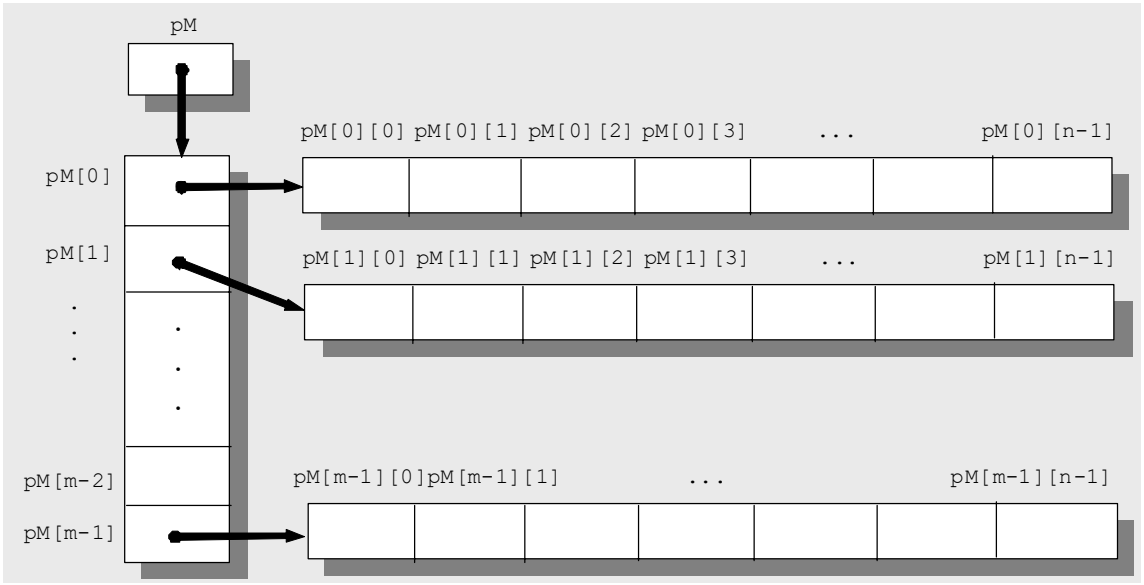
操作結果

```
執行 Swap() 前,
A 是 :
   0     1     2     3     4
B 是 :
   0     3     6     9    12    15    18
執行 Swap() 後,
A 是 :
   0     3     6     9    12    15    18
B 是 :
    0     1     2     3     4
```

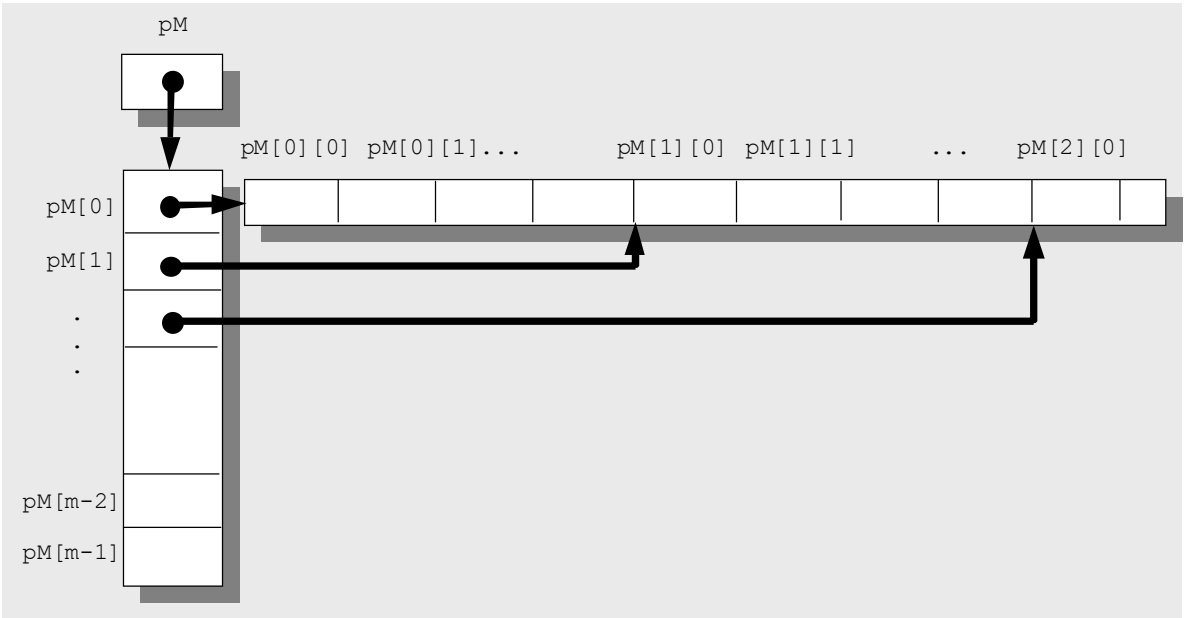

以逐列的方式建構二維陣列

```
int m = 20, n = 50;
// 動態配置長度為 m 的一維指標陣列
double **pM = new double *[m];
for (int i=0; i<m; i++)
pM[i] = new double[n];
// 回收二維陣列的記憶體
for (int i=0; i<m; i++)
   delete pM[i];
delete pM;
```

