	TEST	E DE	MES	A ME	RGES	ORT					
Algoritmo	ORDENAR: SE inicio < firm: INT meio = ((firm-inicio)/2)+inicio ORDENAR(VETOR, inicio, meio) ORDENAR(VETOR, meio+1, firm) COMBINAR(VETOR, inicio, meio, firm) FIM DO ORDENAR; COMBINAR: int *VEsquerda = new int[meio-inicio+1] int *VDireita = new int[firm-meio] PARA i = 0 ATÉ meio-inicio+1 vEsquerda = VETOR[inicio+1] PARA i = 0 ATÉ meio-inicio+1 vEsquerda = VETOR[meio+i+1] int ie = 0 , id = 0, iv = inicio; //variveis de controle dos vEsquerda e vDireita ENQUANTO ie < (meio-inicio+1) E id < (firm-meio): SE vEsquerda[ie] < vDireita[id]: VETOR[iv] = vEsquerda[ie] ie++ SE NÃO: VETOR[iv] = vDireita[id] id++ iv++: ENQUANTO ie < (meio-inicio+1): VETOR[iv] = vEsquerda[ie] ie++ iv++ ENQUANTO id < (firm-meio): VETOR[iv] = vVireita[id] id++ iv++ ENQUANTO id < (firm-meio): VETOR[iv] = vVireita[id]										
Instância	iv++ A = [3,	41,52,2	26,38,5	7,9,49]	, N=8						
	v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	v[6]	v[7]	ie	id	iv
vetor	3	41	52	26	38	57	9	49			
νE	3	41	52	26							
vD	38	57	9	49							
vE1	3	41									
vD1	52	26									
vE2	3										
vD2	41										
vE3	52										
vD3	26										
vE1'	38	57									
vD1'	9	49									
vE2' vD2'	38 57										
	J1										
vE3' vD3'	9 49								0	.0	0
									0	0	0

vE1* vE1*	3 3	41							1 1	0 1	1 2
VE 1									0	0	0
vD1*	26								0	1	1
vD1*	26	52							1	1	2
									0	0	0
vE*	3								1	0	1
vE*	3	26							1	1	2
vE*	3	26	41						2	1	3
vE*	3	26	41	52					2	2	4
5 .41									0	0	0
vE1'	38								1	0	1
∨E1'	38	57							1	1	2
D4!									0	0	0
vD1'	9	40							1	0	1
vD1'	9	49							1 0	1	2
vD*	9								0	0	0
vD*	9	38							1	1	2
VD	9	36							1	ı	2
vD*	9	38	49						1	2	3
vD*	9	38	49	57					2	2	4
, <u>, , , , , , , , , , , , , , , , , , </u>				<u> </u>						_	
									0	0	0
vetor	3								1	0	1
vetor	3	9							1	1	2
vetor	3	9	26						2	1	3
vetor	3	9	26	38					2	2	4
vetor	3	9	26	38	41				3	2	5
vetor	3	9	26	38	41	49			3	3	6
vetor	3	9	26	38	41	49	52		4	3	7
vetor	3	9	26	38	41	49	52	57	4	4	8
Quantidade											
de passos	24										

TESTE DE MESA COMBSORT

ORDENAR:

MAX = 7

div = MAX/1.3

PARA i ATÉ MAX and div != 0:

PARA j ATÉ MAX-div:

SE v[j] > v[j+div]:

Algoritmo aux = v[j+div]

v[j+div] = v[j]

v[j] = aux

div /= 1.3

FIM DO PARA;

FIM DO ORDENAR;

Instância A = [3,41,52,26,38,57,9,49], N=8

	• ′										
	v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	v[6]	v[7]		j	div
vetor	3	41	52	26	38	57	9	49			
vetor	3					57			0	0	5
vetor		9					41		0	1	5
vetor			49					52	0	2	5
vetor	3	9	49	26	38	57	41	52			
vetor	3			26					1	0	3
vetor		9			38				1	1	3
vetor			49			57			1	2	3
vetor				26			41		1	3	3
vetor					38			49	1	4	3
vetor	3	9	49	26	38	57	41	52			
vetor	3		49						2	0	2

vetor		9		26					2	1	2
vetor			38		49				2	2	2
vetor				26		57			2	3	2
vetor				20		31				3	
vetor					41		49		2	4	2
vetor						52		57	2	5	2
vetor	3	9	38	26	41	52	49	57			
vetor	3	9							3	0	1
vetor		9	38						3	1	1
vetor		<u> </u>	30						3	ı	'
vetor			26	38					3	2	1
vetor				38	41				3	3	1
vetor					41	52			3	4	1
						10				_	_
vetor						49	52		3	5	1
vetor							52	57	3	6	1
vetor	3	9	26	38	41	49	52	57			
Quantidade	04										
de passos	21										

TESTE DE MESA BINARIO

```
SOMA:
             aux = 0
             PARA i = 3 ATÉ n:
             soma = vetorA[i-1] + vetorB[i-1] + aux
             SE soma <= 1:
              aux = 0
              vetorC[i] = soma
             SENÃO SE soma == 2:
              aux = 1;
Algoritmo
              vetorC[i] = 0
             SENÃO SE soma == 3:
              aux = 1;
              vetorC[i] = 1
             FIM DO PARA;
            vetorC[0] = aux
           FIM DO SOMA;
```

Vetor A	0	1	1	1	aux		soma
Vetor B	0	1	0	1			
Start					0	3	0
Vetor C				0	1	3	2
Vetor C			0		1	2	2
Vetor C		1			1	1	3
Vetor C	1				1	0	Χ
Vetor C	1	1	0	0			
Passos:	4						

TESTE DE MESA INSERTIONSORT

```
ORDENA:
int aux, j
PARA i = n-1 ATÉ 0:
aux = vetor[i]
j = i+1
ENQUANTO j < n AND vetor[j] <
aux:
vetor[j-1] = vetor[j]
j++
vetor[j-1] = aux
FIM DO PARA;
FIM DO ORDENA;
```

Instância S = [30, 16, 32, 26, 28, 16], N=6

	v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	i	j	aliv	vetor[j]
vetor	30	16	32	26	28	16	5	6	16	Х
vetor	30	16	32	26	28	16				
							4	5	28	16
troca					28	16	4	6	28	16
vetor	30	16	32	26	16	28				
							3	4	26	16
troca				26	16					
vetor	30	16	32	16	26	28				
							3	5	26	28
vetor	30	16	32	16	26	28				
							2	3	32	16
troca			32	16						
vetor	30	16	16	32	26	28				
							2	4	32	26
troca				32	26					
vetor	30	16	16	26	32	28				
							2	5	32	28
					32	28				
vetor	30	16	16	26	28	32				
							1	2	16	16
vetor	30	16	16	26	28	32				
vetor	30	16	16	26	28	32	1	3	16	26
VEIO	1 30	10	10	20	20	32	'	5	10	ا کے

vetor vetor vetor	30 30 30	16 16 16	16 16 16	26 26 26	28 28 28	32 32 32	1 1 1	4 5 6	16 16 16	28 32 X
										4.0
							0	1	30	16
troca	30	16					ı			
vetor	16	30	16	26	28	32				
							0	2	30	16
troca		30	16							
vetor	16	16	30	26	28	32	1			
							0	3	30	26
troca			30	26						
vetor	16	16	26	30	28	32	,			
							0	4	30	28
troca				30	28					
vetor	16	16	26	28	30	32				
							0	5	30	32
vetor	16	16	26	28	30	32				
							0	6	30	Х
vetor	16	16	26	28	30	32				
Quantidade										
de passos	15									