

1 차 종합 평가

작성자: 이상훈

*** 모든 프로그래밍 문제는 함수를 작성하지 않을 경우 40% 감점**

1. 대략 한 달간 느낀점을 작성하시오. **(배점: 4 점)**

하루하루 기존에 배운 것과 더불어 몰랐던 것들을 배우느라 정신이 없었다.

특히 자료구조에서 메모리 구조를 그리는 연습은 동작 오류 시 잘못된 점을 빨리 찾을 수 있게 해줬다.

2. 아래 Code 를 작성하고 이에 대한 기계어 분석과 그림을 그리시오. **(배점: 4 점)**

이때의 ***pppn 의 결과에 대해 그림과 함께 자세히 설명하시오.

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

```
int main(void)
```

```
{
```

```
    int num = 7, num2 = 3;
```

```
    int *pn = &num;
```

```
    int **ppn = &pn;
```

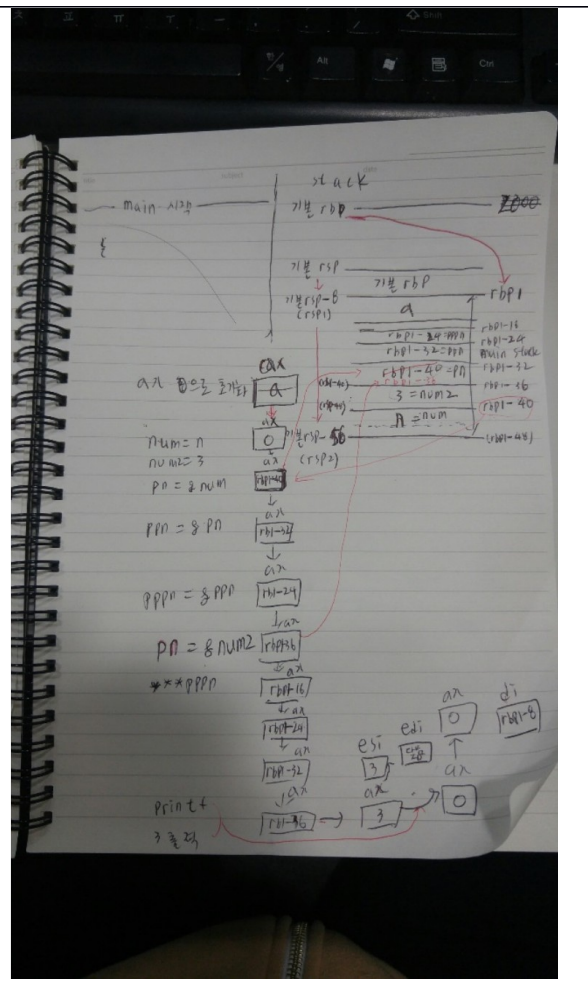
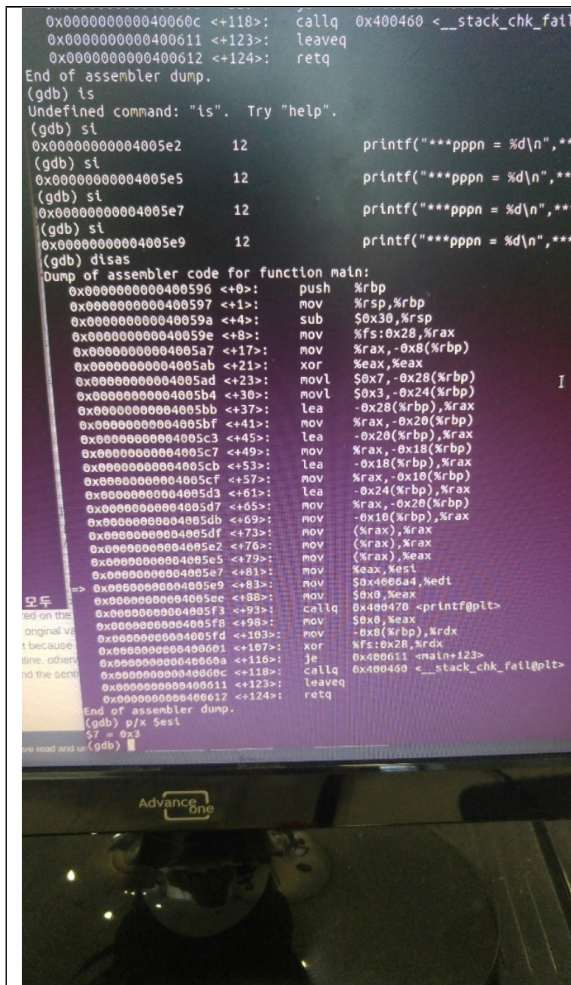
```
    int ***pppn = &ppn;
```

```
    pn = &num2;
```

```
    printf("***pppn = %d\n", ***pppn);
```

```
    return 0;
```

```
}
```



