**C++申请动态连续内存的二维数组**

#include <iostream>

int main()

{

int col, row;

std::cin >> row >> col;

int \*\*dp = new int\*[row];

\*dp = new int[row\*col];

for (int i = 0; i < row; i++)

{

dp[i] = \*dp + i \* col;//以这种方式申请的二维数组在内存上是连续的

}

for (int i = 0; i < row; i++)

{

for (int j = 0; j < col; j++)

{

dp[i][j] = i + j; //给数组赋值

}

}

for (int i = 0; i < row; i++)//测试代码

{

for(int j = 0;j < col;j++)

{

std::cout << "address " << i << "," << j << ": " << &dp[i][j] << std::endl;

}

std::cout << "------------------------------------------" << std::endl;

}

delete[] \*dp;//释放二维数组数据

delete[] dp;//释放行地址空间

while (1);

}

# 函数版本

## 声明

template<class T>

T \*\*CreateArray2D(int row, int col);

template<class T>

void DeleteArray2D(T \*\*p);

## 定义

template<class T>

T \*\*CreateArray2D(int row,int col)

{

T \*\*p = new T\*row];

\*p = new Trow\*col];

for (int i = 0; i < row; i++)

p[i] = \*p + i \* col;

for (int i = 0; i < row; i++)

for (int j = 0; j < col; j++)

p[i][j] = 0;

return p;

}

template<class T>

void DeleteArray2D(T \*\*p)

{

delete[] \* p;

delete[] p;

p = NULL;

}

## 测试用例

int main()

{

int N, M;

std::cin >> M >> N;

int \*\*dp = CreateArray2D<int>(M, N);

for (int i = 0; i < M; i++)

{

for(int j = 0;j < N;j++)

{

std::cout << "address " << i << "," << j << ": " << &dp[i][j] << std::endl;

}

std::cout << "------------------------------------------" << std::endl;

}

DeleteArray2D(dp);

while (1);

}