



مبانی برنامه‌سازی (C) ۴۰۱۵۳
Fundamental of Programming (C)

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

تاریخ آزمون: شنبه ۲۱ دی ۱۳۹۲

وقت: ۱۸۰ دقیقه

آزمون پایان ترم

به نکات زیر توجه داشته باشید.

نکته ۱: جواب سوالات تستی را به ترتیب با ذکر شماره سوال و گزینه درست در پاسخ‌نامه بنویسید. به ازاء هر سوال تستی صرفاً یک جواب درست در بین گزینه‌ها وجود دارد.

نکته ۳: سوال ۴ (سوال تشریحی آخر) مخصوص دانشجویان مهندسی کامپیوتر و برق بوده و پاسخ دادن به این سوال برای این دانشجویان **الزامی** می‌باشد. دانشجویان رشته‌های دیگر در صورت پاسخ به این سوال، نمره اضافی دریافت **نخواهند کرد**.

نکته ۴: کدهای خود را تمیز و خوانا نوشته و کامنت‌گذاری نمائید.

سوالات تستی (۵۰ نمره)

۱. عبارت مقابل چه چیزی را عیان می‌کند؟

```
int *countPtr, count;
```

الف. دو متغیر `int`

ب. یک اشاره‌گر به `int` و یک متغیر `int`

ج. دو اشاره‌گر به `int`

د. این نوع تعریف اشتباه است.

۲. یک سازنده در تعریف کلاس به چه تعداد پارامتر ورودی می‌تواند داشته باشد؟

الف. 0

ب. 1

ج. 2

د. به تعداد متغیر

۳. کدامیک از گزینه‌های زیر در رابطه با تابع func صحیح است؟

```
long func(int num) {
    int i;
    long f=1;
    for (i=1; i<=num; i++)
        f=f*i;
    return f;
}
```

- الف. تابع یک عدد به توان num را محاسبه می‌کند.
ب. تابع جذر یک عدد صحیح را محاسبه می‌کند.
ج. تابع فاکتوریل یک عدد صحیح را محاسبه می‌کند.
د. هیچکدام

۴. خروجی برنامه زیر کدام گزینه است؟

```
int main()
{
    const int x=5;
    const int *ptrx;
    ptrx = &x;
    *ptrx = 10;
    printf("%d\n", x);
    return 0;
}
```

- الف. 5
ب. 10
ج. Error
د. Garbage value

۵. خروجی برنامه زیر کدام گزینه است؟

```
int main()
{
    int a=1;
    const int *aPtr = &a;
    *aPtr = 2;
    printf("%d\n", a);
    return 0;
}
```

- الف. 1
ب. 2
ج. آدرس a
د. Error

۶. خروجی برنامه زیر کدام گزینه است؟

```
int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    printf("%d, %d, %d", i, j, m);
    return 0;
}
```

- الف. 2, 1, 15
ب. 3, 2, 15
ج. 3, 1, 15
د. 2, 3, 30

۷. خروجی برنامه زیر کدام گزینه است؟

```
int main()
{
    union var
    {
        int a;
        int b;
    };
    union var v;
    v.a=10;
    v.b=20;
    printf("%d\n", v.a);
    return 0;
}
```

الف. 10
ب. 20
ج. 0
د. Error

۸. خروجی برنامه مقابل کدام گزینه است؟

```
int main()
{
    int x = 0;
    int &y = x; y = 5;
    while(x <= 5)
    {
        cout<< y++ << " ";
        x++;
    }
    cout<< x;
    return 0;
}
```

الف. 5 6 7 8 9 10
ب. 5 6 7 8 9 10 7
ج. 5 7
د. Compile time error

۹. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```
int main()
{
    char str1[]="Hello";
    char str2[10];
    char *t, *s;
    s = str1;
    t = str2;
    while (*t==*s)
        *t++ = *s++;
    printf("%s\n", str2);
    return 0;
}
```

الف. Hello
ب. HelloHello
ج. ello
د. خروجی ندارد

۱۰. با فرض اینکه آدرس آرایه از محل 1000 حافظه شروع شده و سائز عدد صحیح 4 بایت باشد، خروجی کد زیر کدام گزینه خواهد بود؟

