

22-2 자료구조 3반 중간고사 복습용 연습 퀴즈

마감 11월 1일 오후 3:30

점수 101

문제 6

이용 가능 11월 1일 오후 2:30-11월 1일 오후 3:30 약 1시간

시간 제한 60 분

지침

이 퀴즈의 점수는 최종점수에 반영되지 않습니다.

시도 이력

	시도	시간	점수
시도 이력	시도 1	54분	14 / 101

11월 1일 오후 3:28에 제출됨

답안 작성 요령

- 코드의 빈칸을 채워 완성하는 경우, 각 빈 칸은 토큰(identifier)을 기준으로 한다. 예를 들어 변수, 연산자, 예약어 등은 하나의 토큰으로 간주한다.
- 동일 토큰이 연이어 나오는 경우, 하나의 토큰으로 간주한다. 예를 들어, 포인터 표시인 * 가 연이어 ** 처럼 나오는 경우 하나의 토큰으로 간주하고 답한다.
- 또한, 객체의 속성을 표시하는 경우, 언어에 따라 ‘.’ 또는 ‘->’로 연결된 객체 레퍼런스(포인터 변수) 표현과 함께 인스턴스 변수나 메소드의 이름은 하나의 토큰으로 간주한다. 즉, 객체 a 의 멤버 변수 value 는 하나의 답안으로 처리한다.
- C/파이썬 경우 모두 문자열값 경우, “ ” 를 사용하여 값을 표현한다. C 코드인 경우, 출력이 문자이면 “ ” 를 이용하여 값을 표현하여야 한다.
- 값을 채워야 하는 빈칸은 {ans #} 형태로 표시한다.
- 코드 맨 앞이 빈칸인 경우, { 이 들여쓰임에 맞추어져 있음에 주의.
- 답안에는 문자열인 경우를 제외하고는 스페이스가 포함되어 있지 않음에 주의

문제 1**4 / 30점**

다음은 파이썬으로 작성된 연결 리스트(linked list)를 위한 프로그램의 일부로, reverse_list 함수는 리스트를 역순으로 만드는 함수이다. Node 클래스는 자료값을 가지는 data 와 다음 노드를 가리키는 nextNode 를 가진다. 빈 칸을 채워 코드를 완성하시오.

```
Class Node:
    # value 는 노드의 값, link 는 다음 노드를 가리킴
    def __init__(self, value, link=None):
        {ans 1} = value
        {ans 2} = link
```

```
Class LinkedList:
    def __init__(self):
        self.firstNode = {ans 3}
```

```
# list: LinkedList 형 객체
def reverse_list (list):
    before = None
    current = {ans 4}
    if {ans 5} is {ans 6} : return

    after = {ans 7}
    while {ans 8} :
        {ans 9} = before
        {ans 10} = current
        {ans 11} = after
        after = {ans 12}

    {ans 13} = {ans 14}
    {ans 15} = current
```

{ans 1} = {ans 2} = {ans 3} =

{ans 4} = {ans 5} = {ans 6} =

{ans 7} = {ans 8} = {ans 9} =

{ans 10} = {ans 11} = {ans 12} =

{ans 13} = {ans 14} = {ans 15}

=

정답 1:

제출된 답변

정답

self.data

정답 2:

제출된 답변

정답

self.nextNode

정답 3:

제출된 답변

정답

None

정답 4:

제출된 답변

정답

list.firstNode

정답 5:

정답입니다!

current

정답 6:

정답입니다!

