CSED101. Programming & Problem solving

Fall, 2014

Programming Assignment #3

(70 points)

정태열 (dreamerty@postech.ac.kr)

- ***Due***: **2014. 11. 13 23:59**
- ***Development Environment***: **GNU C Compiler (GCC) and Vi Editor (Editor is optional)**

- ***제출물***
  - **C Code file** (*.c)
    - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙일 것.
  - **보고서 파일** (.doc(x) or .hwp) 예) assn3.doc(x) 또는 assn3.hwp
    - AssnReadMe.pdf 를 참조하여 작성할 것.
    - 리눅스 서버에서 프로그램 컴파일 및 실행하는 과정을 화면 캡쳐하여 보고서에 포함시키고 간단히 설명할 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.)
  - 이번 과제에서 추가 기능 구현에 대한 추가 점수가 없습니다.
  - 과제를 작성하는 데 있어서 전역변수를 선언하여 사용할 수 없다.
  - 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
  - 이번 과제의 주제는 배열입니다. 다음 과제의 주제가 포인터이므로 이번 과제에서는 포인터를 사용할 수 없으며, 사용할 시에는 감점이 됩니다.

■ Problem 1: 배열을 이용한 큰 수의 사칙연산 (40점)

1. 문제

- 배열을 이용한 큰 수의 사칙연산을 구현해 본다.

2. 목적

- 배열의 선언과 사용을 익힌다
- 함수에서 배열을 매개변수로 사용하는 방법을 익힌다
- int형 변수로 표현할 수 없는 큰 정수를 배열로 표현하고 다루어본다.

3. 문제 설명

C의 데이터 타입 중 4바이트 int는 -2,147,483,648 ~ 2,147,483,647까지의 수를 표현할 수 있고 4바이트 unsigned int는 0 ~ 4,294,967,295까지의 수를 표현할 수 있다. 이번 과제에서는 C의 기본 데이터 타입으로 표현하지 못하는 큰 숫자들의 사칙 연산을 행렬을 이용해 구현해 본다.

4. 가정 및 요구사항

(1) 사용자로부터 두 개의 숫자를 입력 받는다.
(2) 두 개의 숫자는 모두 정수이며, 각각 20 자리를 넘지 않는다. (최대 20자리).
(3) 두 숫자의 연산 결과는 40 자리를 넘지 않는다.
(4) 양의 정수뿐 아니라 음의 정수 및 0 에 대해서도 입력 및 계산이 가능해야 한다.
(5) 음의 정수의 입력 및 출력을 위해 숫자 앞에 '-'을 이용한다. 예) -55555
(6) 연산을 위해 숫자를 읽어올 때, 각 자리의 숫자를 character type 으로 하나씩 읽어서 integer 형으로 변환하여 int 형 배열에 저장하여 사용한다.
(7) 덧셈 시 받아올림 처리, 뺄셈 시 받아내림을 처리하는 과정이 필요하다.
(8) 나눗셈은 몫만을 출력한다. 단, 음수의 나눗셈의 경우 몫은 C에서 기본적으로 구현된 것과 같이 -10/3 = -3으로 계산한다 (-10/3 = -4 로 계산하지 않는다)
예) C 에서 printf("%d",-10/3)을 실행할 경우 -3이 출력 됨.
(9) 나눗셈에서 0으로 나누는 경우 "Cannot divide by zero"라는 오류 메시지를 출력한다.
(10) 이 문제를 해결하기 위해 반드시 아래 기능에 해당하는 **<u>사용자 정의 함수</u>**를 정의하고 사용해야 한다. 아래 기능 외에 필요한 함수를 정의해서 사용할 수 있다.
- **add: 두 정수의 덧셈을 수행하는 함수**
- **sub: 두 정수의 뺄셈을 수행하는 함수**
- **mul: 두 정수의 곱셈을 수행하는 함수**
- **div: 두 정수의 나눗셈을 수행하고 그 몫을 구하는 함수**

(11) 사칙연산은 10초 이내에 이루어져야 한다. 즉 프로그램 수행 시간은 10초를 넘을수 없다.
(12) 잘못된 입력은 없다고 가정한다.
(13) 배열을 사용하지 않는 경우 0점 처리한다.

5. 주의사항

- 소스코드는 "assn3_1.c" 로 저장할 것.
- 보고서는 "assn3.doc" or "assn3.hwp"로 저장 할 것 (보고서는 통합하여 작성)

6. Tips

- 배열에 숫자를 저장할 때 배열의 index 0을 1의 자리로 생각하면 쉽게 구현할 수 있다. (array[0]은 1의자리, array[1]은 2의 자리, array[2]은 3의 자리, ...)

