

Hoja 4

Importante: Los ejercicios deben entregarse a través de web (**Domjudge y Blackboard**). Cada ejercicio deberá ir en un fichero con nombre:

mergesort.cpp (Utiliza como base el fichero proporcionado en Blackboard)

combinacionesDinamica.cpp

La fecha de entrega: consultar la página de la actividad en blackboard

Mergesort (3,5 puntos): Implementa un algoritmo que permita ordenar un array de enteros utilizando el algoritmo de ordenamiento por mezcla que es una solución basada en una estrategia divide y vencerás. Puedes encontrar una descripción de este algoritmo en la siguiente dirección:

https://es.wikipedia.org/wiki/Ordenamiento_por_mezcla

(Fíjate en la animación, la cual es muy ilustrativa).

El **input** constará de dos partes. En primer lugar, con formato "%d\n" se indicará la cantidad N de enteros que se introducirán a continuación. La segunda parte de la entrada será una lista de enteros separados por espacios.

El **output** será una línea por cada iteración del algoritmo de ordenación en el que se mostrará una lista de números mostrando la ordenación parcial. Esta lista se imprimirá después de realizar la combinación de las dos listas.

OJO: Observad que existe un espacio después de cada número incluso en el caso del último.

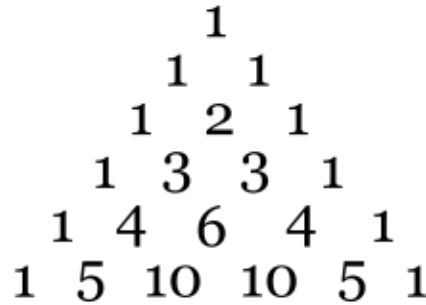
Input:

Output:

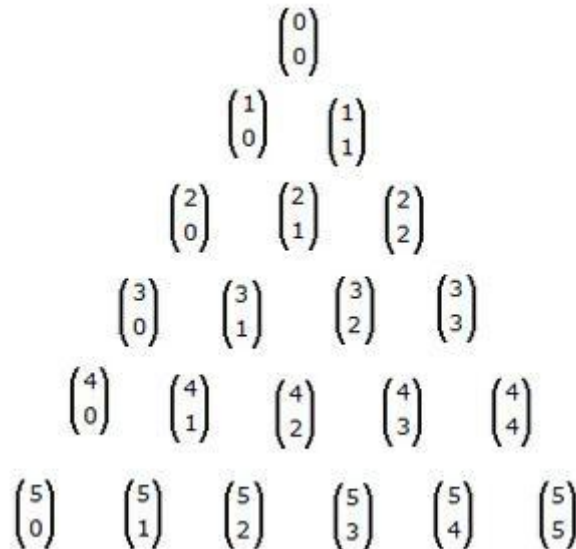
10	8 9
9 8 7 6 5 4 3 2 1 0	5 6
	5 6 7
	5 6 7 8 9
	3 4
	0 1
	0 1 2
	0 1 2 3 4
	0 1 2 3 4 5 6 7 8 9

Algoritmos y Estructuras de Datos

Combinaciones Dinamica (3,5 Puntos): Escribe un programa para calcular las posibles combinaciones de n elementos tomados en grupos de r usando una solución basada en programación dinámica. Esta solución estará basada en el bien conocido triángulo de Tartaglia o Pascal que puede observarse en la siguiente imagen



Cuyos valores corresponden a los siguientes números combinatorios:



El **input** se leerá con el formato "%d %d". El primer entero estará referido al valor de n y el segundo al de r .

El **output** utilizará el formato "%d\n" que corresponderá al número de combinaciones posibles. Si las entradas no cumplen las precondiciones deberá imprimir "ERROR" y continuar leyendo. Si se recibe un valor n negativo se deberá terminar el programa.

Input:

Output:

6 4	15
7 3	35
-1 0	

NOTA: Para cada una de las funciones implementados se deberá incluir una pequeña descripción de su funcionamiento, sus precondiciones mediante `assert_domjudge` si las hubiera y el análisis de su complejidad temporal y espacial. Esta información deberá incluirse en la cabecera de cada función.