

تابعی بنویسید که در ورودی یک آرایه دوبعدی و اندازه آن را بگیرد و به عنوان خروجی، مقدار نقطه زینی آرایه را برگرداند. نقطه زینی یک ماتریس، درایه‌ای از ماتریس است که در سطر مربوط به خود ماکزیمم و در ستون مربوط به خود می‌نیمم باشد. مثلاً عدد 6 در ماتریس زیر نقطه زینی است.

$$a = \begin{bmatrix} 4 & 1 & 9 & 3 \\ 3 & 2 & 6 & 5 \\ 7 & 0 & 8 & 2 \end{bmatrix}$$

فرض کنید دقیقاً یک نقطه زینی وجود دارد. تابع را به صورت زیر تعریف کنید که n تعداد سطرهای ماتریس است و ۲۰ تعداد ستون‌های آن است.

`float zini (float a[] [20] , int n)`

یک تابع بازگشتی بنویسید که یک آرایه و اندازه را به عنوان ورودی بگیرد و در هر مرحله، بزرگترین عضو آرایه را به انتها منتقل کند. سپس این کار را برای اعضای باقیمانده تکرار کند تا در نهایت آرایه مرتب شود. مراحل کار برای آرایه {4,7,1,3} به صورت زیر است:

$\{4, \underline{7}, 1, \underline{3}\} \rightarrow \{\underline{4}, 3, \underline{1}, 7\} \rightarrow \{1, \underline{3}, 4, 7\} \rightarrow \{1, 3, 4, 7\}$

تابع را به صورت زیر تعریف کنید:

`void sort (float a [] , int n)`

```
float zini (float a[][20] , int n)
```

```
{
```

```
    int i,j,k;
```

```
    int mx;
```

```
    bool flag;
```

```
    for ( i=0; i<n; i++)
```

```
    {
```

```
        mx = 0;
```

```
        for (j=1;j<20;j++)
```

```
            if (a[i][j] > a[i][mx])
```

```
                mx = j;
```

```
        flag = true;
```

```
        for (k=0;k<n;k++)
```

```
            if (a[k][mx] < a[i][mx])
```

```
                flag = false;
```

```
        if (flag)
```

```
            return a[i][mx];
```

```
    }
```

```
}
```

یافتن ماکزیمم ۷ نمره - بررسی می نیمم بودن ۷ نمره - خروجی ۳ نمره - سایر ۳ نمره

```
void sort(float a[], int n)
{
    int i;
    float t;
    int mx = 0;

    if (n == 1) return;

    for (i=1;i<n;i++)
        if (a[i] > a[mx])
            mx = i;

    t = a[mx];
    a[mx] = a[n-1];
    a[n-1] = t;

    sort(a,n-1);
}
```

حالت بدیهی (شرط اولیه) ۵ نمره - یافتن ماکزیمم ۵ نمره - جابجایی ۵ نمره - فراخوانی بازگشتی ۵ نمره