7. 실행 예시

(1) 초기 실행 화면

```
[dreamerty@programming ass3]$ ./assn3_1.out
Enter the first number:
```

(2) 사용자가 2개의 숫자 입력 후, 사칙연산 결과 출력 (빨간색 밑줄은 사용자 입력을 뜻함)

```
[dreamerty@programming ass3]$ ./assn3_1.out
Enter the first number: 55555
Enter the second number: 333
a = 55555
b = 333
a + b = 55888
a - b = 55222
a * b = 18499815
a / b = 166
[dreamerty@programming ass3]$
```

(3) 예시

```
Enter the first number: 11111222223333344444
Enter the second number: 555556666677777
a = 11111222223333344444
b = 555556666677777
a + b = 11111777780000022221
a - b = 11110666666666666667
a * b = 617291358111111111080246790122098 8
a / b = 20000
```

```
Enter the first number: -555556666677777888 8
Enter the second number: -444443333322222
a = -555556666677777888 8
b = -444443333322222
a + b = -55556111111111111110
a - b = -55555222224444466666
a * b = 246913456787654222217777783950691 36
a / b = 125000
```

```
Enter the first number: 111112222233333
Enter the second number: 44444555556666677777
a = 111112222233333
b = 44444555556666677777
a + b = 44444666668888911110
a - b = -44444444444444444444
a * b = 4938333334074062962864196419740741
a / b = 0
```

```
Enter the first number: -111112222233333
Enter the second number: -99999888887777766666
a = -111112222233333
b = -99999888887777766666
a + b = -99999999999999999999
a - b = 99999777775555533333
a * b = 11111209877407370370308641481477778
a / b = 0
```

```
Enter the first number: 999997777755555
Enter the second number: -2222244444
a = 999997777755555
b = -2222244444
a + b = 999995555511111
a - b = 999999999999999
a * b = -22222395056296288888864 20
a / b = -449994
```

```
Enter the first number: -888886666644444
Enter the second number: 3333355555
a = -888886666644444
b = 3333355555
a + b = -888883333288889
a - b = -888889999999999
a * b = -29629753080246906172864 20
a / b = -266664
```

```
Enter the first number: 1111122222
Enter the second number: -3333344444
a = 1111122222
b = -3333344444
a + b = -2222222222
a - b = 4444466666
a * b = -37037530853086345 68
a / b = 0
```

```
Enter the first number: -2222233333
Enter the second number: 4444455555
a = -2222233333
b = 4444455555
a + b = 2222222222
a - b = -6666688888
a * b = -98766172813580148 15
a / b = 0
```

```
Enter the first number: 0
Enter the second number: 77777
a = 0
b = 77777
a + b = 77777
a - b = -77777
a * b = 0
a / b = 0
```

```
Enter the first number: 55555
Enter the second number: 0
a = 55555
b = 0
a + b = 55555
a - b = 55555
a * b = 0
a / b = Cannot divide by zero
```

```
Enter the first number: 99999999999999999999
Enter the second number: 99999999999999999999
a = 99999999999999999999
b = 99999999999999999999
a + b = 199999999999999999998
a - b = 0
a * b = 9999999999999999999800000000000000000001
a / b = 1
```

```
Enter the first number: 99999999999999999999
Enter the second number: 1
a = 99999999999999999999
b = 1
a + b = 100000000000000000000
a - b = 99999999999999999998
a * b = 99999999999999999999
a / b = 99999999999999999999
```

```
Enter the first number: -9
Enter the second number: 99999999999999999999
a = -9
b = 99999999999999999999
a + b = 99999999999999999990
a - b = -100000000000000000008
a * b = -899999999999999999991
a / b = 0
```

```
Enter the first number: 7
Enter the second number: -77777777777777777777
a = 7
b = -77777777777777777777
a + b = -77777777777777777770
a - b = 77777777777777777784
a * b = -544444444444444444439
a / b = 0
```

```
Enter the first number: 0
Enter the second number: -77777777777777777777
a = 0
b = -77777777777777777777
a + b = -77777777777777777777
a - b = 77777777777777777777
a * b = 0
a / b = 0
```

```
Enter the first number: -77777777777777777777
Enter the second number: 0
a = -77777777777777777777
b = 0
a + b = -77777777777777777777
a - b = -77777777777777777777
a * b = 0
a / b = Cannot divide by zero
```

■ Problem 2: 삼목 게임 (30점)

1. 문제

- 2차원 배열을 이용하여 삼목 게임을 구현해 본다.