None

정답 7:

제출된 답변

정답

current.nextNode

정답 8:

제출된 답변

node

정답

after

정답 9:

제출된 답변

none

정답

current.nextNode

정답 10:

제출된 답변

self.value

정답

before

정답 11:

제출된 답변

self.link

정답

current

정답 12:

제출된 답변

self.link

정답

after.nextNode

정답

current.nextNode

정답 13:

제출된 답변

current

정답

current.nextNode

정답 14:

제출된 답변

self.link

정답

before

정답 15:

제출된 답변

self.link

정답

list.firstChild

문제 2

2 / 16점

다음은 두 개의 스택을 이용하여 큐를 구현하는 코드이다. 빈 칸을 채워 완성하시오. 스택은 파이썬 queue 에서 제공하는 스택을 사용한다.

```
import queue
```

```
class TwoStackQueue :
```

```
    def __init__(self) :
```

```
        self.firstStack = {ans 1}()
```

```
        self.secondStack = {ans 2}()
```

```
    def enqueue (self, val) : # value는 data값
```

```
        {ans 3}(val)
```

```
    def dequeue (self) :
```

```
        if {ans 4}() :
```

```
            while not {ans 5}() :
```

```
                {ans 6}( {ans 7}() )
```

```
            return {ans 8}()
```

{ans 1} = Queue

{ans 2} = Queue

{ans 3} = self.firstStack

{ans 4} = self.firstStack.empty

{ans 5} = self.fistStack.qsize

{ans 6} = self.secondStack

{ans 7} = self.firstStack.get

{ans 8} = self.fistSstack.get

정답 1:

제출된 답변

Queue

정답

queue.LifoQueue

정답 2:

제출된 답변

Queue

정답

queue.LifoQueue

정답 3:

제출된 답변

self.firstStack

정답

self.firstStack.put

정답 4:

제출된 답변

self.firstStack.empty

정답

self.secondStack.empty

정답 5:

제출된 답변

self.fistStack.qsize

정답

self.firstStack.empty

정답 6:

제출된 답변

self.secondStack

정답

self.secondStack.put

정답 7:

정답입니다!

self.firstStack.get

정답 8:

제출된 답변

self.fistSstack.get

정답

self.secondStack.get

문제 3

6 / 28점

다음과 같이 스택 클래스 정의를 가정할 때, 스페이스로 분리된 중위 표기 수식을 입력 받아 실제로 결과를 계산하는 프로그램이다. 클래스와 precedence 함수는 상세 코드는 없이 필요 선언만 표기했음에 주의할 것. 중위 표기에는 변수 사용 없이 정수와 연산자로 이루어졌다고 가정하고 빈 칸을 채워 완성하시오.

```
class Stack :
    def __init__( self ): # 생성자
    def isEmpty( self ): # 스택이 비었으면 True 아니면 False 를 반환
    def size( self ): # 원소의 갯수를 반환
    def clear( self ): # 내용 삭제
    def push( self, item ): # push 동작
    def pop( self ): # pop 동작. 빈 경우는 None 반환
    def peek( self ): # peek 동작

# precedence 함수:
# - '(' , ')' 이면 0 반환, '+' , '-' 이면 1반환,
# - '*' , '/' 이면 2 반환, 아니면 -1
def precedence (op):

# 중위 표현을 후위 표현으로 변형하여 리스트 형태로 반환.
def Infix2Postfix( expr ):
    s = Stack()
    output = []
    for term in expr :
        if term in '(' :
            s.push( '(' )
        elif term in ')' :
            while not s.isEmpty() :
                op = s.pop()
                if op == '(' : {ans 5};
                else : {ans 6}( op )
        elif term in "+-*/" :
            while not s.isEmpty() :
                op = s.pop()
                if( precedence( {ans 9} ) <= precedence( {ans 10} ) ):
                    {ans 11}( op )
                    {ans 12}()
                else: {ans 13}
            {ans 14}( term )
        else :
            {ans 15}( term )

    while not s.isEmpty() :
        {ans 17}( {ans 18}() )

    return output

#-----

def evalPostfix( expr ):
```

```

s = {ans 19}()
for token in expr :
    if token in "+-*/" :
        val2 = {ans 20}()
        val1 = {ans 21}()
        if (token == '+'): {ans 22}(val1 + val2)
        elif (token == '-'): {ans 23}(val1 - val2)
        elif (token == '*'): {ans 24}(val1 * val2)
        elif (token == '/'): {ans 25}(val1 / val2)
    else :
        {ans 26}( float({ans 27}) )

return {ans 28}()

