

[Trang chủ](#) / [Các khoá học của tôi](#) / [Học kỳ 2 \(2023-2024\)](#) / [Khoa Khoa Học Máy Tính](#) / [IT003.O21](#) / [General](#) / [Quiz 7: stack -queue](#)

<b>Bắt đầu vào lúc</b>	Thứ Hai, 29 tháng 4 2024, 8:25 AM
<b>Trạng thái</b>	Đã xong
<b>Kết thúc lúc</b>	Thứ Hai, 29 tháng 4 2024, 8:27 AM
<b>Thời gian thực hiện</b>	2 phút 17 giây
<b>Điểm</b>	9,00/9,00
<b>Điểm</b>	<b>10,00</b> trên 10,00 ( <b>100%</b> )

**Câu hỏi 1**

Đúng

Đạt điểm 1,00 trên 1,00

Cơ chế hoạt động FIFO (First-In, First-Out) dùng cho cấu trúc dữ liệu

Queue

✓

Cơ chế hoạt động LIFO (Last-In, First-Out) dùng cho cấu trúc dữ liệu

Stack

✓

Array

Linked List

Your answer is correct.

The correct answer is:

Cơ chế hoạt động FIFO (First-In, First-Out) dùng cho cấu trúc dữ liệu [Queue].

Cơ chế hoạt động LIFO (Last-In, First-Out) dùng cho cấu trúc dữ liệu [Stack].



**Câu hỏi 2**

Đúng

Đạt điểm 1,00 trên 1,00

Cho một dãy các phép toán trên stack như sau:

1. Push(1);
2. Pop();
3. Push(2);
4. Push(3);
5. Pop();
6. Push(4);
7. Pop();
8. Pop();
9. Push(5);

Kết quả stack sau các thao tác trên là:

- ☒ a. [5 (top)] ✓
- ☐ b. [4, 5 (top)]
- ☐ c. [1, 2, 3, 4, 5 (top)]
- ☐ d. [2, 5 (top)]

Your answer is correct.

The correct answer is:

[5 (top)]

**Câu hỏi 3**

Đúng

Đạt điểm 1,00 trên 1,00

**Cho đoạn chương trình sau. Hãy điền vào chỗ trống để chương trình có thể in ra kết quả thứ tự từ trái sang phải là: {10 9 8 7 6 5 4 3 2 1 0}.**

```
stack s;  
for (int i = 0; i <= 10; i++)  
    [push] ✓ (s, i);  
while (!empty(s)){  
    cout << [top] ✓ (s) << " ";  
    [pop] ✓ (s);  
}
```

Your answer is correct.

The correct answer is:

**Cho đoạn chương trình sau. Hãy điền vào chỗ trống để chương trình có thể in ra kết quả thứ tự từ trái sang phải là: {10 9 8 7 6 5 4 3 2 1 0}.**

```
stack s;  
for (int i = 0; i <= 10; i++)  
    [push](s, i);  
while (!empty(s)){  
    cout << [top](s) << " ";  
    [pop](s);  
}
```

**Câu hỏi 4**

Đúng

Đạt điểm 1,00 trên 1,00

Cho đoạn chương trình sau, biết rằng các hàm push, pop, empty hoạt động theo cơ chế **stack**, enqueue, dequeue hoạt động theo cơ chế của **queue**.

```
void func(Queue &Q) {
    Stack S;
    while (!empty(Q)) {
        push(S, front(Q));
        dequeue(Q);
    }
    while(!empty(S)){
        enqueue(Q, top(S));
        pop(S);
    }
}

int main(){
    Queue Q;
    enqueue(Q, 1);
    enqueue(Q, 2);
    enqueue(Q, 3);
    func(Q);
    // ...
}
```

Hỏi kết quả của biến Q sau khi gọi hàm func:

- ☐ a. [1 (front), 2]
- ☒ b. Q=[3 (front), 2, 1] ✓
- ☐ c. Q=[] (hàng đợi rỗng)
- ☐ d. Q=[1 (front), 2, 3]

Your answer is correct.

The correct answer is:

Q=[3 (front), 2, 1]

Câu hỏi 5

Đúng

Đạt điểm 1,00 trên 1,00

Chuyển biểu thức infix (trung tố):  $( 4 * 3 ) / ( 6 * 2 )$  thành biểu thức postfix (hậu tố).

Lưu ý khi ghi chuỗi kết quả: mỗi giá trị ghi cách nhau 1 khoảng trắng và không có khoảng trắng đầu và cuối chuỗi kết quả.

