



بسمه تعالی

تاریخ : ۶ آذر ۱۴۰۲

آزمون میان ترم درس مبانی کامپیوتر و برنامه سازی به زبان C

مدت زمان آزمون : ۷۰ دقیقه

(میان ترم اول)

مدرس : یحیی پورسلطانی

نیمسال اول سال تحصیلی ۱۴۰۲-۱۴۰۳

همانا با یاد خدا دل ها آرام می گیرد

نام و نام خانوادگی :

شماره دانشجویی :

رشته تحصیلی :

شماره صندلی :

اینجانب با شماره دانشجویی بر اساس شماره ی صندلی اعلام شده ، در جلسه ی آزمون

حاضر شده و شماره ی صندلی خود را با مشخصات خود تطبیق دادم.

امضا

دانشجوی فرهیخته، با سلام

۱. لطفا مشخصات خودتان را با شماره صندلی اعلام شده تطبیق داده و عبارت کادر بالا را تکمیل کنید. عدم تکمیل عبارت کادر فوق به منزله غیبت

شما در آزمون و درج نمره صفر می باشد.

۲. پاسخگویی به سوالات بدون خط خوردگی ، واضح و خوانا باشد.

۳. پاسخ سوالات را منحصر در محل تعیین شده بنویسید.

۴. نام و نام خانوادگی و شماره دانشجویی خود را در بالای تمامی صفحات اوراق آزمون بنویسید.

۵. این آزمون از ۱۰۰ نمره است و پس از تصحیح، به میزان گفته شده برای آزمون میان ترم مقیاس خواهد شد.

در کادر زیر مطلقا چیزی ننویسید.

													سوال
													امتیاز

پس از تجدید نظر :

۱۰۰

نمره نهایی :

نام و نام خانوادگی

شماره دانشجویی

لطفا در جداول زیر چیزی ننویسید.

	<p>درخواست تجدید نظر</p>
	<p>نتیجه تجدید نظر</p>
<p>اینجانب دانشجوی رشته ی به شماره دانشجویی برگه ی امتحانی خود را بررسی کردم و تمامی اعتراضات خود را به سمع مدرس رسانده و اعتراض دیگری ندارم؛ بنابراین برگه ی امتحانی خود را در تاریخ پس از رسیدگی کامل به اعتراضات تحویل گرفته و اعتراض دیگری نسبت به برگه ی خود ندارم. بنابراین حق هرگونه اعتراض را پس از تحویل برگه از خود سلب کرده و می پذیرم که هیچگونه رسیدگی به اعتراضاتی که در خصوص این امتحان و پس از تحویل برگه مطرح کنم (حتی در صورتی که وارد باشد) نخواهد شد.</p> <p>امضای دانشجو</p>	<p>تحویل برگه</p>

بخش اول : سوالات مفهومی (۱۰ امتیاز)

۱. درستی و یا نادرستی هر کدام از موارد زیر را مشخص کنید؛ توجه کنید که امتیاز همه‌ی موارد یکسان است و در صورت عدم ارائه‌ی توضیحات برای موارد نادرست و یا ارائه‌ی توضیحات غلط برای آن‌ها، کل نمره برای آن مورد لحاظ نخواهد شد؛ همچنین موارد درست نیازمند توضیح نیستند و نمره‌ای به توضیحات بیشتر تعلق نمی‌گیرد (۶ امتیاز). – هر مورد ۲ امتیاز

عبارت	پاسخ	توضیح
در صورتی که <code>arr</code> یک آرایه‌ای از اعداد صحیح (<code>int</code>) باشد، دستور زیر تعداد عناصر آن را به ما می‌دهد : <code>sizeof(arr)</code>	<input type="checkbox"/> درست <input type="checkbox"/> غلط	دستور <code>sizeof</code> صرفاً میزان حافظه‌ی مصرف شده را به ما می‌دهد.
در صورت ارسال یک آرایه به یک تابع (از طریق آرگومان ورودی) می‌توان مقادیر عناصر آرایه را به صورت مستقیم دستکاری کرد.	<input type="checkbox"/> درست <input type="checkbox"/> غلط	
در صورتی که در زمان مقداردهی به یک رشته (آرایه‌ای از کاراکترها)، رشته‌ای با اندازه‌ی بزرگ‌تر از ظرفیت آرایه بدهیم، برنامه متوقف می‌شود.	<input type="checkbox"/> درست <input type="checkbox"/> غلط	رسیدن به انتهای آرایه در زمان اجرا توسط برنامه ممکن است چک نشود و لازم است توسط برنامه‌نویس ملاحظه شود.

۲. هر کدام از دو روش فراخوانی با مقدار (`Call by Value`) و فراخوانی با ارجاع (`Call by Reference`) را در یک خط توضیح دهید؛ سپس در حد یک خط تفاوت اساسی این دو روش فراخوانی را بیان کنید (۴ امتیاز).

در روش فراخوانی با مقدار، صرفاً یک کپی از مقدار پاس شده به تابع به درون تابع ارسال می‌شود (۲ امتیاز).
در روش فراخوانی با ارجاع، آدرس متغیر به تابع داده شده و استفاده و یا تغییر مقدار متغیر به صورت مستقیم و بر روی خود متغیر اعمال می‌شود (۲ امتیاز). به سایر پاسخ‌های صحیح نیز به فراخور، نمره داده شود.

بخش دوم : سوالات تحلیل کد (۶۰ امتیاز)

۳. در هر کدام از موارد زیر، خروجی کد داده شده را در کادر مشخص شده، بنویسید (صرفاً خروجی را بنویسید- به توضیحات بیشتر نمره‌ای تعلق نمی‌گیرد) [۲۵ امتیاز].