```
int main()
{
    int a[3][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    printf("%u, %u, %u\n", a[0]+1, *(a[0]+1), (*(a+0)+1));
    return 0;
}
```

الف. 1004, 2, 2
ب. 1016, 2, 2
ج. 1004, 5, 5
د. 1016, 5, 5

۱۱. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```
int main()
{
    char str1[]="Hello";
    char str2[10];
    char *t, *s;
    s = str1;
    t = str2;
    while (*t==*s)
        *t++ = *s++;
    printf("%s\n", str2);
    return 0;
}
```

الف. 0, 3
ب. 0, 12
ج. 12, 3
د. 12, 12

۱۲. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```
int main()
{
    char *p;
    p="hello";
    printf("%s\n", *&*p);
    return 0;
}
```

الف. llo
ب. ello
ج. hello
د. h

۱۳. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```
int main()
{
    int arr[5], i=0;
    while (i < 5)
        arr[i]=++i;
    for (i=0; i<5; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

الف. 1 2 3 4 5
ب. 0 1 2 3 4
ج. 2 3 4 5 6
د. خطا در برنامه

۱۴. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```

struct course {
    int courseno;
    char coursename[25];
};
int main()
{
    struct course c[] = { {102, "Java"},
                          {103, "PHP"},
                          {104, "DotNet"} };
    printf("%d ", c[1].courseno);
    printf("%s", *(c+2).coursename);
    return 0;
}

```

الف. 103 DotNet
 ب. 102 Java
 ج. 103 PHP
 د. 104 DotNet

۱۵. خروجی برنامه مقابل کدام گزینه خواهد بود؟

```

int main()
{
    int x=1, y=1;
    for(; y; printf("%d %d\n", x, y)){
        y = x++ <= 5;
    }
    printf("\n");
    return 0;
}

```

د. 2 2	ج. 2 1	ب. 2 1	الف. 2 1
3 3	3 1	3 1	3 1
4 4	4 1	4 1	4 1
5 5	5 1	5 1	5 1
		6 1	6 1
			7 0

۱۶. با فرض اینکه آدرس آرایه از محل 500 حافظه شروع شود، خروجی کد زیر کدام گزینه خواهد بود؟

```

int main()
{
    int arr[1][2][3] = {1, 2, 3, 4, 5, 6};
    int *p, *q;
    p = &arr[0][1][2];
    q = (int*) arr;
    printf("%d, %d\n", *p, *q);
    return 0;
}

```

الف. 3, 500
 ب. 6, 500
 ج. 3, 1
 د. 6, 1

۱۷. با در نظر گرفتن تابع مقابل، مشخص کنید کدام گزینه معادل این تابع می‌باشد؟

```
char* strcpy(char *s1, const char *s2) {
    char *p = s1;
    while(*s2!='\0') {
        *s1 = *s2;
        s1++;
        s2++;
    }
    *s1 = '\0';
    return p;
}
```

```
char* strcpy(char s1[], const char s2[]) {
    for(int i=0; i < strlen(s2); i++) {
        s1[i] = s2[i];
    }
    return s1;
}
```

الف.

```
char* strcpy(char *s1, const char *s2) {
    char *p = s1;
    while(*s2) *s1++ = *s2++;
    return p;
}
```

ب.

```
char* strcpy(char *s1, const char *s2) {
    char *p = s1;
    while(*s1++ = *s2++);
    return p;
}
```

ج.

```
char* strcpy(char *s1, const char *s2) {
    int i = 0, j = 0;
    while(s1[++i] = s2[++j]);
    return s1;
}
```

د.

