

# **Algoritmos y Estructuras de Datos Avanzadas (Problemas)**

Estructuras de Datos Avanzadas con  
Programación Orientada a Objetos

## Códigos fuente:

Los códigos fuente de los problemas se encuentran disponibles en el siguiente enlace:

<https://github.com/ull-cs/aeda/tree/main/c%2B%2B/problems>



## Problema 1:

Escriba un programa que cambie cada una de las letras presentes en una cadena de texto por la letra siguiente del alfabeto.

```
$ ./next_letter abecedario  
bcfdfebsjp
```

```
$ ./next_letter Abracadabra  
Bcsbdbebcsb
```

[next\\_letter.cc](#)



## Problema 2:

Escriba un programa que solicite al usuario una cantidad de números y calcule su media.

```
$ ./average_of_numbers 5
Enter 5 items:
27
13
41
-51
9
You entered: 27 13 41 -51 9
Average: 7.8
```

[average\\_of\\_numbers.cpp](#)



### Problema 3:

Escriba un programa que calcule el tiempo requerido para calcular el factorial de un número mediante una función convencional y una en línea.

```
$ ./measure_time 250000  
Midiendo tiempo mediante función convencional:  
Tiempo: 10.3654 ms.  
Midiendo tiempo mediante función en línea:  
Tiempo 3.1079 ms.
```

[measure\\_time.cc](#)



## Problema 4:

Escriba un programa que utilizando una macro muestre el valor de una variable introducida por el usuario es par o impar.

```
$ ./multiline_macro  
Enter a number: 10  
10 is Even
```

[multiline\\_macro.cc](#)



## Problema 5:

Escriba un programa que solicite al usuario un número, cree una pila compuesta por números aleatorios y la vacíe.

```
$ ./stack 5  
4 has been pushed  
7 has been pushed  
8 has been pushed  
6 has been pushed  
4 has been pushed  
Elements present in stack: 4 6 8 7 4
```

stack/



## Problema 6:

Escriba un programa que solicite al usuario un número, cree una cola de prioridad compuesta por elementos con prioridades aleatorias y la vacíe.

```
$ ./priority_queue 4
94 with priority 6 has been queued
84 with priority 7 has been queued
87 with priority 3 has been queued
78 with priority 6 has been queued
Elements present in queue:
84 has the highest priority
94 has the highest priority
78 has the highest priority
87 has the highest priority
```

[priority\\_queue/](#)





## Problema 7:

Escriba un programa que solicite al usuario un número, cree una cola doble (deque) compuesta por elementos aleatorios y la vacíe.

```
$ ./deque 4  
4 has been inserted from front  
8 has been inserted from rear  
4 has been inserted from rear  
7 has been inserted from front  
Rear element: 4  
Rear element: 8  
Front element: 7  
Rear element: 4
```

[deque/](#)



## Problema 8:

Escriba un programa que solicite al usuario un número, cree una cola mediante plantillas compuesta por números aleatorios y la vacíe.

```
$ ./queue 5  
8 has been pushed  
1 has been pushed  
5 has been pushed  
3 has been pushed  
5 has been pushed  
Elements present in queue: 8 1 5 3 5
```

queue/



## Problema 9:

Escriba un programa que cree una lista enlazada mediante plantillas e implemente las operaciones básicas sobre ella.

```
---- Link list items ----  
49 | 27 | 18 | 72 |  
-----
```

[linked\\_list/](#)



## Problema 10:

Escriba un programa que solicite al usuario un número, cree una lista de asociación con pares clave-valor aleatorios y muestre la lista.

```
$ ./associationlist 5
{10,47} has been added
{3,87} has been added
{9,83} has been added
{4,51} has been added
{8,76} has been added
Elements present in list:
----List items-----
{10,47} | {3,87} | {9,83} | {4,51} | {8,76} |
-----
```

association\_list/



## Problema 11:

Escriba un programa que compruebe si una expresión compuesta por corchetes, paréntesis y llaves está balanceada.

```
$ ./balanced_brackets "{}"  
Expression is balanced  
$ ./balanced_brackets "[({}[]())]"  
Expression is balanced  
$ ./balanced_brackets "{()}[]{"  
Expression is not balanced  
$ ./balanced_brackets "([({}[]())})"  
Expression is not balanced
```

[balanced\\_brackets/](#)



## Problema 12:

Escriba un programa que convierta una expresión dada en notación infija en notación postfija.

```
$ ./infix_to_postfix "a+b*(c^d-e)^(f+g*h)-i"  
abcd^e-fgh*+^*+i-
```

```
$ ./infix_to_postfix "A+B*C/(E-F)"  
ABC*EF-/+
```

[infix\\_to\\_postfix/](#)



## Problema 13:

Escriba un programa que solicite al usuario un número, cree una pila compuesta por números aleatorios y la vacíe. La pila debe permitir consultar el valor mínimo almacenado.

```
$ ./special_stack 4  
8 has been pushed  
5 has been pushed  
3 has been pushed  
4 has been pushed  
Elements present in stack: 4 (min=3) 3 (min=3) 5 (min=5) 8 (min=8)
```