3. 단 한 번의 연산으로 대소문자 전환을 할 수 있는 연산에 대해 기술하시오. (배점: 2 점)
(프로그래밍 하시오)

4. Stack 과 Queue 의 차이점에 대해 작성하시오. (배점: 1 점)

stack 은 가장 나중에 들어온 데이터가 가장 먼저 출력되고 Queue 는 가장 처음 들어온 데이터가 먼저 출력된다.

5. Stack 및 Queue 외에 Tree 라는 자료구조가 있다. (배점: 1 점)

Tree 는 어떠한 이점이 있는가 ?

Tree 는 하나의 조건 비교를 통해 이진 탐색이 가능하므로 스택이나 큐에 비해 검색의 양을 줄여 검색 속도를 향상 시킬 수 있다.

6. Tree 다음으로는 AVL Tree 라는 것이 존재한다. (배점: 1 점)

AVL Tree 는 어떠한 이유로 사용이 되는가 ?

그리고 AVL Tree 에는 어떠한 약점이 존재하는가 ?

Avl Tree 는 데이터의 입력 삭제 시 마다 데이터 위치 변동이 잦으므로 서버에 부담을 줄 수 있다.

7. 임의의 값 x 가 있고,

이를 134217728 단위로 정렬하고 싶다면 어떻게 할까 ? (배점: 4 점)

(힌트 : $134217728 = 2^{27}$)

8. int p[7]와 int (*p)[7]의 차이에 대해 기술하시오. (배점: 1 점)

p[7]은 배열이고 (*p)[7]은 배열 포인터로 p 라는 단순 변수이다.

9. `char *str = "Damn it! Where is my Important Pointer ?"`라는 문자열이 있다 **(배점: 2 점)**

여기에 대문자가 총 몇 개 사용되었는지 세는 프로그램을 만들어보자

10. 임의의 구조체를 한 개 만들고

해당 구조체에 대한 변수 2 개를 선언한 이후 **(배점: 2 점)**

구조체의 멤버들에 적절한 값을 입력한 후 해당 값을 서로 교환해보라
(프로그래밍 하시오)

11. Intel Architecture 와 ARM Architecture 의 차이점은 ? **(배점: 1 점)**

intel 은 CISK 구조이고 ARM 은 RISK 구조이다.

12. 이것이 없으면 C 언어를 사용할 수 없다. **(배점: 1 점)**

이것은 무엇일까 ?

C 컴파일러

13. 1, 3, 4, 7, 11, 18, 29, 47, 76, ... 형태로 숫자가 진행된다. **(배점: 2 점)**

23 번째 숫자는 무엇일까 ?

(프로그래밍 하시오)

14. 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... 형태로 숫자가 진행된다. **(배점: 2 점)**

1 ~ 57 번째까지의 수들로 홀수들의 합을 하고 짝수들의 합을 구한다.

홀수들의 합 - 짝수들의 합의 결과를 출력하시오.

(프로그래밍 하시오)

[복합문제 15 - 17]

15. 난수를 활용해서 Queue 을 구성한다. **(배점: 2 점)**

(중복을 허용하지 않는다)

16. 그 중에서 홀 수만 빼내서 Binary Tree 를 구성하도록 한다. **(배점: 2 점)**

17. 나머지(짝 수)는 Red-Black Tree 로 구성한다. **(배점: 2 점)**

[복합문제 18 - 20]

18. 값이 1 ~ 4096 까지 무작위로 할당하여 배열에 저장하시오. **(배점: 1 점)**

(배열의 크기는 100 개정도로 잡는다)

19. 여기서 각 배열은 물건을 담을 수 있는 공간에 해당한다. **(배점: 2 점)**

앞서서 100 개의 공간에 물건들을 담았는데 공간의 낭비가 있을 수 있다.

이 공간의 낭비가 얼마나 발생했는지 파악하는 프로그램을 작성하시오.

