

۱- در هر یک از شبه کدهای زیر، دستور $x++$ چند بار تکرار می شود؟

<pre>for (i=1; i<=n; i++) { j=n; while(j>=1) { x++; i=i/2; } }</pre> <p>(الف) (بارم ۱)</p>	<pre>i=1; While(i<=n) { for(j=1; j<=i; j++) { x++; i++; } }</pre> <p>(ب) (بارم ۱)</p>	<pre>for(i=1; i<=n; i++) { for(j=1; j<=n; j++) { x++; j=1; while(j<n) { x++; j=j*2; } } }</pre> <p>(ج) (بارم ۱)</p>
--	---	--

۲- (الف) معادل Postfix عبارت محاسباتی روبرو را بنویسید. (بارم ۱)

$$>> A*(B+C/(D-E)+F)/G$$
وضعیت پشته را در هنگام تبدیل عبارت Infix فوق به Postfix در هر یک از موقعیت های زیر نشان دهید.

(ب) قبل از پردازش سمبل C. (بارم ۱)	(ج) قبل از پردازش سمبل E. (بارم ۱)
(د) قبل از پردازش سمبل G. (بارم ۱)	

۳- معادل Infix عبارت Prefix روبرو چیست؟ (بارم ۱)
Prefix: - + - / A B C * D E * A C

۴- یک لیست خطی یکطرفه با دو اشاره گر F و R که به ترتیب به عنصر اول و آخر لیست اشاره می کنند پیاده سازی شده است. هزینه هر کدام از اعمال زیر چیست؟

(الف) حذف اولین عنصر. (بارم ۰/۵)	(ب) حذف آخرین عنصر. (بارم ۰/۵)
(ج) درج یک عنصر در انتهای لیست. (بارم ۰/۵)	(د) درج یک عنصر در ابتدای لیست. (بارم ۰/۵)

۵- اگر آدرس شروع آرایه A در حافظه ۱۰۰ باشد و ۴ بایت برای هر متغیر int نیاز باشد، آدرس شروع درایه $A[15][9][3]$ در هر یک از حالت زیر چیست؟
 $\text{int } A[20][10][5]$
(الف) آرایه به صورت Row major ذخیره شده باشد. (بارم ۱)
(ب) آرایه به صورت Col. major ذخیره شده باشد. (بارم ۱)

۶- با توجه به لیست داده شده در زیر، خروجی هر یک از شبه کدهای زیر را مشخص کنید.

```
first → [1] → [2] → [3] → [4] → [5] → Λ
```

<pre>void what(node *p) { if (p && p->link) { cout<<p->data; p=p->link; what(p->link); } } what(first);</pre> <p>(الف) (بارم ۱)</p>	<pre>node *what(node *p) { if (p && p->link) return what(p->link); else return p; } node *q=what(first); cout<<q->data;</pre> <p>(ب) (بارم ۱)</p>	<pre>node *what(node *p) { if (p && p->link) return what(p->link->link); else return p; } node *q=what(first); cout<<q->data;</pre> <p>(ج) (بارم ۱)</p>	<pre>void what(node *p) { if (p) { what(p->link); cout<<p->data; what(p->link); } } what(first);</pre> <p>(د) (بارم ۱)</p>
--	---	--	--

الگوریتم Insertion Sort

```
for(j=2;j<=n;j++)
{
  r=List[j];
  i=j-1;
  While(i>0 && r<List[i])
  {
    List[i+1]=List[i];
    i--;
  }
  List[i+1]=r;
}
```

۷- شبه کد الگوریتم Insertion Sort در روبرو آمده است. این شبه کد را به گونه ای تغییر دهید که بجای یک آرایه، عناصر یک لیست پیوندی دو طرفه خطی را مرتب کند. هیچ پیش فرضی را در مورد تعداد عناصر لیست در نظر نگیرید و تمام حالات خاص را پوشش دهید. (بارم ۴)

List → Λ [] [] [] [] [] ... [] [] Λ