2. 목적

- 2차원 배열의 선언과 사용을 익힌다
- 함수에서 2차원 배열을 매개변수로 사용하는 방법을 익힌다

3. 게임 규칙

(1) 보드의 크기는 3 X 3인 정사각형으로 모든 칸은 처음에는 비어 있다.
(2) 게임에 참여하는 두 사람은 번갈아 가며 자기 차례에 하나씩 O나 X를 빈칸에 표시하여 차지한다.
(3) 오직 빈칸에만 표시할 수 있다.
(4) 먼저 가로, 세로 혹은 대각선 한 줄을 차지하는 사람이 이긴다.
(5) 모든 빈 칸에 표시가 되었는데도 아무도 한 줄을 차지하지 못하면 비기는 경우가 된다.
(6) 한 게임은 누군가가 이기거나 빈칸이 모두 채워질 때까지 진행된다.

4. 설명 및 요구사항

(1) 삼목 게임을 컴퓨터와 대결하는 방식으로 구현하도록 한다.
(2) 게임을 시작할 때 컴퓨터와 사용자가 중 돌을 누가 먼저 놓을지는 매 경기마다 임의로 결정되게 한다.
(3) 출력되는 보드의 좌표는 아래와 같다.

(0, 0)	(0, 1)	(0, 2)
(1, 0)	(1, 1)	(1, 2)
(2, 0)	(2, 1)	(2, 2)

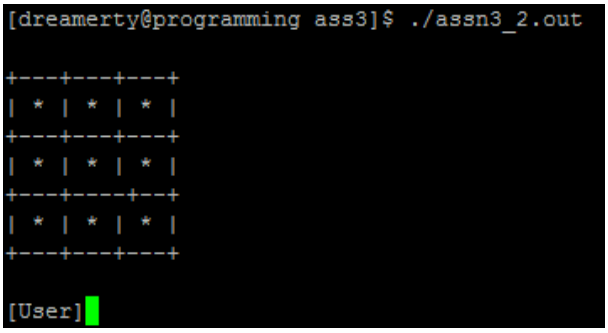
(4) 사용자는 자기 순서에, 보드의 좌표에 해당되는 2개의 정수 (정수 범위: 0~2 사이)를 입력한다. 틀린 범위의 값과 같은 잘못된 입력은 없다고 가정한다. 단, 사용자가 실수로 이미 채워진 값을 선택한 경우에는, 다시 좌표를 입력 받도록 한다.
(5) 컴퓨터는 자기 순서에, 임의로 놓을 곳을 정한다.
(6) 누군가가 가로, 세로, 혹은 대각선 한 줄을 차지하거나 더 이상 놓을 곳이 남아 있지 않은 경우 경기가 끝난다. 경기가 끝날 때 사용자가 이겼는지, 사용자가 졌는지 혹은 비겼는지에 대해 메시지를 출력한다. ('You Win', 'Computer Wins', 'Draw' 중 출력)
(7) 출력되는 보드판은 빈칸을 '*', 그 경기를 먼저 시작하는 쪽이 'O' (알파벳 대문자 O), 나중에 시작하는 쪽이 'X' (알파벳 대문자 X)로 표시한다.
(8) 한 경기가 끝나면 경기를 계속할지 묻는다. 사용자가 정수 1을 입력하면 새 경기를 시작하고 2를 입력하면 'Good bye'를 출력하고 프로그램을 종료한다. 잘못된 입력은 고려하지 않는다.
(9) 이 문제를 해결하기 위해 반드시 아래 기능에 해당하는 **사용자 정의 함수**를 정의하고 사용해야 한다. 아래 기능 외의 필요한 함수를 정의해서 사용할 수 있다.

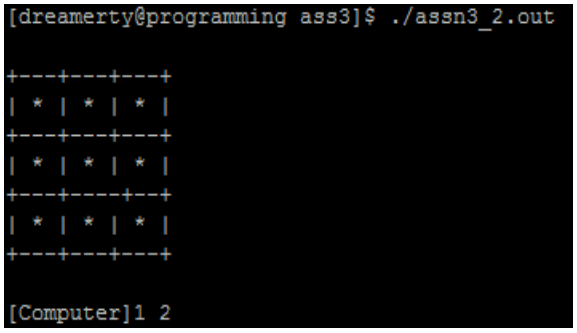
- **print_board: 보드판을 출력하는 기능**
- **check_board: 게임의 승패를 검사하는 기능**

5. 실행 예시

(1) 초기 실행 화면

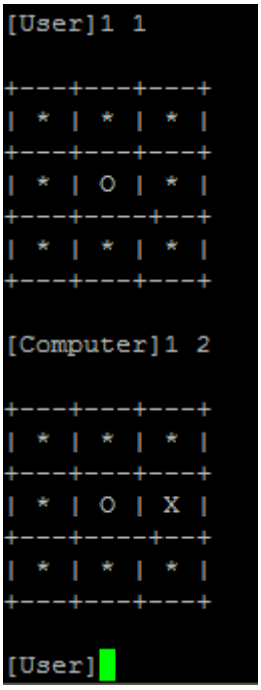
- 컴퓨터와 사용자가 중 돌을 누가 먼저 놓을지 매 경기마다 임의로 결정되게 한다.



