**2017년 컴퓨터 프로그래밍 1**

* **HW 02 -**

|  |  |
| --- | --- |
| **제 출 일 자** | 2017.09.14 |
| **이 름** | 서주연 |
| **학 번** | 201702025 |
| **분 반** | 08 |

|  |
| --- |
| **실습 2-1** |
|  |
| 이차방정식의 해를 구하는 프로그램으로 계수를 미리 정해두고 컴퓨터가 이차방정식의 해를 구한다. |
| Q. 이 프로그램을 수행했을 경우 어떤 현상이 발생하는가?(경험한 그대로를 기록할 것)  A. 계수를 (1)두 개의 서로 다른 실근이 존재하는 경우, (2)두 개가 동일한 실근, 즉 중근이 존재하는 경우, (3)실근이 존재하지 않는 경우 이 세가지 경우에 대해 각각 2번씩 프로그램을 실행하였다. (1)과 (2)의 경우에는 실근인 해가 나오지만 실근이 존재하지 않는 (3)의 경우에는 해가 NaN으로 출력이 된다. |

|  |
| --- |
| **실습 2-2** |
|  |
| 1부터 5까지의 합을 출력하는 프로그램이다. |
| Q1-1. 알고리즘은?  A1-1. 1  1+2  1+2+3  1+2+3+4  1+2+3+4+5  Q1-2. 쉽게 처리하는 방법?  A1-2. 매번 출력하기 전에 1부터 해당 숫자까지의 합을 계산한다. 그리고 그 합을 출력한다.  Q2-1. 필요한 변수는?  A2-1. Sum  Q2-2. 매 출력 단계에서 합을 계산한 값을 저장하려면?  A2-2. sum = 1 ;  System.out.println(……);  sum = 1 + 2 ;  System.out.println(……);  sum = 1 + 2 + 3 ;  System.out.println(……);  ……  이런 식으로 sum에 계산한 값을 저장한다.  Q2-3. 이들은 정수형인가 실수형인가?  A2-3. 정수형이다. 실수형으로도 계산이 가능하지만 실수형으로 쓸 경우 컴퓨터가 계산하는 과정이 더 복잡해지기 때문에 정수형을 사용한다.  Q3-1. 좀 더 효과적인 방법은?  A3-1. 첫 번째 출력 단계에서 1까지의 합을 계산했다. 그 다음 두 번째 출력을 위해서는, 앞 단계에서 계산한 합에 2를 더하면 그 합을 계산할 수 있다. 세 번째도 두 번째 출력과 마찬가지로 앞 단계에서 계산한 합에 3을 더하면 그 합을 계산할 수 있다.  Q4-1. 각 프로그램은 계산을 얼마나 했나?  A4-1. 첫 번째 프로그램에서는 15번, 두 번째 프로그램에서는 9번을 계산했다.  Q4-2. 어느 프로그램이 계산 횟수가 적은가?  A4-2. 두 번째 프로그램(전 단계의 sum 다음단계에서 이용한 프로그램)의 계산횟수가 적다. 따라서 두 번째 프로그램이 더 효율적이다. |

|  |
| --- |
| **실습 2-3** |
|  |
| 1부터 10까지의 제곱의 합을 출력하는 프로그램이다. |
| Q1-1. 필요한 변수는?  A1-1. 합을 나타내는 변수 sum이 필요하다.  Q1-2. 변수의 자료형은?  A1-2. 정수형(실수형도 불가능한 것은 아니다.)  Q1-3. 알고리즘은?  A1-3. 1\*1   1\*1+2\*2   1\*1+2\*2+3\*3   1\*1+2\*2+3\*3+4\*4   1\*1+2\*2+3\*3+4\*4+5\*5   1\*1+2\*2+3\*3+4\*4+5\*5+6\*6   ……  이런 식으로 알고리즘을 만들면 된다.  Q2-1. 계산 회수는?  A2-1. 29회로 이전단계의 sum을 활용하지 않는 것보다 활용하는 것이 더 효율적이다. (지금까지 배운 내용으로는 계산과정을 더 줄일 수 없을 것 같다.) |