을 제외한 임의의 수를 출력한다. 둘째 자리와 셋째 자리는 사용자입력과 정답이 일치하므로 그대로 출력된다. 넷째 자리는 정답과 일치하지 않으므로 사용자 숫자 7 과 정답 숫자 4 를 제외한 임의의 수를 출력한다. (위 예제에서 컴퓨터는 규칙에 따라 6420 을 출력했다.)

사용자 정의 함수

- **is_correct.** 사용자 입력과 컴퓨터가 생성한 숫자가 일치하는지 판별해주는 함수
- **remake_num.** 사용자 입력과 컴퓨터가 생성한 숫자를 비교하여 규칙에 맞게 숫자를 생성하는 함수

4. 규칙에 따라 임의의 숫자를 출력한 후, 사용자 입력을 받기 위해 기다린다.

이렇게 정답을 맞추기 위한 사용자의 입력(4 자리 숫자)이 있을 때마다, 게임 규칙에 따른 4 자리 숫자를 출력한다.



```
C:\Windows\system32\cmd.exe
Number Guessing Game!
[Trial 1]
Input 4 digits(0 to quit) -> 1427
Output digits -> 6420

[Trial 2]
Input 4 digits(0 to quit) -> 8421
Output digits -> 8429

[Trial 3]
Input 4 digits(0 to quit) -> 8424
Congratulations! You got a correct answer!
Try again (y/n) -> y

Number Guessing Game!
[Trial 1]
Input 4 digits(0 to quit) -> 0
Hmm... You didn't find an answer.
Answer is 7724
Try again (y/n) -> n

계속하려면 아무 키나 누르십시오 . . .
```

그림 4. 게임 실행 화면

5. 게임의 종료

- (1) 4 자리를 모두 맞히면 축하 메시지를 표시한다.

위의 실행예시는 3 번의 시도 만에 정답을 맞힌 경우로, 축하 메시지 표시 후, 다시 게임을 시작할 것인지 물어본다.

- (2) 0 을 입력하면 정답 맞히기를 포기한다.

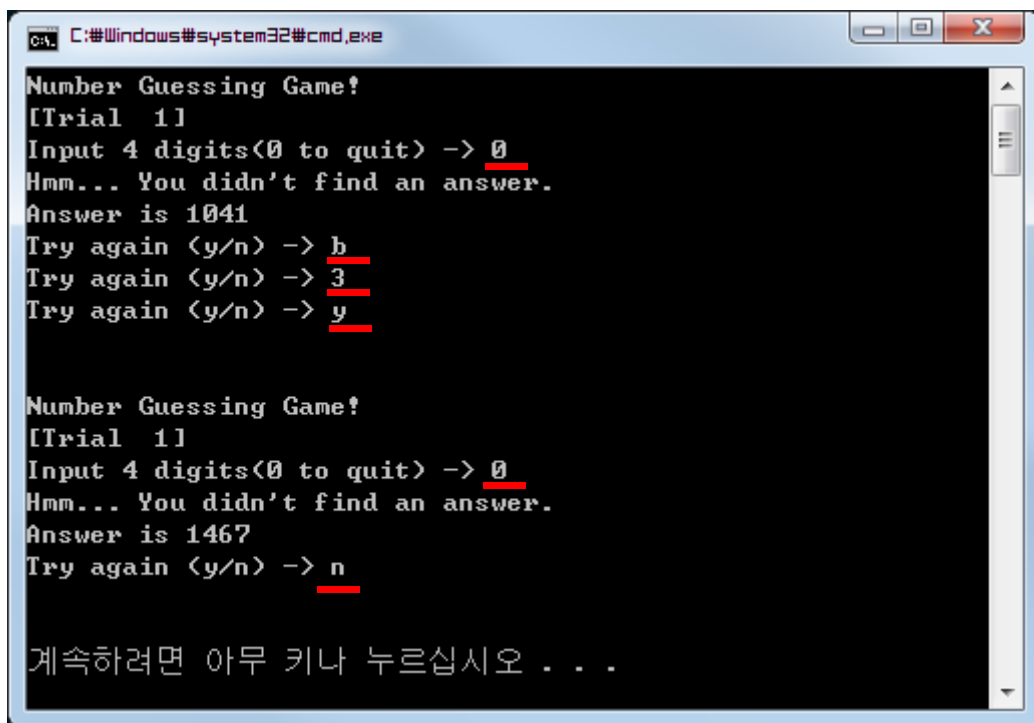
정답 맞히기를 위한 사용자 입력에 0 을 입력하면, 게임이 종료된다. 메시지와 함께 정답 숫자를 화면에 표기한 후, 다시 게임을 시작할 것인지 물어본다.

- (3) 다시 게임을 진행할 것인지 물어보는 질문에 y 또는 Y 를 입력한 경우, 게임을 다시 시작하고, n 또는 N 을 입력한 경우, 프로그램이 종료된다.

