

## سری مسائل ۲

### کارگاه برنامه‌نویسی مقدماتی C++

#### انجمن علمی رباتیک دانشگاه شهید بهشتی

۱- برنامه‌ای بنویسید که یک ماتریس  $7 \times 7$  را با اعداد تصادفی در بازه 20- تا 20 پر کند و جدول را چاپ کند. سپس از روی ماتریس ایجادشده یک ماتریس جدید بسازد که هر عنصر از ماتریس جدید برابر با چهار عنصر مجاور در ماتریس قبلی است (منظور مجاور دایروی است). در نهایت ماتریس جدید را به عنوان خروجی نمایش دهد.

توضیح: به عنوان مثال اگر ماتریس ورودی به صورت زیر باشد

$$\begin{bmatrix} 1 & -2 & 3 & 0 & 0 & 5 & 10 \\ 9 & -7 & 3 & 2 & 1 & -1 & -6 \\ 9 & 15 & -14 & 13 & 5 & 0 & -7 \\ -6 & 3 & 2 & 8 & -3 & -6 & 6 \\ 0 & 1 & 5 & 7 & -9 & -8 & 0 \\ -3 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 9 & -6 & -18 & 1 & 5 & -8 \end{bmatrix}$$

برای اینکه سطر سوم و ستون سوم ماتریس جدید را به دست آوریم باید در ماتریس فوق درایه‌های مجاور آن را جمع

کنیم:  $3 + 13 + 15 + 2 = 33$

$$\begin{bmatrix} 1 & -2 & 3 & 0 & 0 & 5 & 10 \\ 9 & -7 & 3 & 2 & 1 & -1 & -6 \\ 9 & 15 & -14 & 13 & 5 & 0 & -7 \\ -6 & 3 & 2 & 8 & -3 & -6 & 6 \\ 0 & 1 & 5 & 7 & -9 & -8 & 0 \\ -3 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 9 & -6 & -18 & 1 & 5 & -8 \end{bmatrix}$$

برای اینکه سطر اول و ستون سوم ماتریس جدید را به دست آوریم باید در ماتریس فوق درایه‌های مجاور آن را جمع

کنیم، چون در سطر اول هستیم و همسایه بالایی برای آن وجود ندارد باید به جای همسایه بالایی عدد پایین همان ستون را در نظر بگیریم (مانند این است که سر و ته هر ستون به هم وصل هستند):  $-2 + 3 + 0 + (-6) = -5$

$$\begin{bmatrix} 1 & -2 & 3 & 0 & 0 & 5 & 10 \\ 9 & -7 & 3 & 2 & 1 & -1 & -6 \\ 9 & 15 & -14 & 13 & 5 & 0 & -7 \\ -6 & 3 & 2 & 8 & -3 & -6 & 6 \\ 0 & 1 & 5 & 7 & -9 & -8 & 0 \\ -3 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 9 & -6 & -18 & 1 & 5 & -8 \end{bmatrix}$$

برای اینکه سطر چهارم و ستون هفتم ماتریس جدید را به دست آوریم باید در ماتریس فوق درایه‌های مجاور آن را جمع

کنیم، چون در ستون هفتم هستیم و همسایه سمت راست برای آن وجود ندارد باید به جای همسایه سمت راست عدد

سمت چپ همان سطر را در نظر بگیریم (مانند این است که سر و ته هر سطر به هم وصل هستند):

$$-6 + 0 - 6 - 7 = -19$$

$$\begin{bmatrix} 1 & -2 & 3 & 0 & 0 & 5 & 10 \\ 9 & -7 & 3 & 2 & 1 & -1 & -6 \\ 9 & 15 & -14 & 13 & 5 & 0 & -7 \\ -6 & 3 & 2 & 8 & -3 & -6 & 6 \\ 0 & 1 & 5 & 7 & -9 & -8 & 0 \\ -3 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 9 & -6 & -18 & 1 & 5 & -8 \end{bmatrix}$$

برای هر کدام از اعداد روی سطرها و ستون‌ها که گوشه نباشند وضعیت مشابه دو مثال فوق را داریم.

