W02 - Arrays and Array-based Structures results for Ali Asad

Correct answers are hidden.

Score for this attempt: **20** out of 20

Submitted 27 Jan at 8:40

This attempt took 1 minute.

Question 1

2 / 2 pts

For a *List* of size *n* implemented as an *ArrayStack*, how many shifts take place for a valid *insert* at index *i*?

n-i

Question 2

2 / 2 pts

For a *List* of size *n* implemented as an *ArrayStack*, how many shifts take place for a valid *remove* at index *i*?

n-i-1

Question 3

2 / 2 pts

What is the cost in O-notation of 1 shift in an *ArrayStack*?

O(1)

Question 4

2 / 2 pts

What is the cost in O-notation of S shifts in an *ArrayStack*?

O(S)

Question 5

2 / 2 pts

Which of the following are a characteristic of arrays:

Mark all that apply.

☑ random access

☑ static

☐ dynamic

☐ heterogenous

☑ linear

☑ homogenous

Question 6

2 / 2 pts

The base address of an array:

☑ is the address of the first element of the array.

□ is the address of the last element of the array.

□ does not point to any particular array element.

Question 7

2 / 2 pts

Arrays occupy contiguous cells in memory which:

☑ helps in randomly accessing an array element.

□ allows them to be dynamic.

☑ restricts them to be homogenous.

□ allows them to be heterogenous.

☑ requires them to be of fixed size.

Question 8

2 / 2 pts

Given the following sequence of operations on a deque,

```
add_first(1)
add_last(6)
add_first(0)
```

indicate the numbers when the following operations are performed on the deque in the given order.

remove_last(): 6
remove_first(): 0
remove_last(): 1

Answer 1:

6

Answer 2:

0

Answer 3:

1

Question 9

2 / 2 pts

Which of the following statements are true about Abstraction?

☑ It exposes relevant information to use a component/class.

☐ It exposes in-depth implementation details.

☑ It gives the flexibility to change the implementation without changing the interface.

☐ It tightly couples interface with implementation.

☑ It lets you focus on the higher/conceptual view.

Question 10

2 / 2 pts

DEQUE is a double-ended queue that can also be used as a *Stack* by:

☑ pushing and popping from the front end.

☑ pushing and popping from the back end.

□ Pushing from one end and popping from the other end.

□ None of the above

Quiz score: **20** out of 20