```

```

#-----
expr = input("입력 수식(공백문자로 분리)=")
infix = expr.split(' ')
postfix = Infix2Postfix(infix)
result = evalPostfix(postfix)

```

{ans 1} =

Stack

{ans 2} =

s.push

{ans 3} =

s.top

{ans 4} =

{ans 5} =

{ans 6} =

{ans 7} =

{ans 8} =

{ans 9} =

{ans 10} =

{ans 11} =

{ans 12} =

{ans 13} =	<input type="text"/>
{ans 14} =	<input type="text"/>
{ans 15} =	<input type="text"/>
{ans 16} =	s.isEmpty
{ans 17} =	output.append
{ans 18} =	s.pop
{ans 19} =	Stack
{ans 20} =	<input type="text"/>
{ans 21} =	<input type="text"/>
{ans 22} =	<input type="text"/>
{ans 23} =	<input type="text"/>
{ans 24} =	<input type="text"/>
{ans 25} =	<input type="text"/>
{ans 26} =	<input type="text"/>
{ans 27} =	<input type="text"/>
{ans 28} =	<input type="text"/>

정답 1:

정답입니다! Stack

정답 2:

정답입니다!

s.push

정답 3:

제출된 답변

s.top

정답

s.isEmpty

정답 4:

제출된 답변

(답변 안 함)

정답

s.pop

정답 5:

제출된 답변

(답변 안 함)

정답

break

정답 6:

제출된 답변

(답변 안 함)

정답

output.append

정답 7:

제출된 답변

(답변 안 함)

정답

s.isEmpty

정답 8:

제출된 답변

(답변 안 함)

정답

s.peek

정답 9:

제출된 답변

(답변 안 함)

정답

term

정답 10:

제출된 답변

(답변 안 함)

정답

op

정답 11:

제출된 답변

(답변 안 함)

정답

output.append

정답 12:

제출된 답변

(답변 안 함)

정답

s.pop

정답 13:

제출된 답변

(답변 안 함)

정답

break

정답 14:

제출된 답변

(답변 안 함)

정답

s.push

정답 15:

제출된 답변

(답변 안 함)

정답

output.append

정답 16:

정답입니다!

s.isEmpty

정답 17:

정답입니다!

output.append

정답 18:

정답입니다!

s.pop

정답입니다!	정답 19:
	Stack
제출된 답변	정답 20:
	(답변 안 함)
정답	s.pop
제출된 답변	정답 21:
	(답변 안 함)
정답	s.pop
제출된 답변	정답 22:
	(답변 안 함)
정답	s.push
제출된 답변	정답 23:
	(답변 안 함)
정답	s.push
제출된 답변	정답 24:
	(답변 안 함)
정답	s.push
제출된 답변	정답 25:
	(답변 안 함)
정답	s.push
제출된 답변	정답 26:
	(답변 안 함)
정답	s.push

정답 27:

제출된 답변

(답변 안 함)

정답

token

정답 28:

제출된 답변

(답변 안 함)

정답

s.pop

답변 안 함

문제 4

0 / 9점

2차원 배열에 저장된 정수값을 합을 구하는 프로그램을 완성하시오.

```
int getSum(int {ans 1}arr, int row, int col) {
    int i, j, sum = 0;
    for( i = 0; i < row; i++)
        for( j = 0; j < col; j++)
            sum += {ans 2}({ans 3}(arr + {ans 4}) + {ans 5});
    return sum;
}
```

```
int main() {
    int a[ 4 ] = {1,2,3,4}, b[ 4 ] = {5,6,7,8}, c[ 4 ] = {9,10,11,12};
    int {ans 6}array[ 3 ] = {a, b, c};
```

```
    int s = getSum( {ans 7}{ans 8}[{ans 9}], 3, 4);
} // main
```

