

Comenzado en Thursday, 24 de February de 2022, 07:02

Estado Terminados

Finalizado en Thursday, 24 de February de 2022, 08:57

Tiempo empleado 1 hora 55 mins

Calificación 74.00 de un total de 100.00

Pregunta **1**

Completada

Puntúa 0.00 sobre 3.00

Técnica de asignación de memoria en donde cada partición, en vez de corresponder a un programa completo, corresponde a un segmento de él.

- ☐ a. Particiones estáticas
- ☐ b. Memoria virtual
- ☐ c. Paginación
- ☐ d. Segmentación
- ☒ e. Particiones dinámicas

Pregunta **2**

Completada

Puntúa 3.00 sobre 3.00

Técnicas de asignación de memoria a programas por parte del sistema operativo

- ☒ a. Segmentación
- ☐ b. Ninguna es correcta
- ☒ c. Memoria virtual
- ☒ d. Paginación
- ☒ e. Particiones dinámicas
- ☒ f. Particiones estáticas



Pregunta **3**

Completada

Puntúa 3.00 sobre 3.00

Forma de liberar la memoria principal de los procesos que han finalizado

- ☐ a. Allocation memory
- ☒ b. Deallocation memory
- ☐ c. Virtualización
- ☐ d. Ninguna es correcta

Pregunta **4**

Completada

Puntúa 3.00 sobre 3.00

Componente encargado de la transformación de direcciones virtuales en direcciones físicas

- ☒ a. Unidad de Gestión de Memoria
- ☐ b. Directorio Virtual
- ☐ c. Ninguna es correcta
- ☐ d. Gestor de Memoria Física
- ☐ e. Unidad Lógica Aritmética

Pregunta **5**

Completada

Puntúa 3.00 sobre 3.00

Sistema encargado de liberar memoria que no está siendo utilizada por ningún proceso

- ☐ a. Unidad de Gestión de Memoria
- ☐ b. Deallocation memory
- ☒ c. Recolector de basura
- ☐ d. Ninguna es correcta



Pregunta **6**

Completada

Puntúa 3.00 sobre 3.00

Técnica de asignación de memoria en donde la memoria se divide en particiones con la misma capacidad para contener procesos

- ☐ a. Memoria virtual
- ☐ b. Particiones dinámicas
- ☒ c. Particiones estáticas
- ☐ d. Segmentación
- ☐ e. Paginación

Pregunta **7**

Completada

Puntúa 3.00 sobre 3.00

Proceso mediante el cual a los programas se les asigna espacio de memoria física o virtual

- ☐ a. Virtualización
- ☐ b. Ninguna es correcta
- ☐ c. Deallocation memory
- ☒ d. Allocation memory

Pregunta **8**

Completada

Puntúa 3.00 sobre 3.00

Técnica de asignación de memoria en donde se presenta al usuario una memoria principal mayor que la memoria física real

- ☐ a. Paginación
- ☐ b. Segmentación
- ☐ c. Particiones estáticas
- ☒ d. Memoria virtual
- ☐ e. Particiones dinámicas



Pregunta **9**

Completada

Puntúa 3.00 sobre 3.00

Técnica de asignación de memoria en donde los programas se consideran divididos en zonas consecutivas

- ☐ a. Segmentación
- ☐ b. Particiones estáticas
- ☐ c. Memoria virtual
- ☐ d. Particiones dinámicas
- ☒ e. Paginación

Pregunta **10**

Completada

Puntúa 0.00 sobre 3.00

Técnica de asignación de memoria en donde los programas son introducidos por el SO en posiciones consecutivas de memoria.

- ☐ a. Particiones estáticas
- ☐ b. Particiones dinámicas
- ☐ c. Segmentación
- ☐ d. Memoria virtual
- ☒ e. Paginación



Pregunta **11**

Completada

Puntúa 5,00 sobre 5,00

Describe los elementos claves de un nodo de una lista doblemente enlazada.

1	2	3
---	---	---

Anterior : hace referencia al nodo anterior.

Información : Datos guardados en el nodo .

Siguiente: hace referencia la nodo siguiente .



