

문제 1

1점

s_val - q_val 의 값은?

```
s = Stack()
s.push(15)
s.push(21)
s.pop()
s.push(54)
s.push(76)
s_val = s.pop()

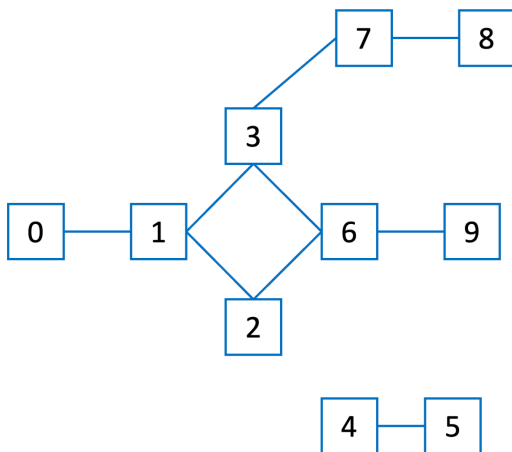
q = Queue()
q.enqueue(32)
q.enqueue(43)
q.dequeue()
q.enqueue(34)
q_val = q.dequeue()
```

- ☐ 33
- ☐ 68
- ☐ 77
- ☐ 81
- ☐ 19

문제 2

1점

Postorder로 다음 그래프를 Depth-First 순회했을 때 출력될 수 있는 노드의 순서로 옳바른 것은? (neighbor list에는 오름차순으로 노드가 지정되어 있다.)



- ☐ 8739621054
- ☐ 9687354210
- ☐ 9687310254
- ☐ 0123786945
- ☐ 8739654210

문제 3

1점

다음 중 옳은 것을 모두 고르시오 (2개)

- ☐ Interpreter receives input from the preprocessor
- ☐ Compiler always keeps the order of instructions from the preprocessed code
- ☐ Compiler executes the program after translating the whole program into a machine language program
- ☐ The C preprocessor takes as input the original C source and header files
- ☐ Linker receives object modules and links together all object modules

문제 4

1점

다음은 Python과 C에 대한 설명이다. 옳은 설명을 모두 고르시오 (2개)

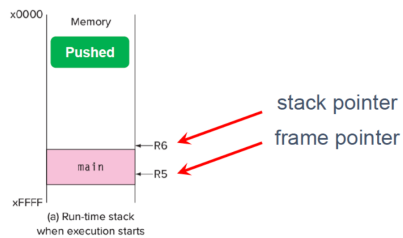
- ☐ In Python, memory management is automatic and handled by the language runtime
- ☐ Python is a the low-level language that can be used to write highly optimized, memory-efficient code
- ☐ C is optimized for object-oriented programming having principles of encapsulation, abstraction, inheritance, and polymorphism
- ☐ Pointers in C can be passed as function arguments, allowing functions to modify variables outside their scope
- ☐ A group of reusable variables and functions packaged into shareable units are Classes in Python

문제 5

1점

다음과 같은 C의 메모리 구조를 가정할 때, 다음 중 틀린 것은?

('x0000'은 가장 낮은 메모리 주소를, 'xFFFF'는 가장 높은 메모리 주소를 나타낸다)



- ☐ 함수가 return되면 스택 메모리가 deallocate되고 stack pointer가 감소한다
- ☐ 스택에 더 많은 데이터가 추가될수록 stack pointer가 감소한다
- ☐ 가장 나중에 실행된 함수가 가장 먼저 끝난다
- ☐ Frame pointer는 현재 실행 중인 function의 시작을 가리킨다
- ☐ C에서의 function call의 메모리 관리의 LIFO 데이터 구조를 사용하여 구현된다

문제 6

1점

다음 중 옳지 않은 것은?

- ☐ Every graph is a tree, but not every tree is a graph.
- ☐ Computer systems achieve dependability via redundancy.
- ☐ Memory hierarchy is meaningless without principle of locality.
- ☐ A recursive function that calls itself, either directly or indirectly.
- ☐ Principle of locality refers to the tendency of the computer program to access the same piece of data for a particular time period.

다음과 같은 코드가 주어질 때, **main 함수 내에서 호출하고 있는 함수**에 대한 알맞은 **Function name**, **Function call**, **Parameter**, **Argument**를 각각 찾아 고르시오.

```
C pointer.c > main(void)
1  #include <stdio.h>
2
3  void newSwap(int *firstVal, int *secondVal);
4
5  int main(void) {
6      int valA = 7;
7      int valB = 5;
8      printf("Before Swap: valA = %d, valB = %d\n", valA, valB);
9      newSwap(&valA, &valB);
10     printf("After Swap: valA = %d, valB = %d\n", valA, valB);
11     return 0;
12 }
13
14 void newSwap(int *firstVal, int *secondVal) {
15     int tempVal;
16     tempVal = *firstVal;
17     *firstVal = *secondVal;
18     *secondVal = tempVal;
19     printf("In Swap: firstVal = %d, secondVal = %d\n", *firstVal, *secondVal);
20 }
```

<보기>

- newSwap(&valA, &valB)
- 7, 5
- *int firstVal, *int secondVal
- newSwap
- return 0
- printf
- &valA, &valB

위의 보기에서 **알맞은** 답을 골라 아래 표의 빈칸을 채우시오.

Function name	[a]
Function call	[b]
Parameter	[c]
Argument	[d]

a 의 답변 표시

newSwap

newSwap, printf

b 의 답변 표시

newSwap(&valA, &valB)

c 의 답변 표시

*int firstVal, *int secondVal

d 의 답변 표시

&valA, &valB

문제 8

1점

아래 코드를 실행하여 4를 입력한 뒤 answer에 들어갈 답을 숫자로 적으시오.

```
C quiz.c > main(void)
1  #include <stdio.h>
2  int quiz(int n);
3
4  int main(void) {
5      int number;
6      int answer;
7      printf("Input a number: ");
8      scanf("%d", &number);
9      answer = quiz(number);
10     printf("The answer is %d\n", answer);
11 }
12
13 int quiz(int n) {
14     int result = 1;
15     for(int i=1; i<=n; i++)
16         result*=i;
17     return result;
18 }
```

24

문제 9

1점

빈칸에 들어갈 단어로 알맞은 것을 쓰시오.

[a] create names and make them unique to a particular region of code.

In most program environment, the standard output by default is the screen, and the C++ stream object defined to access it is [b].

In most program environment, the standard input by default is the keyboard, and the C++ stream object defined to access it is [c].

The [d] manipulator produces a newline character, exactly as the insertion of '\n' does.

a 의 답변 표시

namespace
Namespace
namespaces
Namespaces

b 의 답변 표시

cout
std::cout

c 의 답변 표시

cin
std::cin

d 의 답변 표시

endl
std::endl

아래 코드를 실행 했을 때 출력 될 값을 순서대로 적으시오.

```
namespace.cpp > main()
1  #include <iostream>
2
3  namespace spaceA {
4      int value() { return 12; }
5  }
6
7  namespace spaceB {
8      const double log_e = 0.43429;
9      double value() {
10         return 2*log_e;
11     }
12 }
13
14 int main() {
15     std::cout << spaceA::value() << '\n';
16     std::cout << spaceB::value() << '\n';
17     std::cout << spaceB::log_e << '\n';
18     return 0;
19 }
```

	[a]
	[b]
	[c]

a 의 답변 표시

12

b 의 답변 표시

0.86858

c 의 답변 표시

0.43429