{ans 1} = {ans 2} = {ans 3} =

{ans 4} = {ans 5} = {ans 6} =

{ans 7} =

{ans 8} =

{ans 9} =

정답 1:

제출된 답변

(답변 안 함)

정답

**

정답 2:

제출된 답변

(답변 안 함)

정답

*

정답 3:

제출된 답변

(답변 안 함)

정답

*

정답 4:

제출된 답변

(답변 안 함)

정답

i

정답 5:

제출된 답변

(답변 안 함)

정답

j

정답 6:

제출된 답변

(답변 안 함)

정답

*

정답 7:

제출된 답변

(답변 안 함)

정답	&
제출된 답변	<p>정답 8:</p> <div>(답변 안 함)</div>
정답	array
제출된 답변	<p>정답 9:</p> <div>(답변 안 함)</div>
정답	0

문제 5

2 / 8점

다음과 같이 C 코드가 주어졌다고 가정하시오.

```
char* ptrArray[ 2 ];
char** ptrptr = ptrArray;
int i;
ptrArray[ 0 ] = "first data";
ptrArray[ 1 ] = "second string";
int str1_len = strlen(ptrArray[ 0 ]);
int str2_len = strlen(ptrArray[ 1 ]);
```

위의 코드가 실행 시, 다음의 메모리 주소를 이용해 값들이 저장된다고 가정하시오.

ptrArray의 주소: 100
 "First Data"의 주소: 1000
 "Second String"의 주소: 2000
 메모리 주소 크기: 8 바이트

• printf("%u", &ptrArray[1]); 의 결과는 무엇인가? 2000

• printf("%u", ptrArray[1]); 의 결과는 무엇인가? 2000

• printf("%c", *ptrArray[1]); 의 결과는 무엇인가? 2000

• printf("%s", *(ptrArray + 1) + 5); 의 결과는 무엇인가?

2000

정답 1:

제출된 답변

2000

정답

108

정답 2:

정답입니다!

2000

정답 3:

제출된 답변

2000

정답

's'

정답 4:

제출된 답변

2000

정답

"d string"

답변 안 함

문제 6**0 / 10점**

다음에 주어진 구조체 list 를 이용하여 두 개의 연결 리스트 x, y 가 생성되었다고 가정하시오. 각 리스트의 맨 앞으로부터 두번째 노드를 교환(swap) 하기 위한 코드를 완성하시오. 주어진 변수 이외는 사용할 수 없음에 주의. 단, 교환되는 노드들의 뒤에 오는 노드는 NULL 이 아니라고 가정하시오.

```
struct list { int data; struct list *next;} *x, *y, *temp;
{ans 1} = x->next->next;
{ans 2} = {ans 3};
{ans 4} = {ans 5};
{ans 6} = x->next;
{ans 7} = {ans 8};
{ans 9} = {ans 10};
```

{ans 1} =

{ans 2} =

{ans 3} =

{ans 4} =

{ans 5} =

{ans 6} =

{ans 7} =

{ans 8} =

{ans 9} =

{ans 10} =

정답 1:

제출된 답변

(답변 안 함)

정답

temp

정답 2:

제출된 답변

(답변 안 함)

정답

x->next->next

정답 3:

제출된 답변

(답변 안 함)

정답

y->next->next

정답 4:

제출된 답변

(답변 안 함)

정답

y->next->next

정답 5:

제출된 답변

(답변 안 함)

정답

temp

정답 6:

제출된 답변

(답변 안 함)

정답

temp

정답 7:

제출된 답변

(답변 안 함)

정답

x->next

정답 8:

제출된 답변

(답변 안 함)

정답

y->next

정답 9:

제출된 답변

(답변 안 함)

정답

y->next

정답 10:

제출된 답변

(답변 안 함)

정답

temp