۱۸. خروجی برنامه زیر چیست؟

```
class Point
{
public:
    Point(double xi, double yi)
    {
        x = xi;
        y = yi;
    }
    void print()
    {
        cout << "(" << x << "," << y << ")" << endl;
    }
    double x;
private:
    double y;
};

int main()
{
    Point p1(1.0, 2.0);
    p1.x = 111;
    p1.y = 222;
    p1.print();
    return 0;
}
```

الف. خطا در دسترسی به x

ب. خطا در دسترسی به y

ج. (111, 222)

د. (222, 111)

۱۹. با فرض اینکه سایز یک عدد صحیح 4 بایت باشد، خروجی کد زیر کدام گزینه خواهد بود؟

```
int main()
{
    int a[] = {1, 2};
    int b[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    for (int i = -1; i < sizeof(a); ++i)
        printf("%d", b[i+1]);
    return 0;
}
```

الف. 1234

ب. 12345678

ج. 123456789

د. چیزی چاپ نمی‌شود.

۲۰. با فرض اینکه سایز یک عدد صحیح 4 بایت باشد، خروجی کد زیر کدام گزینه خواهد بود؟

```
class Test {
    public:
        Test(int); // constructor
        ~Test(); // destructor
    private:
        int ID;
};
Test::Test(int a)
{
    ID=a;
    cout<<ID;
}
Test::~~Test()
{
    cout<<ID;
}

Test test1(1); // global object

int main()
{
    Test test2(2); // local object
    static Test test3(3); // static local object
    return 0;
}
```

الف. 123321

ب. 123231

ج. 132321

د. 132231

سوالات تشریحی (۵۰ نمره)

سوال ۱. (۱۰ نمره)

برنامه زیر را در نظر بگیرید:

```
int iterativeFunc(int m, int n)
{
    int temp;
    while(n != m % n) {
        temp = n;
        n = m % n;
        m = temp;
        if(n == 0) return m;
    }
    return m;
}
```

یک تابع بازگشتی معادل برنامه فوق بنویسید.

سوال ۲. (۲۰ نمره)

کلاس identity را به صورت زیر تعریف کنید:

اعضا داده‌ای: دو متغیر از نوع رشته که نام و فامیل را نشان می‌دهند و یک متغیر از نوع رشته که شماره تلفن را نشان می‌دهد (public نباشند).

کلاس دو تا سازنده باید داشته باشد، یک سازنده فقط نام و فامیل را می‌گیرد سازنده‌ی دیگر نام، فامیل و شماره تلفن را می‌گیرد (public باشند).

اعضاء عملکردی: تابعی که عدد صحیحی به عنوان ورودی می‌گیرد و آن را به عنوان شماره تلفن صاحب پلاک به روز رسانی می‌کند (public باشد).

توجه کنید که نام و فامیل بیشتر از ۲۵ حرف نمی‌توانند باشند و اگر بیشتر بودند، تنها ۲۵ حرف اول هر کدام برداشته شده و ذخیره می‌گردد. همچنین برای ذخیره‌ی شماره موبایل باید ابتدای آن صفر باشد و پس از آن ده رقم آمده باشد. در غیر این صورت (چه در سازنده و چه در تابع عملکردی) عضو داده‌ای مربوط به شماره موبایل مقداردهی نمی‌شود.

سوال ۳. (۲۰ نمره)