[special\\_stack/](#)



## Problema 14:

Escriba un programa que implemente conjuntos disjuntos. Implemente sus funcionalidades básicas.

```
$ ./disjoint_set
Elements 0 and 2 are included into the same set.
Elements 4 and 2 are included into the same set.
Elements 3 and 1 are included into the same set.
Elements 4 and 0 belongs to the same set.
Elements 1 and 0 do not belong to the same set.
```

[disjoint\\_set/](#)





## Problema 15:

Dada una expresión balanceada que puede contener paréntesis abiertos y cerrados, (), comprobar si contiene paréntesis duplicados.

```
$ ./duplicated_parenthesis "(x+y)"  
Expression '(x+y)' does not have duplicated parenthesis.
```

```
$ ./duplicated_parenthesis "((x+y))+z"  
Expression '((x+y))+z' has duplicated parenthesis.
```

```
$ ./duplicated_parenthesis "(x+y)+()  
Expression '(x+y)+()' has duplicated parenthesis.
```

```
$ ./duplicated_parenthesis "((x+y)+((z)))"  
Expression '((x+y)+((z)))' has duplicated parenthesis.
```

[duplicated\\_parenthesis/](#)



## Problema 15:

Se puede emplear una pila para resolver el problema. La idea es recorrer la expresión dada y

- Si el carácter no es `)`, añadir el carácter a la pila.
- Si el carácter es `)`, comprobar si el carácter en la parte alta de la pila es `(`:
  - Si es `(`, la subexpresión que acaba en el `)` tiene la forma  
**`((subexpresión))`**
  - Si no es `(`, sacar caracteres de la pila hasta encontrar `(`.

[duplicated\\_parenthesis/](#)



## Problema 16:

Dado un número **n**, generar números binarios entre **1** y **n** empleando una cola.

```
$ ./binary_numbers "5"
```

```
1 10 11 100 101
```

```
$ ./binary_numbers "10"
```

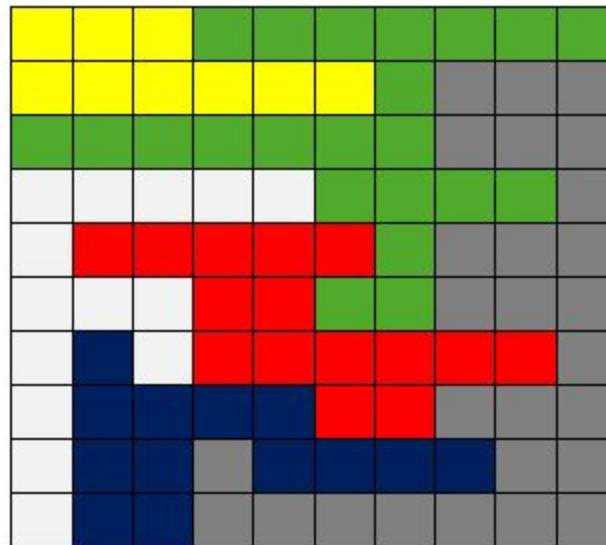
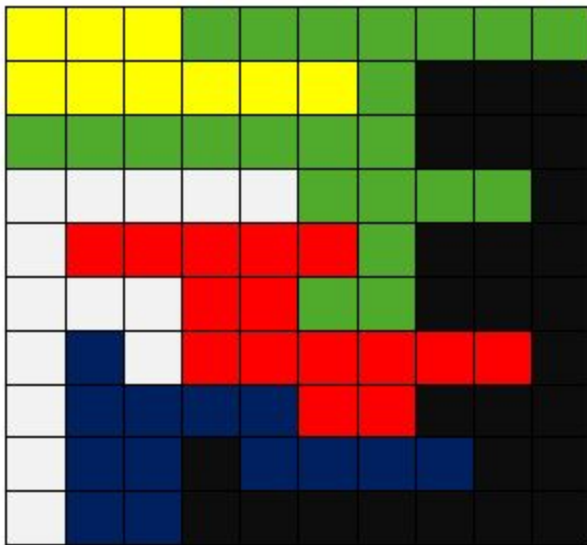
```
1 10 11 100 101 110 111 1000 1001 1010
```

[binary\\_numbers/](#)



## Problema 17:

Dada una matriz compuesta por colores, una posición de comienzo y un color, modificar el color de la matriz a partir de la posición de comienzo.



[flood\\_fill/](#)



## Problema 15:

Sparse table...

```
$ ./associationlist 5
{10,47} has been added
{3,87} has been added
{9,83} has been added
{4,51} has been added
{8,76} has been added
Elements present in list:
----List items-----
{10,47} | {3,87} | {9,83} | {4,51} | {8,76} |
-----
```

balanced\_brackets.cc