برای اینکه سطر اول و ستون اول ماتریس جدید را به دست آوریم باید در ماتریس فوق درایه‌های مجاور آن را جمع کنیم، چون در سطر اول هستیم و همسایه بالایی برای آن وجود ندارد باید به جای همسایه بالایی عدد پایین همان ستون را در نظر بگیریم و همچنین چون در ستون اول هستیم و همسایه سمت چپ برای آن وجود ندارد باید به جای همسایه سمت چپ عدد سمت راست همان سطر را در نظر بگیریم:  $9 + (-2) + 8 + 10 = 25$

$$\begin{bmatrix} 1 & -2 & 3 & 0 & 0 & 5 & 10 \\ 9 & -7 & 3 & 2 & 1 & -1 & -6 \\ 9 & 15 & -14 & 13 & 5 & 0 & -7 \\ -6 & 3 & 2 & 8 & -3 & -6 & 6 \\ 0 & 1 & 5 & 7 & -9 & -8 & 0 \\ -3 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 9 & -6 & -18 & 1 & 5 & -8 \end{bmatrix}$$

در نهایت در ماتریس زیر برخی از درایه‌های ماتریس خروجی مشخص شده‌اند.

$$\begin{bmatrix} 25 & ? & -5 & ? & 7 & ? & -8 \\ ? & ? & ? & ? & ? & ? & ? \\ 11 & ? & ? & 1 & ? & ? & 9 \\ ? & ? & 2 & 19 & -2 & ? & ? \\ -8 & ? & ? & 5 & ? & ? & 2 \\ ? & ? & ? & ? & ? & ? & ? \\ -1 & ? & -6 & ? & -11 & ? & 27 \end{bmatrix}$$

۲- برنامه‌ای بنویسید که یک متن از کاربر دریافت کند، هر جا در متن دریافت‌شده کلمه "big" وجود داشت آن را حذف کند و به جای آن کلمه "enormous" را قرار دهد.

توضیح: مثلاً اگر ورودی به صورت زیر باشد

The big cat lounged under the big tree, staring at the big sky, enjoying a peaceful afternoon.

خروجی باید به صورت زیر شود

The enormous cat lounged under the enormous tree, staring at the enormous sky, enjoying a peaceful afternoon.

۳- تابعی بنویسید که یک آرایه به عنوان ورودی دریافت می‌کند و چیزی برنمیگرداند. با هر بار فراخوانی این تابع، آرایه‌ای که به عنوان ورودی به آن داده شده است را از کوچک به بزرگ مرتب می‌کند. (اگر این کار به کمک Dynamic Memory انجام شود یعنی بعد از اینکه آرایه مرتب‌شده چاپ شد حافظه اختصاص داده شده به آن پاک شود امتیاز اضافه خواهد داشت).

۴- برنامه‌ای بنویسید که یک آرایه به صورت پویا تعریف کند که شامل ۵ عنصر است. سپس آن ۵ عنصر را از کاربر دریافت کند. بزرگترین عدد را در یک متغیر ذخیره کند و سپس حافظه اشغال‌شده را پاک کند.

۵- تابعی بنویسید که دو vector از اعداد صحیح را به عنوان ورودی دریافت کرده و vector دوم را در ادامه vector اول قرار دهد. (یا اصطلاحاً دو vector را concatenate کند)

۶- تابعی بنویسید که دو ماتریس به عنوان ورودی دریافت کند و «ضرب کرونگر» آنها را محاسبه کند و نمایش دهد.  
توضیح: تعریف ضرب کرونگر (Kronecker product) برای ماتریس‌ها:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}.$$

مثال :

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \times 0 & 1 \times 5 & 2 \times 0 & 2 \times 5 \\ 1 \times 6 & 1 \times 7 & 2 \times 6 & 2 \times 7 \\ 3 \times 0 & 3 \times 5 & 4 \times 0 & 4 \times 5 \\ 3 \times 6 & 3 \times 7 & 4 \times 6 & 4 \times 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -4 & 7 \\ -2 & 3 & 3 \end{bmatrix} \otimes \begin{bmatrix} 8 & -9 & -6 & 5 \\ 1 & -3 & -4 & 7 \\ 2 & 8 & -8 & -3 \\ 1 & 2 & -5 & -1 \end{bmatrix} = \begin{bmatrix} 8 & -9 & -6 & 5 & -32 & 36 & 24 & -20 & 56 & -63 & -42 & 35 \\ 1 & -3 & -4 & 7 & -4 & 12 & 16 & -28 & 7 & -21 & -28 & 49 \\ 2 & 8 & -8 & -3 & -8 & -32 & 32 & 12 & 14 & 56 & -56 & -21 \\ 1 & 2 & -5 & -1 & -4 & -8 & 20 & 4 & 7 & 14 & -35 & -7 \\ -16 & 18 & 12 & -10 & 24 & -27 & -18 & 15 & 24 & -27 & -18 & 15 \\ -2 & 6 & 8 & -14 & 3 & -9 & -12 & 21 & 3 & -9 & -12 & 21 \\ -4 & -16 & 16 & 6 & 6 & 24 & -24 & -9 & 6 & 24 & -24 & -9 \\ -2 & -4 & 10 & 2 & 3 & 6 & -15 & -3 & 3 & 6 & -15 & -3 \end{bmatrix}$$