فرض کنید آرایه‌ای از اعداد صحیح نامرتب به صورت $A[n]$ داشته باشیم. در واقع $A[0], A[1], \dots, A[n-1]$ به دنباله‌ای از اعداد صحیح اشاره می‌کنند که مرتب نمی‌باشند. می‌خواهیم الگوریتمی برای مرتب‌سازی این آرایه بنویسیم. در ابتدا و بعد از هر مرحله تکرار در این الگوریتم، کل آرایه را می‌توان به دو قسمت مرتب شده $A[0], \dots, A[i-1]$ و نیز مرتب نشده $A[i], \dots, A[n-1]$ تقسیم کرد که $i \in 1, \dots, n-1$. برای جابجا کردن عنصر $A[i]$ از قسمت مرتب نشده به قسمت مرتب شده، این عنصر باید با $A[i-1], A[i-2], \dots, A[0]$ مقایسه شود و در محل مربوط به خود درج گردد. مثلاً اگر $A[j] \leq A[i]$ پیدا شد که $A[j] \leq A[i]$ ، عنصر $A[i]$ بعد از آن قرار می‌گیرد و در غیر اینصورت عنصر $A[i]$ در ابتدای آرایه درج می‌شود. پس از درج $A[i]$ در قسمت مرتب شده، از طول قسمت مرتب نشده یک واحد کاسته شده و به طول قسمت مرتب شده یک واحد افزوده می‌شود. در مرحله‌ی بعد، $A[i+1]$ به قسمت مرتب شده افزوده می‌شود و این عملیات تا جایی ادامه می‌یابد که همه‌ی عناصر به قسمت مرتب شده وارد شوند. در واقع در هنگام شروع الگوریتم، فقط عنصر $A[0]$ در قسمت مرتب شده قرار دارد و عناصر $A[1], A[2], \dots, A[n-1]$ در قسمت مرتب نشده قرار دارند و در انتهای الگوریتم، همه‌ی عناصر در قسمت مرتب شده قرار خواهند داشت.

تابعی بنویسید که الگوریتم مرتب‌سازی فوق را پیاده‌سازی نماید. فرض کنید تابعی با prototype زیر داریم که آرایه‌ی نامرتب A و سائز آرایه (n) ورودی‌های آن هستند. شما فقط کد مربوط به مرتب‌سازی را بنویسید.

```
void insertionSort(int A[], int n)
{
    /* Your code */
}
```

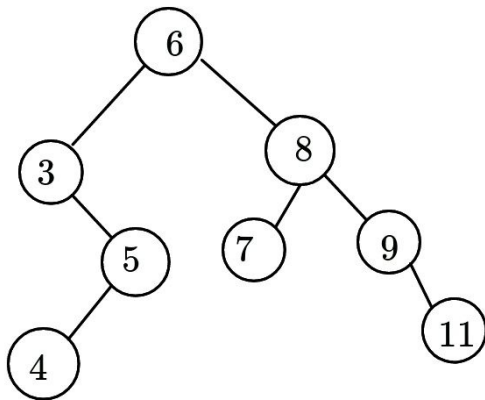
(راهنمایی: به عنوان نمونه‌ای از اجرای الگوریتم فوق، آرایه‌ی $A[8]=\{5, 7, 0, 3, 4, 2, 6, 1\}$ را در نظر بگیرید. شکل زیر، اجرای الگوریتم فوق را بر روی آرایه‌ی A و به صورت مرحله به مرحله نشان می‌دهد. اعداد موجود در قسمت مرتب شده در هر مرحله از اجرای الگوریتم، به صورت پررنگ و اعداد موجود در قسمت مرتب نشده، به صورت کم‌رنگ نشان داده شده‌اند. توجه داشته باشید که "مرحله اول" خود آرایه را نشان می‌دهد.)

5	7	0	3	4	2	6	1	(مرحله اول)
5	7	0	3	4	2	6	1	(مرحله دوم)
0	5	7	3	4	2	6	1	(مرحله سوم)
0	3	5	7	4	2	6	1	(مرحله چهارم)
0	3	4	5	7	2	6	1	(مرحله پنجم)
0	2	3	4	5	7	6	1	(مرحله ششم)
0	2	3	4	5	6	7	1	(مرحله هفتم)
0	1	2	3	4	5	6	7	(مرحله هشتم)

سوال ۴ - مخصوص دانشجویان مهندسی کامپیوتر و برق. (۲۰ نمره)