- 사용자 정의 함수

- **is_restart()**: 게임을 재 시작 할 지 여부를 물어보는 함수

예외처리) 게임 재 시작 여부에 대한 사용자 입력은 y, Y, n, N 4 가지(대소문자)가 가능하며, 그 외의 입력이 있을 경우 [그림 5]의 실행 예제처럼 다시 입력을 받도록 예외 처리를 한다.



```
C:\Windows\system32\cmd.exe

Number Guessing Game?
[Trial 1]
Input 4 digits(0 to quit) -> 0
Hmm... You didn't find an answer.
Answer is 1041
Try again (y/n) -> 0
Try again (y/n) -> 3
Try again (y/n) -> y

Number Guessing Game?
[Trial 1]
Input 4 digits(0 to quit) -> 0
Hmm... You didn't find an answer.
Answer is 1467
Try again (y/n) -> n

계속하려면 아무 키나 누르십시오 . . .
```

그림 5. 재 시작 여부에 대한 예외처리

■ Problem 2: 볼록 다각형 판별 및 넓이 구하기 (30점)

(목적)

본 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

(주의사항)

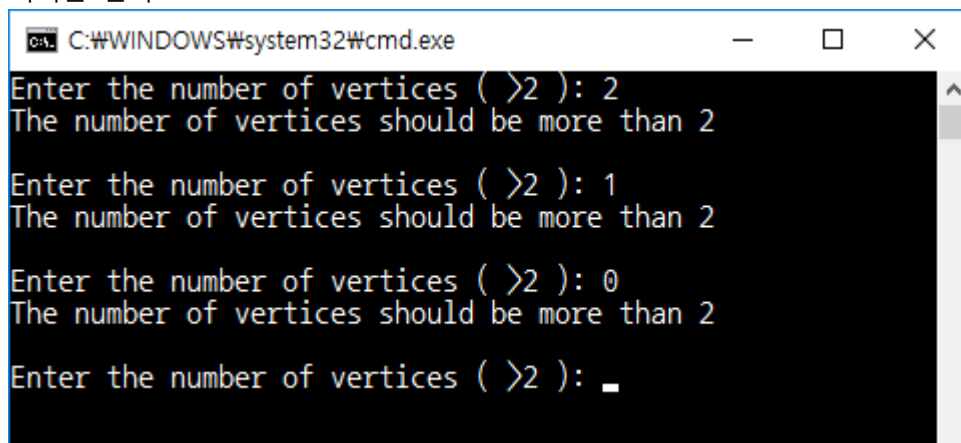
- 파일이름은 확장자를 포함하여 "assn2_2.c"로 할 것
- 보고서는 "assn2.doc" or "assn2.hwp"로 저장 할 것 (보고서는 통합하여 작성)
- 본 과제는 사용자 함수를 정의하고 사용하는 방법을 익히는 문제이므로 사용자 정의 함수를 사용하지 않고, main함수에 모든 기능을 구현할 경우 감점 처리함. (반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다. 그 외에 필요한 함수는 추가적으로 정의해서 사용할 수 있다.)
- 전역변수, 배열 및 goto 문은 사용할 수 없음
- 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

(문제)

사용자로부터 n 각형의 좌표를 입력 받아서, 그 좌표로 이루어진 n 각형이 볼록 다각형인지 판별하고, 볼록 다각형인 경우 n 각형의 넓이를 계산하여 출력한다.

(설명)

1. 사용자로부터 n각형의 숫자를 입력 받는다. (범위: 3-100각형)
 - 다각형 크기 입력 시, 2이하의 숫자가 입력될 경우 아래와 같이 입력을 다시 받도록 예외 처리를 한다.



```
C:\WINDOWS\system32\cmd.exe
Enter the number of vertices (>2): 2
The number of vertices should be more than 2

Enter the number of vertices (>2): 1
The number of vertices should be more than 2

Enter the number of vertices (>2): 0
The number of vertices should be more than 2

Enter the number of vertices (>2): _
```

그림 1. 입력 크기에 대한 예외처리

2. 사용자로부터 n개의 다각형 좌표를 순서대로 입력 받는다.
 - 좌표 입력 시, x, y 값의 구분은 스페이스로 한다.
 - 연속된 세 점이 일직선 상에 있는 좌표의 입력은 고려하지 않는다.

3. 입력 받은 n 개의 좌표를 입력순서대로 [그림2] 와 같이 반 시계방향으로 검사하면서 볼록 다각형인지 판별한다.
4. n 개의 좌표가 볼록 다각형을 이룰 경우 다각형 넓이를 계산하여 출력한다.
5. n 개의 좌표를 순차적으로 검사하는 중, 오목다각형으로 판단되는 즉시 메시지를 출력하고 프로그램을 종료한다.

