

Práctica 2. Estructuras de Datos

P2.1. Implementa una cola de enteros (First In First Out) en C. El programa debe presentar un menú en el que las opciones sean:

1. Encolar un entero
2. Desencolar un entero
3. Mostrar el valor del primer elemento de la cola
4. Comprobar si la cola está vacía
5. Vaciar una cola
6. Salir del programa

Ejemplo:

Menu de la cola:

1 – Encolar; 2 – Desencolar; 3 – Mostrar; 4 – Comprobar; 5 – Vaciar; 6 – Salir

1

Introduce un elemento: 5

1 – Encolar; 2 – Desencolar; 3 – Mostrar; 4 – Comprobar; 5 – Vaciar; 6 – Salir

3

Primer elemento: 5

1 – Encolar; 2 – Desencolar; 3 – Mostrar; 4 – Comprobar; 5 – Vaciar; 6 – Salir

2

1 – Encolar; 2 – Desencolar; 3 – Mostrar; 4 – Comprobar; 5 – Vaciar; 6 – Salir

2

La cola esta vacia, no se puede desencolar

1 – Encolar; 2 – Desencolar; 3 – Mostrar; 4 – Comprobar; 5 – Vaciar; 6 – Salir

6

P2.2. Desarrolla un programa que, mediante una estructura de datos dinámica, compruebe si en un texto los paréntesis existentes están bien balanceados o no.

Ejemplo:

Introduce un texto: La (casa (es)) (verde)

Los parentesis SI estan balanceados

Introduce un texto: (la) (casa es)) verde(

Los parentesis NO estan balanceados

Introduce un texto: (La casa(es (verde))

Los parentesis NO estan balanceados

P2.3. Implementa una cola de prioridad. Una cola de prioridad es una estructura de datos en la que los elementos se atienden en el orden indicado por una prioridad asociada a cada uno (un número de prioridad alto implica una mayor prioridad, y por lo tanto, debe ser atendido antes que los de prioridad más baja). Si varios elementos tienen la misma prioridad, se atenderán de modo convencional de una cola (First In First Out). El programa debe presentar un menú en el que las opciones sean:

1. Encolar un elemento
2. Mostrar el primer elemento y desencolarlo
3. Comprobar si la cola está vacía
4. Vaciar la cola
5. Salir del programa

Ejemplo:

Menu de la cola:

1 – Encolar; 2 – Desencolar; 3 – Comprobar; 4 – Vaciar; 5 – Salir

1

Introduce un numero y su prioridad: 1 3
Menu de la cola:
1 – Encolar; 2 – Desencolar; 3 – Comprobar; 4 – Vaciar; 5 – Salir
1
Introduce un numero y su prioridad: 2 3
Menu de la cola:
1 – Encolar; 2 – Desencolar; 3 – Comprobar; 4 – Vaciar; 5 – Salir
1
Introduce un numero y su prioridad: 3 4
Menu de la cola:
1 – Encolar; 2 – Desencolar; 3 – Comprobar; 4 – Vaciar; 5 – Salir
4
Desencolando elemento 3 con prioridad 4
Desencolando elemento 1 con prioridad 3
Desencolando elemento 2 con prioridad 3
Menu de la cola:
1 – Encolar; 2 – Desencolar; 3 – Comprobar; 4 – Vaciar; 5 – Salir
5

P2.4. Implementa una lista ordenada doblemente enlazada de enteros en C. El programa debe presentar un menú en el que las opciones sean:

1. Insertar un elemento en la lista
2. Eliminar un elemento coincidente con un valor
3. Comprobar si un elemento está en la lista
4. Mostrar la lista en orden ascendente
5. Mostrar la lista en orden descendente
6. Salir del programa

Ejemplo:

Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
1
Introduce un numero: 5
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
1
Introduce un numero: 3
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
1
Introduce un numero: 7
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
3
Introduce un numero: 5
El elemento 5 esta en la lista
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
4
3 5 7
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir
5
7 5 3
Menu de la lista ordenada doblemente enlazada:
1 – Insertar; 2 – Eliminar; 3 – Buscar; 4 – Mostrar ascendente; 5 – Mostrar descendente; 6 – Salir

P2.5. Implementa un árbol binario en C para almacenar una expresión matemática binaria. La operación llega en notación postfija (primero los argumentos y después el operando) y debemos almacenarla. A partir del árbol, debemos mostrar la operación en notación infija (primer argumento, operador, segundo argumento) utilizando paréntesis para agrupar suboperaciones. Ejemplo:

Introduce la operación: 5 3 + 6 2 * - 4 -

La operación es $((5+3)-(6*2))-4$

P2.6. Implementa el algoritmo heapsort para ordenar un conjunto de números enteros mediante un montículo.

Ejemplo:

Introduce el numero de elementos: 8

Introduce los elementos: 8 4 2 1 6 3 7 4

Los elementos ordenados son: 1 2 3 4 4 6 7 8

P2.7. Cuando un seto crece, unas ramas se alargan más que otras y, al final, siempre quedan puntas que afean el conjunto y que el jardinero debe recortar hasta el tamaño del resto. De la misma manera, un árbol binario puede crecer de forma desigual, y por tanto no ser completo. Escribir un programa que genere un árbol binario. A partir de ese árbol, escribir una función que recorte todas las ramas que sean más largas que las demás, de tal forma que se obtenga un árbol binario lleno. Por último, mostrar este árbol por pantalla mediante un recorrido en profundidad.

Ejemplo:

Introduce elementos para el árbol binario. Finalizar con 0:

15 8 12 21 23 3 14 17 20 22 26 13 0

El árbol podado es: 15 8 3 12 21 17 23

P2.8. Escribir un programa que permita almacenar varias secuencias de números enteros. Por ejemplo, podemos tener las siguientes secuencias:

16, 49, 21, 2, 58, 15

50, 71, 23, 40, 81, 34, 23, 75

23, 53, 25, 15

45, 90, 12

Para ello, se propone utilizar una lista de listas. Una vez almacenadas todas ellas, escribir una función que calcule el mínimo de los máximos de las secuencias. Para el ejemplo anterior, los máximos de cada secuencia son 58, 81, 53, 90, y el mínimo de todos ellos es 53.

Escribir además un programa principal que permita comprobar el funcionamiento.

Ejemplo:

Introduce el número de listas: 4

Para la lista 1, introduce los elementos y termina con 0: 16 49 21 2 58 15 0

Para la lista 2, introduce los elementos y termina con 0: 50 71 23 40 81 34 23 75 0

Para la lista 3, introduce los elementos y termina con 0: 23 53 25 15 0

Para la lista 4, introduce los elementos y termina con 0: 45 90 12

El minimo de los máximos es 53