

مدت آزمون: ۱۲۰ دقیقه
تیر ۱۳۸۵

به نام خدا
آزمون پایان ترم درس ساختمان داده ها

نام:
شماره دانشجویی:

۱- اگر آدرس شروع آرایه A در حافظه ۱۰۰ باشد و ۴ بایت برای هر متغیر int نیاز باشد، آدرس شروع درایه A[5][10][15] در هر یک از حالت زیر چیست؟
int A[10][20][30]

(ب) آرایه به صورت Col. major ذخیره شده باشد. (۰/۵)

پاسخ

(الف) آرایه به صورت Row major ذخیره شده باشد. (بارم ۰/۵)

پاسخ

۲- در هر یک از موارد زیر پیمایش خواسته شده را بدست آورید.

Postorder: 5, 6, 15, 10, 23, 24, 22, 26, 20

(الف) پیمایش Preorder یک درخت جستجوی دودویی BST با پیمایش Postorder به صورت زیر: (بارم ۱)

Preorder

Preorder: A,D,F,G,H,K,L,P,Q,R,W,Z
Inorder: G,F,H,K,D,L,A,W,R,Q,P,Z

(ب) پیمایش Postorder یک درخت دودویی با پیمایشهای Inorder و Preorder به صورت روبرو: (بارم ۱)

Postorder

Inorder: 5, 6, 15, 10, 23, 24, 22, 26, 20

(ج) پیمایش Levelorder یک درخت دودویی کامل (Complete) با پیمایش Inorder به صورت روبرو: (بارم ۱)

Levelorder

1	2	3	4	5	6	7	8	9

۳- الف) به یک Min-heap خالی گره هایی با کلیدهای (به ترتیب از راست به چپ) <<2,50,45,70,75,22,40,55,45 اضافه شده است. Min-heap حاصل را رسم کنید. (بارم ۰/۵)

1	2	3	4	5	6	7	8	9

(ب) پس از انجام سه عمل حذف (Delete) بر روی Min-heap بند الف، Min-heap حاصل را رسم کنید. (بارم ۰/۵)

۴- لیست روبرو به روش QuickSort مرتب خواهد شد. (لیست اولیه = (28, 22, 85, 75, 61, 73, 54, 13, 60, 18, 53, 37, 81))

محتوی لیست را پس از قرار گرفتن 28 در محل نهایی خود نشان دهید. (بارم ۰/۵)

۵- لیست روبرو به روش HeapSort مرتب خواهد شد. (لیست اولیه = (28, 22, 85, 75, 61, 73, 54, 13, 60, 18, 53, 37, 81))

لیست اولیه را مشخص کنید. توجه کنید که Heap اولیه از پایین به بالا ساخته می شود. (بارم ۰/۵)

Prefix: / * A - B / * C - + D E F G - H I

۶- معادل Infix عبارت Prefix روبرو چیست؟ (بارم ۱)

Infix:

۷- با توجه به کد Radix sort که در زیر آمده است، به سوالات زیر پاسخ دهید:

```
void Radixsort(Element* list, const int d, const int n) {
// Records list = (list[1], ... , list[n]) are sorted on the keys key[0], ... , key[d-1].
// The range of each key is 0 <= key[i] < radix. radix is a constant.
// Sorting within a key is done using a bin sort.
    int e[radix], f[radix];
    for (int i = 1; i <= n; i++) list[i].link = i + 1;
    list[n].link = 0; int current = 1;
    for (i = d-1; i >= 0; i--) // Main Loop
    {
        for (int j=0; j < radix; j++) f[j] = 0;
        for (; current; current=list[current].link) {
            int k = list[current].key[i];
            if (f[k] == 0) f[k] = current;
            else ??????????????;
            e[k] = current;
        }
        for (j=0; f[j] == 0; j++);
        current = f[j]; int last = e[j];
        for (int k = j+1; k < radix; k++)
            if (f[k]) {
                list[last].link = f[k]; *****;
            }
        list[last].link = 0;
    } // end of for (i=d-1; i >= 0; i--)
// end of function
}
```

الف) به جای ?????????? چه عبارتی باید قرار گیرد؟ (بارم ۱)

پاسخ

ب) به جای ***** چه عبارتی باید قرار گیرد؟ (بارم ۱)

پاسخ

فرض کنید که لیست زیر ورودی الگوریتم باشد.

