

動態建 $m \times n$ 矩陣 Array

⇒ 採用 vector 自動控管 memory 不用 new/delete

$m \times n = 6 \times 5$

$i=0$	$j=0$	$V[0]$	$V[1]$	$V[2]$	$V[3]$	$V[4]$
1		[5]	[6]	[7]	[8]	[9]
2		[10]	[11]	[12]	[13]	[14]
3		[15]	[16]	[17]	[18]	[19]
4		[20]	[21]	[22]	[23]	[24]
5		[25]	[26]	[27]	[28]	[29]

陣列的索引 l index

int main () 自行作

int m, n;

cin >> m >> n;

// Array 大小 = $m \times n$, 元素: 整數

// 索引換值轉換: $l = i * n + j$
 $18 = 3 * 5 + 3 = 18$

(i, j) : 2D Array ind
 (l) : 1D Array ind

vector<int> v;

所容納元素
型態

v.assign(m*n, 1); // 產生 $m \times n$ 大小的陣列, 所有值 = 1

for (int i=1; i<m; i++) {

for (int j=1; j<n; j++) {

// $v(i, j) =$ 上方 $v(i-1, j)$ + 左邊 $v(i, j-1)$ = 維索引

// v: 一維陣列

// $v(i*n+j) = v((i-1)*n+j) + v(i*n+j-1)$ - 維索引

$v[i*n+j] = v[(i-1)*n+j] + v[i*n+j-1];$

cout << v[i*n+j] << " ";

}
 cout << endl;

// 印出陣列