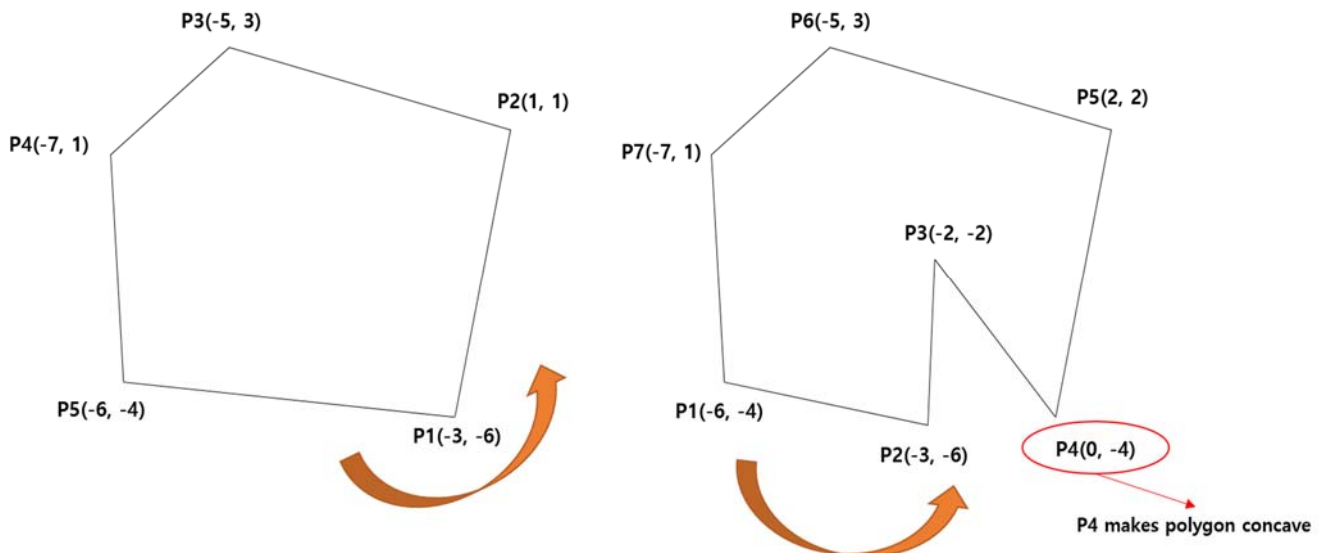


그림 2. n 개 좌표가 입력 순으로 다각형의 볼록 여부를 확인

(프로그램 실행 예시)

1. n 각형이 볼록으로 판별되는 경우, n 각형의 넓이를 계산한다.

```

C:\WINDOWS\system32\cmd.exe
Enter the number of vertices (>2): 5
Input the value x and y of P1: -3 -6
Input the value x and y of P2: 1 1
Input the value x and y of P3: -5 3
Input the value x and y of P4: -7 1
Input the value x and y of P5: -6 -4
Area of 5-polygon(convex) : 42.50
계속하려면 아무 키나 누르십시오 . . .
  
```

그림 1. 볼록 5각형이 입력된 경우

2. n 각형의 입력 좌표가 오목으로 판단되는 즉시 메시지를 출력하고 프로그램을 종료한다.

```

C:\WINDOWS\system32\cmd.exe
Enter the number of vertices (>2): 5
Input the value x and y of P1: -6 -4
Input the value x and y of P2: -3 -6
Input the value x and y of P3: -2 -2
Input the value x and y of P4: 0 -4
Concave polygon!
계속하려면 아무 키나 누르십시오 . . .
  
```

그림 2. 5각형이 P4에서 오목으로 판별되어, 그 즉시 메시지를 출력하고 종료

(사용자 정의 함수)

- `cal_area()`: 다각형의 넓이 계산하는 함수
- `is_ccw()`: 입력된 좌표를 검사하면서, 볼록 다각형인지 아닌지 판단하는 함수

(힌트)

- 3개의 좌표 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ 가 주어졌을 때, 삼각형의 넓이를 구하는 공식은 외적을 이용하면 되며, 수식은 아래와 같다.

$$S = \frac{|(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)|}{2}$$

- 두 벡터 \vec{a}, \vec{b} 가 주어졌을 때, \vec{b} 가 \vec{a} 의 어느 방향에 있는지 알기 위해 외적의 부호를 이용하면 되며, 수식은 아래와 같다.

$$\vec{a} = (x_1, y_1)$$

$$\vec{b} = (x_2, y_2)$$

$$ccw = (x_1 y_2 - x_2 y_1)$$

$ccw > 0$ 인 경우, \vec{b} 는 \vec{a} 의 반 시계방향으로 있다.

$ccw = 0$ 인 경우, \vec{b} 는 \vec{a} 와 일직선상에 있다.

$ccw < 0$ 인 경우, \vec{b} 는 \vec{a} 의 시계방향으로 있다.