	1	2	3	4	5	6	7	8	9	10
Key	33	172	9	55	489	958	39	603	802	971
Link										

ج) وضعیت لیست پس خاتمه اولین تکرار حلقه Main Loop چیست؟ (بارم ۱/۵)

	1	2	3	4	5	6	7	8	9	10
Key										
Link										

د) به جای توضیح //end of function کدی بنویسید که در انتهای تابع کلیدهای لیست را به صورت مرتب شده صعودی چاپ کند. (بارم ۱/۵)

مدت آزمون: ۱۲۰ دقیقه
تیر ۱۳۸۵

به نام خدا
آزمون پایان ترم درس ساختمان داده ها

نام:
شماره دانشجویی:

۸- محل های نقطه چین را پر کنید. (بارم ۱)

الف) یک هیپ دودویی با n عنصر دقیقاً گره غیر برگ دارد.

ب) عمق یک گره در درایه i در یک هیپ دودویی با n گره دقیقاً است.

ج) حداقل تعداد عناصر یک هیپ دودویی با ارتفاع h دقیقاً برابر است.

د) سومین کوچکترین کلید در یک Min Heap با کلیدهای متمایز در درایه های با اندیسهای می تواند باشد.

۹- تابعی زیر چک می کند که آیا یک درخت دودویی با عناصر صحیح (int) و متمایز و با ریشه root یک درخت جستجوی دودویی است یا خیر. محل های خالی A تا D در تابع زیر را کامل کنید. (بارم ۲)

bool IsBST(tree* t)

```
{
    Return IsBST(t, MININT, MAXINT);
}
```

int IsBST(tree *t, int min, int max)

```
{
    If (t==NULL) return TRUE;
    If (t->data<min || t->data>max) return FALSE;
    Return IsBST(t->LeftChild, ...A..., ...B...) && IsBST(t->RightChild, ...C..., ...D...);
}
```

A	
B	
C	
D	

* ۱- تابعی بنویسید که عنصر با بزرگترین کلید کوچکتر از کلید ریشه را در صورت وجود از یک درخت جستجوی دودویی (BST) حذف کند. (بارم ۲/۰)

مدت آزمون: ۱۲۰ دقیقه
تیر ۱۳۸۵

به نام خدا
آزمون پایان ترم درس ساختمان داده ها

نام:
شماره دانشجویی:

۱۱- یک درخت دودویی کامل به کمک آرایه و به صورت زیر پیاده سازی شده است.

Class CompBinTree

```
{
private:
    int n; // تعداد عناصر موجود در درخت
    int tree[MaxSize+1]; // استفاده نمی شود.
public:
    CompBinTree() {n=0;}
    void PrintInoeder(); // محتوی گره های درخت را با پیمایش Inorder چاپ میکند.
    int NoLeaf(); // تعداد گره های برگ درخت را برمیگرداند.
    .....;
};
```

الف) تابع PrintInorder را که محتوی گره های درخت مذکور را با پیمایش Inorder چاپ میکند، بنویسید. (بارم ۱/۰)

ب) تابع NoLeaf را که تعداد گره های برگ درخت مذکور را برمیگرداند، بنویسید. (بارم ۱/۰)

مدت آزمون: ۱۲۰ دقیقه
تیر ۱۳۸۵

به نام خدا
آزمون پایان ترم درس ساختمان داده ها

نام:
شماره دانشجویی:

۱۲- بر اساس تعریف کلاس لیست تعمیم یافته روبرو به سوال زیر پاسخ دهید.
تابع `GetListAsStr(char *)` را کامل کنید. این تابع یک رشته از ورودی می گیرد و از روی آن گره های لیست را می سازد. (بارم ۲/۰)
مثال رشته ورودی: `(ab(cd)(e(fg)(hi)))`

```

class GenListNode
{
    friend class GenList;
    Private:
        GenListNode *link;
        int tag; // 0 for data, 1 for dlink, 2 for ref
        union
        {
            int data;
            GenListNode *dlink;
            int ref;
        };
};

Class GenList
{
    Private:
        GenListNode *first;
    Public:
        GenList();
        ~GenList();
        void GetListAsStr(char *);
};

```