Задание №12. Постройте функцию, которая возвращает медиану массива X. *Медиана* – это элемент, который после упорядочивания множества оказывается в его середине. Если в множестве четное число элементов, то медиана – это полусумма двух средних элементов.

```
double mdn(int* X, int nx){
    int* tmp = new int[nx];                    //создаем массив tmp и
    for(int i=0; i<nx; i++) tmp[i]=X[i];//копируем в него массив X
    mysrt(tmp, nx); //сортируем tmp и помещаем в res его медиану
    double res = nx%2 ? tmp[nx/2] : (tmp[nx/2-1]+tmp[nx/2])/2.;
    delete[]tmp; return res;   //удаляем tmp и возвращаем медиану
}
```

Отсортируйте строки статической матрицы по убыванию их медианы:

```
=== Задание 12 ================
Исходная матрица и медианы ее строк:
  19  51  23  64 -44 -39 -98 -11 -94  -8 |  -9.5
  20  42 -22 -94 -46  10 -48 -93  36  23 |  -6.0
  21   7  -2  26  37 -34  88 -54  67  57 |  23.5
  32 -29  76  43 -12  57 -48  54 -60 -13 |  10.0
 -37 -83 -58  81  33  97 -23  50  27 -23 |   2.0
Итоговая матрица и медианы ее строк:
  21   7  -2  26  37 -34  88 -54  67  57 |  23.5
  32 -29  76  43 -12  57 -48  54 -60 -13 |  10.0
 -37 -83 -58  81  33  97 -23  50  27 -23 |   2.0
  20  42 -22 -94 -46  10 -48 -93  36  23 |  -6.0
  19  51  23  64 -44 -39 -98 -11 -94  -8 |  -9.5
```

```
//объявляем и инициализируем статическую матрицу M
const int nr=5, nc=10; int M[nr][nc]; myrnd(M[0],nr*nc,99,99);
//объявляем массив X и инициализируем его медианами строк M
double X[nr]; for(int i=0; i<nr; i++) X[i] = mdn(M[i],nc);

cout<<"Исходная матрица и медианы ее строк:\n"<<setprecision(1);
for(int i=0; i<nr; i++) {
    myout(M[i],nc,nc+5,4); cout<<" "<<X[i]<<endl;
}
//помещаем в массив указателей адреса строк
int* pM[nr]; for(int i=0; i<nr; i++) pM[i] = M[i];

//сортируем строки матрицы M (тип pM должен соответствовать типу X)
mysrt(X,nr,4,(double**)pM);

cout<<"Итоговая матрица и медианы ее строк:\n";
for(int i=0; i<nr; i++) {
    myout(pM[i],nc,nc+5,4);  cout<<" "<<X[i]<<endl;
}
```

Задание №13. Поместите элементы исходной матрицы, расположенные над побочной диагональю, в первую полудинамическую матрицу, а элементы, расположенные под побочной диагональю – во вторую:

```
=== Задание 13 ============
Исходная матрица:
   9  -8  -3  -3  -5   4
  -2   3  -6   7  -3  -1
  -5   6   3   9   5   7
   3  -8   5  -7   4   4
   2   6  -8  -6   8  -4
   1  -3   7  -4  -7   0
Элементы над побочной диагональю:
   9  -8  -3  -3  -5
  -2   3  -6   7
  -5   6   3
   3  -8
   2
Элементы под побочной диагональю:
                  -1
               5   7
          -7   4   4
      -8  -6   8  -4
  -3   7  -4  -7   0
```

```
//объявляем, инициализируем и выводим статическую матрицу M
 n  M  myrnd(M
cout<<  myout(M
//объявляем массивы указателей и соэдаем массивы размеров
* A  * B
* sizeA = new  * sizeB = new
//определяем размеры строк матриц A, B и выделяем им память
for(int i  ) {
    sizeA  =  A  = new
    sizeB  =  B  = new
}
for(int i  ) {   //цикл по строкам
    int ia = 0, ib = 0;    //индексы строк A[i], B[i-1]
    for(int j  ) //цикл по элементам строки
        //определяем положение текущего элемента и помещаем его
        if     (  ) A[ ]  [ ++] = M  //в A[i]
        else if(  ) B[ ][ ++] = M  //или в B[i-1]
}
cout<<
for(int i  ) myout(A
cout<<
for(int i  ) {
    cout<<setw(  *4)<<  ; myout(B    }
//удаляем строки полудинамических матриц, удаляем массивы размеров
for(int i  ) {  }
```