یکی از راه‌های ذخیره‌سازی تعدادی عدد، کاراکتر و به طور کلی مجموعه‌ای از عناصر مقایسه‌پذیر استفاده از آرایه‌هاست. اما جستجو برای پی‌بردن به وجود یا عدم وجود یک عنصر در آرایه به طور متوسط نیاز به دسترسی و مقایسه‌ی نیمی از اعضای آرایه دارد. یکی از راه‌های سرعت بخشیدن به جستجو استفاده از درخت دودویی جستجو (Binary Search Tree) است که آنرا بطور مخفف **دج** می‌نامیم. دج یک گراف همبند بدون دور (درخت) است که هر راس آن متناظر با یک عدد است که برچسب آن راس نامیده می‌شود و هر راس به حداکثر دو راس دیگر (به نام فرزند چپ و فرزند راست) متصل است به طوری که عدد متناظر با فرزند راست (برچسب فرزند راست) حتماً از برچسب این

راس بزرگتر و برچسب فرزند چپ حتماً از برچسب این راس کوچکتر است. مثالی از یک دج را در شکل مقابل مشاهده می‌کنید.



در این شکل 6 را ریشه می‌نامیم، 3 و 8 به ترتیب فرزندان چپ و راست 6 هستند، 6 و 8 دو فرزند دارند ولی 3، 5 و 9 فقط یک فرزند دارند و بسته به شرایط یک گره‌ی دیگر می‌تواند فرزند آن‌ها شود و 4، 7 و 11 هیچ فرزندی ندارند و سمت چپ و راست آن‌ها خالی است و می‌توانند فرزندان جدیدی بپذیرند. به عنوان مثال اگر گره‌ای با برچسب 10 بخواهد به این درخت افزوده شود، فرزند سمت چپ 11 قرار داده خواهد شد.

برای پیاده‌سازی دج ساختمان Node را به شکلی که در کد می‌بینید تعریف کرده ایم. هر Node برای ذخیره‌سازی یک راس (یا گره) به کار می‌رود و lbl نشانگر برچسب این راس و اشاره‌گرهای left و right به ترتیب اشاره‌گر به فرزندان راست و چپ هستند. همچنین تابع CreateNode یک گره‌ی جدید با برچسب label که در آرگومان‌ش می‌گیرد ایجاد می‌کند و اشاره‌گرهای فرزند راست و چپ آن‌را برابر NULL قرار می‌دهد (به این معنی که فعلاً فرزند راست و چپ برای آن تعریف نشده).

الف) با توجه به ویژگی خاص گره‌ها در دج تابعی بازگشتی بنویسید که با گرفتن یک عدد (label) و اشاره‌گر به یک Node به عنوان ریشه (root)، در درختی به ریشه‌ی root به دنبال گره‌ای با برچسب برابر label بگردد. اگر همچین گره‌ای در درخت موجود بود عدد ۱ و اگر نه عدد ۰ را برگرداند.

ب) بخش الف) را به صورت غیر بازگشتی با یک حلقه بنویسید.

پ) تابعی بنویسید که یک عدد به عنوان label بگیرد و آن‌را در مکان خالی مناسبی (فرزند سمت راست یا چپ یک گره‌ی مناسب) در درخت اضافه کند. (راهنمایی: در کد جستجوی بخش ب یا الف تغییرات مناسب را اعمال کنید)

توجه: برای پاسخگویی به این سوالات کد زیر را در پاسخنامه کامل کنید (بدنه‌ی سه تابع را بنویسید):

```

typedef struct {
    int lbl;
    Node* right;
    Node* left;
} Node;

Node* CreateNode( int label ) {
    Node* newNode;
    newNode = (Node*) malloc(sizeof(Node));
    newNode->lbl = label;
    newNode->right = nullptr;
    newNode->left = nullptr;

    return newNode;
}

int FindNodeRecursive( int label , Node* root) {

    // Your code

}

int FindNodeNonRecursive( int label , Node* root ) {

    // Your code

}

void AddNode( int label ) {

    // Your code

}

```

موفق باشید.