Answer:  ✓

Bước	Token	Stack	Output
1	(	(	
2	4	(	4
3	*	( *	4
4	3	( *	4 3
5	)		4 3 *
6	/	/	4 3 *
7	(	/(	4 3 *
8	6	/(	4 3 * 6
9	*	/( *	4 3 * 6
10	2	/( *	4 3 * 6 2
11	)	/	4 3 * 6 2 *
12		Empty	4 3 * 6 2 * /

The correct answer is: 4 3 6 2 /

Câu hỏi 6

Đúng

Đạt điểm 1,00 trên 1,00

Chuyển biểu thức infix (trung tố):  $2 ^ { ( 7 - 5 ) } * 3 ^ { 8 } + ( 2 + 3 ) ^ { 4 }$  thành biểu thức postfix (hậu tố). Cho biết **stack** tại thời điểm xét xong **token 8**.

- ☐ a.  $[-, ^ { (top)}]$
- ☒ b.  $[*, ^ { (top)}]$  ✓
- ☐ c.  $[^, ( (top)]$
- ☐ d.  $[*, ( (top)]$

Your answer is correct.

Bước	Token	Stack	Output
1	2	Empty	2
2	^	^	2
3	(	^ (	2
4	7	^ (	2 7
5	-	^ ( -	2 7
6	5	^ ( -	2 7 5
7	)	^	2 7 5 -
8	*	*	2 7 5 - ^
9	3	*	2 7 5 - ^ 3
10	^	* ^	2 7 5 - ^ 3
11	8	* ^	2 7 5 - ^ 3 8
12	+	+	2 7 5 - ^ 3 8 ^ *
13	(	+ (	2 7 5 - ^ 3 8 ^ *
14	2	+ (	2 7 5 - ^ 3 8 ^ * 2
15	+	+ ( +	2 7 5 - ^ 3 8 ^ * 2
16	3	+ ( +	2 7 5 - ^ 3 8 ^ * 2 3
17	)	+	2 7 5 - ^ 3 8 ^ * 2 3 +
18	^	+ ^	2 7 5 - ^ 3 8 ^ * 2 3 +
19	4	+ ^	2 7 5 - ^ 3 8 ^ * 2 3 + 4
20		Empty	2 7 5 - ^ 3 8 ^ * 2 3 + 4 ^ +

The correct answer is:

$[*, ^ { (top)}]$

**Câu hỏi 7**

Đúng

Đạt điểm 1,00 trên 1,00

Tính giá trị biểu thức hậu tố sau:  $3\ 8\ ^\wedge\ 2\ 4\ ^\wedge\ +$ .

Answer: 

Bước	Token	Stack
1	3	3
2	8	3 8
3	$^\wedge$	6561
4	2	6561 2
5	4	6561 2 4
6	$^\wedge$	6561 16
7	+	6577

The correct answer is: 6577

**Câu hỏi 8**

Đúng

Đạt điểm 1,00 trên 1,00

Chuyển biểu thức infix (trung tố): **A and B or C** thành biểu thức postfix (hậu tố).

**Lưu ý khi ghi chuỗi kết quả: mỗi giá trị ghi cách nhau 1 khoảng trắng. Không có khoảng trắng đầu và cuối chuỗi kết quả. Kết quả phân biệt chữ hoa chữ thường**

Answer: A B and C or



Bước	Token	Stack	Output
1	A		A
2	and	and	A
3	B	and	A B
4	or	or	A B and
5	C	or	A B and C
6		Empty	A B and C or

The correct answer is: A B and C or

**Câu hỏi 9**

Đúng

Đạt điểm 1,00 trên 1,00

Chuyển biểu thức infix (trung tố): **A + B / C + D** thành biểu thức postfix (hậu tố). Cho biết stack tại thời điểm xét xong **token C**. Kéo thả các kết quả đúng vào các ô trống bên dưới:

stack: [  ✓ ,  ✓ (top)]

Your answer is correct.

The correct answer is:

Chuyển biểu thức infix (trung tố): **A + B / C + D** thành biểu thức postfix (hậu tố). Cho biết stack tại thời điểm xét xong **token C**. Kéo thả các kết quả đúng vào các ô trống bên dưới:

stack: [ [+], [/] (top)]

◀ Quiz 6: DSLK (unlimited) (Cô ko tách ra theo nội dung đã học hôm nay, nên các em chỉ cần làm các câu có thể làm được, còn các câu có code có thể làm sau, cô cho thời gian qua tuần sau để hoàn thành)

Chuyển tới...