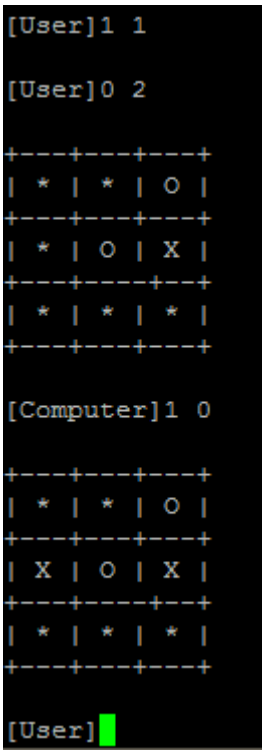


(2) 사용자가 먼저 돌을 놓는 경우, 빈 보드판이 출력되고 사용자의 입력을 기다린다.

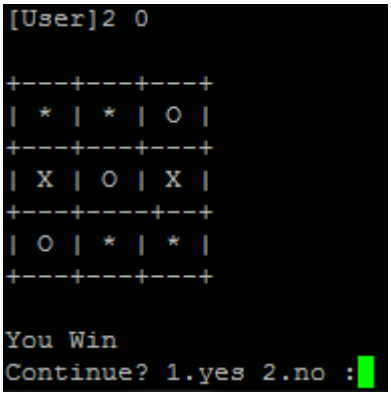
(3) 사용자가 (1, 1)의 자리에 돌을 놓았다. 사용자가 입력한 좌표가 반영된 보드가 출력되고 즉시, 컴퓨터는 (1, 2)의 자리에 돌을 놓았다.



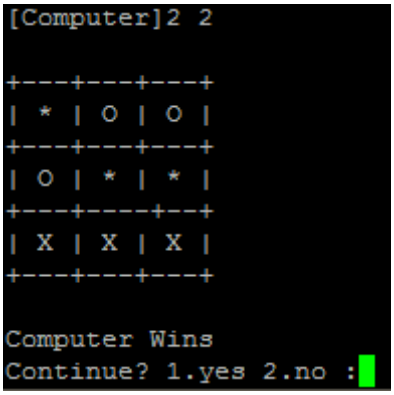
(4) 사용자가 이미 채워진 (1, 1)에 돌을 놓으려고 하면, 돌을 놓을 수 없는 자리이므로 다시 입력을 받아 (0, 2)에 돌을 놓았다. 컴퓨터는 (1, 0)에 돌을 놓았다.



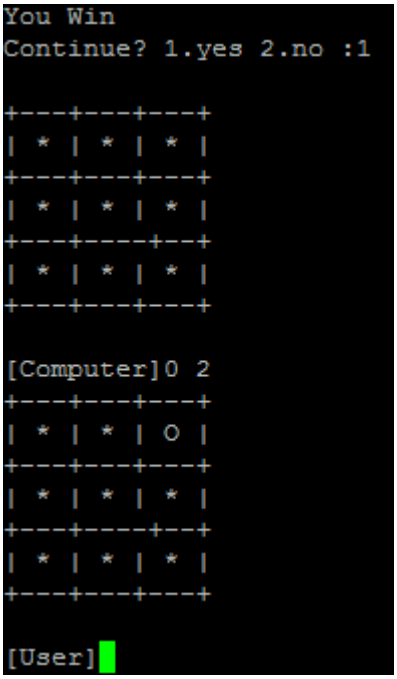
(5) 사용자가 (2, 0)의 자리에 돌을 놓았다. 사용자의 돌(O)이 먼저 대각선 한 줄을 채워 사용자가 게임을 이겼다. 게임을 계속 할지를 묻는 메뉴가 출력된다.



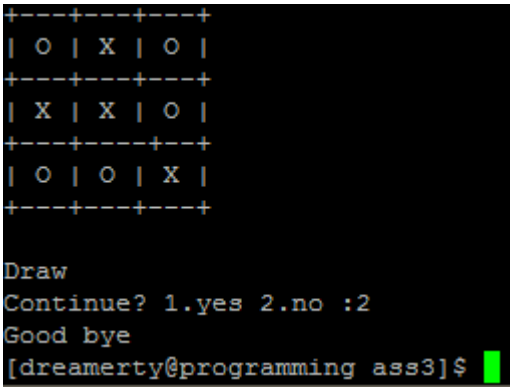
컴퓨터가 이길 경우 아래와 같이 출력된다.



(6) 사용자로부터 1을 입력 받아 새 경기를 시작했다.



(7) 경기를 비기고 2를 입력 받아 프로그램을 종료했다.



6. 주의사항

- 소스코드는 "assn3_2.c" 로 저장할 것.
- 보고서는 "assn3.doc" or "assn3.hwp"로 저장 할 것 (보고서는 통합하여 작성)