Дано масив А розміром 4х8. Посортувати за спаданням стовбець масиву, що містить найбільший за модулем елемент усього масиву.

|  |
| --- |
| #include <iostream>  #include <time.h>  using namespace std;  int rnd(int i, int a){srand(time(0)+rand());return rand()%(a-i+1)+i;}  void sd(int\*, int, int);  void hs(int\*, int);//Heap  void ss(int\*, int);//Shell  const int \_n = 20;  const int cols = 8;  const int rows = 4;  int M[cols][rows];  /\*  Дано масив А розміром 4х8.  Посортувати за спаданням стовбець масиву, що містить  найбільший за модулем елемент усього масиву. Сортування здійснити 2 методами: Шелла та пірамідальним сортуванням  \*/  void main(){  for(int j=0;j<rows;j++){  for(int i=0;i<cols;i++){  M[i][j] = rnd(10, 50) \* (rnd(0,5)?1:-1);  if(M[i][j] > 0)cout << " ";  cout << M[i][j] << " ";  }  cout << "\r\n";  }  cout << "\r\n\r\n";    int col = 0, ma = M[0][0];  for(int i=0;i<cols;i++){  for(int j=0;j<rows;j++){  if(M[i][j] > ma){  ma = M[i][j];  col = i;  }  }  }    cout << ma << " " << col << " ";    //shell Sort  cout << "Shell Sort: \r\n";  ss(M[col], rows);    for(int j=0;j<rows;j++){  for(int i=0;i<cols;i++){  if(M[i][j] > 0)cout << " ";  cout << M[i][j] << " ";  }  cout << "\r\n";  }  cout << "\r\n\r\n";  //heap Sort  cout << "Shell Sort: \r\n";  hs(M[col], rows);    for(int j=0;j<rows;j++){  for(int i=0;i<cols;i++){  if(M[i][j] > 0)cout << " ";  cout << M[i][j] << " ";  }  cout << "\r\n";  }  system("pause>>NUL");  }  ///////////////////////////////////////  void sd(int \*arr, int start, int end){  int root = start;  int l = 2\*root + 1;  int r = 2\*root + 2;  int m;  if(l > end){  return;  }else if(l == end){  m = l;  }else if(arr[l] > arr[r]){  m = l;  }else{  m = r;  }  if(arr[m] > arr[root]){  swap(arr[m], arr[root]);  sd(arr, m, end);  }  }  void hs(int \*arr, int n){  for (int i=0; i<n-2; i++){  swap(arr[0], arr[n-1-i]);  sd(arr, 0, n-i-1);  }  }  void ss(int\* arr, int n){  int j;  for (int gap = n / 2; gap > 0; gap /= 2){  for (int i = gap; i < n; ++i){  int temp = arr[i];  for (j = i; j >= gap && temp < arr[j - gap]; j -= gap){  arr[j] = arr[j - gap];  }  arr[j] = temp;  }  }  } |