(한 공간의 크기가 4096 을 나타낸다)

20. 공간을 보다 효율적으로 관리하고 싶어서 **(배점: 2 점)**

4096, 8192, 16384 등의 4096 배수로 크기를 확장할 수 있는 시스템을 도입했다.

이제부터 공간의 크기는 4096 의 배수이고

최소 크기는 4096, 최대 크기는 131072 에 해당한다.

발생할 수 있는 난수는 1 ~ 131072 로 설정하고

이를 효율적으로 관리하는 프로그램을 작성하시오.

21. 다음을 적분하는 프로그램을 만들어보자! **(배점: 4 점)**

$3x^2 + 7x$ 를 1 ~ 2 까지 정적분하도록 구현해보자!

22. Memory Hierarchy(메모리 계층 구조)에 대해 기술하시오. **(배점: 1 점)**

메모리 계층 구조는 피라미드 형식으로 꼭대기로 올라가 수록 데이터 처리 속도는 빠르지만 활용할 수 있는 공간은 작다.

제일 꼭대기부터 레지스터, 캐시, ram, rom, 하드디스크 등의 보조 기억장치 순으로 배치된다.

23. C 언어에서 중요시하는 메모리 구조에 대해 기술하시오. **(배점: 1 점)**

메모리 구조는 stack, data, heap, text 3 부분으로 이루어져 있다.

24. 변수의 정의를 기술하시오. **(배점: 1 점)**

데이터를 저장하는 메모리 공간

25. 포인터의 정의를 기술하시오. **(배점: 1 점)**

메모리 주소를 저장하는 메모리 공간

26. 함수 포인터의 정의를 기술하시오. **(배점: 1 점)**

함수를 가리키는 주소를 저장하는 메모리 공간

27. **(배점: 2 점)**

((함수 포인터를 반환하고 함수 포인터를 인자로 취하는) 함수 포인터를 반환하고
함수 포인터를 인자로 취하는) 함수 포인터를 반환하는 함수를 프로그래밍 해보자!

28. goto 를 사용하는 이유에 대해 기술하시오. **(배점: 1 점)**

goto 를 사용하면 jmp 어셈블리어 한 줄로 반복문에서 예외처리를 쉽게 할 수 있다.

29. 포인터의 크기에 대해 가상 메모리와 관련하여 기술하시오. **(배점: 1 점)**

포인터 크기는 cpu 가 운영할 수 있는 가상메모리 크기에 따라 달라진다. 포인터 크기 1bit 당 2bit 의 메모리를 운영할 수 있다. 포인터 크기가 1bit 증가할 수로 2 의 승수만큼 운영할 수 있는 공간이 커진다. 운영할 수 있는 공간이 4GB 라면 이때의 포인터 크기는 4byte 이다.

30. 16 비트 ADC 를 가지고 있는 장치가 있다. **(배점: 2 점)**

보드는 12 V 로 동작하고 있고 ADC 는 -5 ~ 5 V 로 동작한다.

ADC 에서 읽은 값이 12677 일 때 이 신호를 전압으로 변환해보자!

31. 24 비트 DAC 장치가 있다. **(배점: 2 점)**

이 장치는 -12 V ~ 12 V 로 동작하며 보드는 5 V 로 동작한다.

DAC 에서 나온 전압이 9.7 V 일 때

어떤 디지털 신호를 입력 받은것인지 파악해보자!

32. $\sin(x)$ 값을 프로그램으로 구현해보도록 한다. **(배점: 2 점)**

어떤 radian 값을 넣든지 그에 적절한 결과를 산출할 수 있도록 프로그래밍 한다.

my_sin() 함수를 구현해서 sin() 과 같은 결과가 나오게 만들어 보라는 의미임.

33. Red Black Tree 를 사용하는 이유를 기술하시오. (배점: 1 점)

AVL 트리는 트리의 양쪽의 균형은 완벽하게 맞추지만 데이터 출입 시 균형을 맞추는 데 너무 많은 자원을 사용한다. 이를 완화하고자 사용하는 것이 Red Black tree 이다.

34. 이중 포인터를 활용하는 이유는 무엇인가 ? (배점: 1 점)

이중 포인터를 사용하면 2 번의 접근의 기회를 얻을 수 있다. 그렇게 함으로서 처음 접근한 값을 변경시키지 않고도 2 번 접근한 값을 변경 시킬 수 있다. 그러므로 첫번째 접근한 메모리 주소에 있는 값을 보호할 수 있다.