Pregunta **12**

Completada

Puntúa 5.00 sobre 5.00

¿Cuál es la salida del siguiente segmento de código, teniendo en cuenta que el tipo de dato de la pila es int?

```
.....  
Pila p = new Pila();  
int x = 4, y;  
p.push(x);  
System.out.println("\n" + p.top());  
y = p.pop();  
p.push(32);  
p.push(p.pop());  
do{  
    System.out.println("\n" + p.pop());  
} while (!p.isEmpty());
```

4

32

Pregunta **13**

Completada

Puntúa 5.00 sobre 5.00

¿Qué tipos de listas enlazadas existe?

simplemente enlazada

doblemente enlazada

circulares simple .

doblemente circulares.



Pregunta **14**

Completada

Puntúa 5,00 sobre 5,00

¿Por qué se dice que la lista es dinámica y un arreglo es estático?

El arreglo tiene una posición reservada desde el inicio y no se puede modificar(tiene tamaño definido)

La lista ocupa memoria a medida que la va necesitando (no tiene tamaño predefinido)

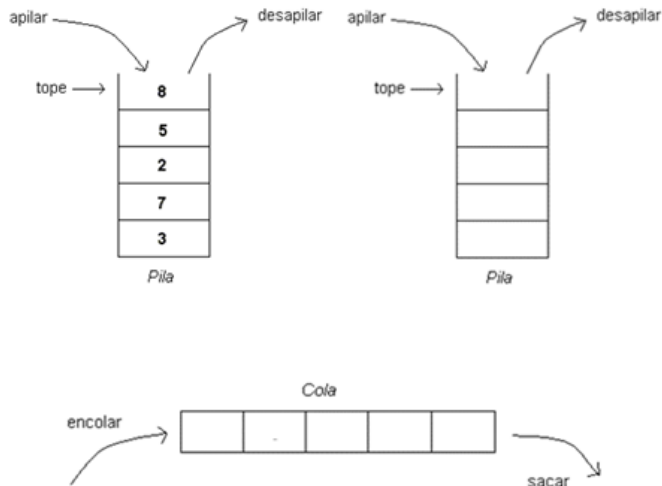


Pregunta **15**

Completada

Puntúa 20.00 sobre 20.00

Imagine que posee dos TDA Pila (LIFO) y desea utilizarlas para simular un TDA Cola (FIFO). Explique detalladamente cada uno de los pasos de la operatoria a ejecutar para realizar dicha simulación utilizando los métodos push y pop de las pilas para simular los métodos enqueue y dequeue de la cola.



métodos push o enqueue : agrega un elemento a la lista

método pop o dequeue : elimina elementos de la pila

Simulación :

en una de las pilas se usa para almacenar los datos almacenados,

y la otra se usa para invertir los datos para realizar la salida.

Así en el método eliminar se guardan todos los datos en una de las listas y se elimina el primer ejemplo

lista a:12345

eliminando : 5

lista b 5432

y para agregar se pasan los datos a una lista comenzando con el dato a guardar

lista a :1234

agregar 5

lista b: 51234

Pregunta **16**

Completada

Puntúa 10.00 sobre 30.00

Se le solicita que implemente un TDA lista enlazada simple en la cual se registraran contactos telefónicos. De los contactos se requiere almacenar: su nombre, teléfono fijo, teléfono de trabajo y teléfono móvil. La inserción dentro de la lista debe realizarse de forma alfabética por el campo nombre. Coloque el código asociado a la solución.

class ListaSimple:

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

```
    self.inicio = None
```

```
    #insercion
```

```
#Creacion de nuevo objeto
```

```
def insertar(self, nombre, teléfono fijo, teléfono de trabajo y teléfono móvil):
```

```
    nuevo = Contacto(nombre, familia,,nombre, teléfono fijo, teléfono de trabajo y teléfono móvil)
```

```
    #verifica la incersion
```

```
    if self.inicio is None:
```

```
        self.inicio = nuevo
```

```
    else:
```

```
        tmp = self.inicio
```

```
        while tmp.siguiente is not None:
```

```
            tmp = tmp.siguiente
```

```
            tmp.siguiente = nuevo
```

```
#clase nodo de los contactos agregados
```

```
class NodoContactos :
```

```
def __init__(self, nombre, teléfono fijo, teléfono de trabajo y teléfono móvil):
```

```
    self.nombre = nombre
```

```
    self.telefono fijo = telefono fijo
```

```
    self.telefono de trabajo = telefono de trabajo
```

```
    self.telefono movil= telefono movil
```

```
    self.nombre = nombre
```

```
    self.siguiente = None
```



[◀ Avisos Generales](#)

Ir a...

[2do. Parcial ▶](#)