二維陣列的動態記憶體配置 (以逐列的方式建構二維陣列)



以連續記憶空間的方式建構二維陣列



以連續記憶空間的方式建構二維陣列

```
int m = 20, n = 50;
// 動態配置長度為 m 的一維指標陣列
double **pM = new double *[m];
pM[0] = new double [m*n];// pM[0]指向新向量的開頭處
for (int i=1; i<m; i++)
      pM[i] = pM[i-1]+n;  // 每個pM[i]指向陣列的特殊
delete pM[0];  // 釋放指標pM[0]所指的向量
delete pM;// 釋放pM所指的指標向量
```


範例程式 DynMatrix.cpp
向量和矩陣用在檢查亂數的平均值

```
// DynMatrix.cpp
#include <iomanip>
#include <iostream>
#include <new>
using namespace std;
const int m = 2;
const int n = 3;
// --- 各函數的宣告 ------------------------
void  ShowMatrix(double **);
double MatrixAvg (double **);
void  Sum(double **, double **, double **);
void  LackMemory()
{
  cerr << "記憶體不足!\n";
  abort();
}
```

```
// --- 主程式 ----------------------------
int main()
{
  // 動態記憶體配置 pMa
  set_new_handler(LackMemory);
  double **pMa = new double *[m];
  for (int i=0; i<m; i++)
      pMa[i] = new double[n];

  for (int i=0; i< m; i++)
      for (int j=0; j< n; j++)
         pMa[i][j]= (i*i+2.0*j)/2.0;
  // 動態記憶體配置 pMb
  double **pMb = new double *[m];
  pMb[0] = new double [m*n];
  for (int i=1; i<m; i++)
      pMb[i] = pMb[i-1]+n;
  for (int i=0; i< m; i++)
    for (int j=0; j< n; j++)
      pMb[i][j] = double(i+j)/2.0;
```



```
// 動態記憶體配置 pMc
  double **pMc = new double *[m];
  pMc[0] = new double [m*n];
  for (int i=1; i<m; i++)
      pMc[i] = pMc[i-1]+n;
// 顯示 pMa 和 pMb
  cout  << "陣列 pMa 是: " << endl;
  ShowMatrix(pMa);
  cout  << "陣列 pMb 是: " << endl;
  ShowMatrix(pMb);
  // 求 pMc
  Sum(pMa, pMb, pMc);
  cout  << "陣列 pMa + pMb 是: "   << endl;
  ShowMatrix(pMc);
  // 求 pMa 的平均值
  cout  << "陣列 pMa 的平均值是: "  << MatrixAvg(pMa) <<
endl;
// 回收 pMa
 for (int i=0; i<m; i++)
  delete [] pMa[i];
 delete [] pMa;
```

```
// 回收 pMb
  delete [] pMb[0];
  delete [] pMb;
  // 回收 pMc
  delete [] pMc[0];
  delete [] pMc;
  return 0;
}
// --- 函數 ShowMatrix() 的定義 ----------
void ShowMatrix(double **M)
{
 for (int i=0; i< m; i++)
 {
    for (int j=0; j< n; j++)
       cout << setw(5) << M[i][j];
    cout << endl;
 }
 cout << endl;
 return;
}
```



```
// --- 函數 MatrixAvg() 的定義 ------------
double MatrixAvg(double **M)
{
  double Sum = 0;
  for (int i=0; i< m; i++)
    for (int j=0; j< n; j++)
         Sum+= M[i][j];
  return  Sum / double(m*n);
}
// --- 函數 Sum() 的定義 -------------------
void Sum(double **X, double **Y, double **Z)
{
  for (int i=0; i< m; i++)
    for (int j=0; j< n; j++)
        Z[i][j]= X[i][j]+Y[i][j];
  return;
}
```

程式 DynMatrix.cpp 操作結果

```
陣列pMa是：
  01  2
  0.51.52.5
陣列pMb是：
  00.51
  0.511.5
陣列pMa + pMb是：
  01.53
  12.54
陣列pMa的平均值是：1.25
```