35. 재귀 호출에 관하여 아는대로 기술하시오. (배점: 1 점)

재귀호출은 함수 동작 중에 자기 자신과 같은 형태의 함수를 다시 호출하는 것을 말한다. 단 이때 새로 호출된 함수는 기존에 호출한 함수와 별도의 stack 공간을 할당 받는다.

36. Cache Affinity(캐시 친화성)에 대해 아는대로 기술하시오. (배점: 1 점)

캐시 친화성이란 메모리에 올라와 있는 데이터를 얼마만큼 캐시로 끌고 올 수 있는 지를 말한다. 많은 양을 끌고 올 수 있을수록 친화성이 좋다.

37. 삽입 정렬을 구현하시오(내림차순) (배점: 2 점)

38. goto 가 할 수 없는 일은 무엇인가 ? (배점: 1 점)

39. 수업시간에 배운 Memory Pool 기법으로 Queue 를 작성하시오. (배점: 2 점)

[복합 문제 40 - 42]

40. rand() 를 두 번 사용하여 이중 배열을 만들도록 한다. (배점: 2 점)

각 index 를 결정하는데 rand() 가 사용된다.

41. 생성된 배열에 랜덤값을 할당하도록 한다. (배점: 2 점)

(중복된 결과를 허용하지 않음)

42. 생성한 배열의 항목들을 오름차순으로 정렬하라. (배점: 2 점)

Ex) {1, 3, 5}, {1, 4, 2}, {1, 2, 3}, {4, 2, 1} -> {1, 2, 3}, {1, 3, 5}, {1, 4, 2}, {4, 2, 1}

43. C++ 의 Class 를 C 언어로 구현해보도록 하자! (배점: 2 점)

C++ 에서 학습한 대표적인 케이스로 Car Class 를 C 언어로 구현해보도록 한다.

[복합 문제 44 - 47]

44. C++ 의 연산자 오버로딩을 통해서 복소수의 덧셈을 구현해보도록 한다. (배점: 1 점)

$a = 3 + 3i$, $b = 2 + 2j$

45. 복소수의 뺄셈을 구현하도록 한다. (배점: 1 점)

46. 복소수의 곱셈을 구현하도록 한다. (배점: 2 점)

47. 복소수의 크기를 구하시오. **(배점: 2 점)**

48. 사람은 이름과 나이를 가지고 있다. **(배점: 1 점)**

학생은 이름과 나이와 전공을 가지고 있다.

이를 C++ 기반으로 표현해보시오.

[복합 문제 49 - 52]

49. C++ 기반으로 Queue 의 enqueue 를 작성하도록 한다. **(배점: 2 점)**

50. Queue 의 print_queue 를 작성하도록 한다. **(배점: 2 점)**

51. Queue 의 find_data 를 작성하도록 한다. **(배점: 2 점)**

52. Queue 의 dequeue 를 작성하도록 한다. **(배점: 2 점)**

[복합 문제]

53. Triangle 클래스를 구현하시오. **(배점: 1 점)**

54. 높이와 밑변을 기반으로 넓이를 구하는 함수를 작성하시오. **(배점: 1 점)**

55. 삼각형의 사이 각도와 두 변의 길이를 알 때 **(배점: 2 점)**

삼각형의 넓이를 구하는 함수를 작성하시오.

56. 삼각형의 좌표 3 개를 알 때 삼각형의 넓이를 구하는 함수를 작성하시오. **(배점: 2 점)**

[복합 문제 57 - 58]

57. MMORPG 를 만든다 가정하고 전사, 도적, 마법사 클래스를 구현하시오. **(배점: 1 점)**

58. 2 차 전직을 구현하시오. **(배점: 1 점)**

59. 수업중 배웠던 리눅스 명령어가 대략 13 개에 해당한다. **(배점: 1 점)**

이중 기억나는 것 10 가지를 적고 각 명령어에 대해 설명하시오.

ls - 목록 보기, cd - 경로 변경, mkdir - 디렉토리 만들기, cp - 파일 복사, rm -rf -파일 삭제,
mv -파일 이동 또는 이름바꾸기, cat -파일 소스 보기, gdb -디버깅, vi-입력창, sudo - 관리자
권한 임시 부여, gcc -.c 파일 컴파일

60. 디버깅 용도와 배포(release) 용도의 프로그램을 동시에 개발하고자 한다 **(배점: 1 점)**

이를 대표할 수 있을만한 프로그램을 간략하게 만들어보고

특수한 컴파일 옵션이 필요할텐데 이에 대해 기술하시오.