#### Trabajo Práctico N° 1: Pilas

Codificar los siguientes programas. En el caso de ser necesario se pueden utilizar pilas auxiliares.

1. Cargar desde el teclado una pila DADA con 5 elementos. Pasar los tres primeros elementos a la pila AUX1 y los dos restantes a la pila AUX2, ambas pilas inicializadas en vacío.

2. Cargar desde el teclado la pila ORIGEN e inicializar en vacío la pila DESTINO. Pasar todos los elementos de la pila ORIGEN a la pila DESTINO.

3. Cargar desde teclado una pila DADA y pasar a la pila DISTINTOS todos aquellos elementos distintos al valor 8.

4. Cargar desde el teclado la pila ORIGEN e inicializar en vacío la pila DESTINO. Pasar los elementos de la pila ORIGEN a la pila DESTINO, pero dejándolos en el mismo orden.

5. Cargar desde el teclado la pila DADA. Invertir la pila de manera que DADA contenga los elementos cargados originalmente en ella, pero en orden inverso.

6. Pasar el primer elemento (tope) de la pila DADA a su última posición (base), dejando los restantes elementos en el mismo orden.

7. Pasar el último elemento (base) de la pila DADA a su primera posición (tope), dejando los restantes elementos en el mismo orden.

8. Repartir los elementos de la pila MAZO en las pilas JUGADOR1 y JUGADOR2 en forma alternativa.

9. Comparar la cantidad de elementos de las pilas A y B. Mostrar por pantalla el resultado.

10. Comparar las pilas A y B. Mostrar por pantalla el resultado.

11. Suponiendo la existencia de una pila MODELO que no esté vacía, eliminar de la pila DADA todos los elementos que sean iguales al tope de la pila MODELO.

12. Suponiendo la existencia de una pila MODELO (vacía o no), eliminar de la pila DADA todos los elementos que existan en MODELO.

13. Suponiendo la existencia de una pila LÍMITE, pasar los elementos de la pila DADA que sean mayores o iguales que el tope de LIMITE a la pila MAYORES, y los elementos que sean menores a la pila MENORES.

14. Determinar si la cantidad de elementos de la pila DADA es par. Si es par, pasar el elemento del tope de la pila AUX a la pila PAR y si es impar pasar el tope a la pila IMPAR.

15. Cual es la condición del siguiente ciclo? .Cuando finaliza el ciclo? (Pila1, Pila2, y Descarte son pilas):

|  |
| --- |
| **while (!pilavacia(&Pila1)) {   apilar (&Pila2, desapilar(&Descarte)) }** |

La condición del ciclo es apilar los elementos de “Pila1” a “Pila2”, pero no se va a poder cumplir con el ciclo ya que en el código figura que se va a desapilar la pila “Descarte”, por ende, el ciclo va a ser infinito.

Para que se cumpla el ciclo correctamente tiene que ser así:

**while (!pilavacia(&Pila1)) {   
 apilar (&Pila2, desapilar(&Pila1))  
}**

Por su puesto con las pilas declaradas e inicializadas.

16. Qué realiza el siguiente código escrito en lenguaje C (*Pila1*, *Aux* y *Result* son pilas):

|  |
| --- |
| **while (!pilavacia(&Pila1)) {   if (tope(&Pila1) == 5) {  apilar (&Aux, desapilar(&Pila1));**  **apilar (&Result, desapilar(&Aux));**  **} }** |

Lo que realiza el código es apilar los elementos iguales al tope de la pila “Pila1” a la pila “Aux”, luego los elementos de la pila “Aux” se desapilan y se apilan nuevamente en la pila “Result”. Pero lo correcto seria así:

**while (!pilavacia(&Pila1)) {   
 if (tope(&Pila1) == 5) {  
 apilar (&Aux, desapilar(&Pila1));**

**}else{**

**apilar (&Result, desapilar(&Pila1));**

**}  
}**

17. Para el ejercicio “Cargar por teclado una pila ORIGEN y pasar a la pila DISTINTO todos aquellos elementos que preceden al valor 5 (elementos entre el tope y el valor 5). No se asegura que exista algún valor 5”, se realizo el siguiente programa:

// este programa carga por teclado una pila Origen y pasa a la pila Destino todos aquellos elementos que preceden el valor 5

|  |
| --- |
| **int main() {   Pila Origen, Distinto;  inicpila(&Origen);  inicpila(&Distinto);  leer(&Origen);  if (!pilaVacia(&Origen)){   while (tope(&Origen) <> 5) {  apilar (&Distinto, desapilar(&Origen));  }   } }** |

1. .Resuelve el problema planteado?

No, porque para comparar un elemento se usa condición, sino el programa no va a funcionar correctamente.

1. .Cuales son los errores que encuentra?

La condición está primero y luego el ciclo esta abajo, la condición no es correcta, lo correcto sería “if(tope(&Origen) >= 5) || tope(&Origen) <= 5)…

// sentencia //

1. Reescribir el código para que resuelva adecuadamente el problema planteado.

**int main() {   
 Pila Origen, Distinto;  
 inicpila(&Origen);  
 inicpila(&Distinto);  
 leer(&Origen);  
 while (!pilaVacia(&Origen)){   
 if(tope(&Origen) <= 5) || if(tope(&Origen) >= 5 {  
 apilar (&Distinto, desapilar(&Origen));  
 }   
 }  
}**

d. Indicar las componentes del programa.

**int main() { ////  
 Pila Origen, Distinto; // Son el tipo de dato, en este caso son un tipo de dato PILA. //   
 inicpila(&Origen);   
 inicpila(&Distinto); // esta dos funciones permite inicializar las pilas sin basura //   
 leer(&Origen); // esta función permite ingresar datos desde el teclado //   
 while (!pilaVacia(&Origen)){ // Ciclo “mientras”… //  
 if(tope(&Origen) <= 5) || if(tope(&Origen) >= 5 { // Condicion //  
 apilar (&Distinto, desapilar(&Origen)); // Funcion apilar y desapilar //  
 }   
 }**

**}**

18. Dado el siguiente ciclo (*Pila1*, *Pila2* y *Descarte* son pilas):

|  |
| --- |
| **while ( (!pilaVacia(&Pila1)) && (!pilaVacia(&Pila2)) ) {   apilar (&Descarte, desapilar(&Pila1));  apilar (&Descarte, desapilar(&Pila2)) }** |

1. .Cual es la condición del ciclo? Explique con sus palabras:

La condición del ciclo es que si las pilas “Pila1” y “Pila2” no están vacías, los elementos de ambas

pilas se van a pasar a la pila “Descarte.

1. .Cuales son los posibles estados de ambas pilas al finalizar el ciclo?

Al finalizar el ciclo ambas pilas van a quedar vacías.