در پاسخگویی به این سوال، موارد زیر را مد نظر داشته‌باشید:

- ✓ در صورتی که آرایه‌ای از اعداد (از هر نوع از انواع اعداد) تعریف شده‌بود و به برخی از عناصر آن مقدار داده نشده‌بود، فرض کنید که آن مقدار صفر است.
- ✓ مقدار پیش‌فرض هرگونه متغیر عددی که مقداردهی نشده‌باشد، برابر با صفر است.
- ✓ توجه کنید که عدم تعریف پروتوتایپ برای توابع منجر به خطای کامپایلی نمی‌شود.

الف (۸ امتیاز)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int f(int);
4
5  int main()
6  {
7      int a=5;
8      int b;
9      b = f(a++);
10     int c = f(++a);
11     printf("b = %d\n", b);
12     printf("c = %d\n", c);
13     return 0;
14 }
15
16 int f(int x){
17     return 3*x+4;
18 }
19 
```

b = 19

c = 25

(درست نوشتن هر کدام از دو مقدار فوق ۴ امتیاز دارد)

شماره دانشجویی

نام و نام خانوادگی

ب (۵ امتیاز)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int f(int);
4  int main()
5  {
6      printf("%d\n", f(5));
7      return 0;
8  }
9
10 int f(int n) {
11     if (n==0 || n==1) {
12         return n;
13     } else {
14         return n+f(n-1);
15     }
16 }
```

15

ج (۱۲ امتیاز)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  void aFuction();
4  void anotherFunction();
5
6  int x=1;
7
8  int main()
9  {
10     aFuction();
11     printf("->");
12     anotherFunction();
13     printf("->");
14     aFuction();
15     return 0;
16 }
17
18 void aFuction() {
19     printf("%d", x++);
20 }
21 void anotherFunction() {
22     int x = 5;
23     printf("%d", x++);
24 }
```

1->5->2

درست نوشتن هر کدام از ارقام فوق ۴ امتیاز دارد.

شماره دانشجویی

نام و نام خانوادگی

۴. با در نظر گرفتن کد زیر به سوالات مطرح شده جواب دهید (۱۰ امتیاز).

الف) آیا تابع زیر در خصوص تعریف و به کارگیری متغیر x دچار خطای کامپایلری می شود یا خیر؟ در هر صورت دلیل درستی و یا عدم

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int x=1;
4
5  int main()
6  {
7      printf("x0 = %d\n", x);
8
9      int x = 2;
10
11     printf("x1 = %d\n", x);
12
13     if(x>0){
14         int x=3;
15         printf("x2 = %d\n", x);
16     }
17
18     printf("x3 = %d\n", x);
19
20     return 0;
21 }
```

درستی را (حد اکثر در دو خط) توضیح دهید (۵ امتیاز).

درست است (۲ امتیاز). چون هر کدام از آن ها در قلمروی بلوکی که در آن تعریف شده اند معتبر هستند (به پاسخ های مشابه و صحیح نمره داده شود) (۳ امتیاز).

ب) خروجی کد را در کادر زیر بنویسد (۵ امتیاز).

x0 = 1	مقادیر مندرج در هر خط دارای ۱
x1 = 2	امتیاز است. ۱ امتیاز نیز به چاپ
x2 = 3	درست مقادیر تعلق گیرد (هر
x3 = 2	متغیر باید در یک خط چاپ شود).

۵. بر اساس برنامه ی زیر به سوالات پاسخ دهید (۵ امتیاز).

الف) خروجی این برنامه به ازای ۱۴۰۲ چیست (۳ امتیاز)؟

2041

ب) در دو خط توضیح دهید که تابع f با دریافت یک عدد مثل n، چه

کاری را بر اساس آن انجام می دهد؟ (از شرح خط به خط برنامه

خودداری کنید) (۲ امتیاز).

یک عدد را دریافت کرده و معکوس آن را نمایش می دهد.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void f(int);
4  int main()
5  {
6      int inputNumber;
7      scanf("%d", &inputNumber);
8      f(inputNumber);
9      return 0;
10 }
11
12 void f(int n){
13     if(n<10){
14         printf("%d",n);
15     }else{
16         printf("%d", n%10);
17         f(n/10);
18     }
19 }
20
```

شماره دانشجویی

نام و نام خانوادگی

۶. تابع زیر را در نظر بگیرید و به سوالات زیر پاسخ دهید (فرض کنید سایر بخش‌های برنامه به درستی پیاده‌سازی شده‌اند و شما صرفاً

در حال مشاهده‌ی تعریف تابع هستید) (۲۰ امتیاز).

نکته : تضمین می‌شود که مقادیر ورودی x و y مقادیری مثبت هستند.

```
int f(int x,int y) {
    if(y==1) {
        return x;
    }else{
        return x*f(x,y-1);
    }
}
```

الف (خروجی تابع به ازای ورودی‌های زیر چند می‌شود (۶ امتیاز)؟

هر سطر دارای ۲ امتیاز است.

۱. $f(2,1)$: 2.

۲. $f(2,2)$: 4.

۳. $f(2,3)$: 8.

ب (این تابع چه عملیات ریاضی را بر اساس x و y پیاده‌سازی می‌کند (۴ امتیاز) ؟

عدد صحیح x را به توان y می‌رساند.

ج (برای تابع، پشته‌ی فراخوانی (Call Stack) را (تا رسیدن به شرط پایه‌ای) مطابق با الگوی ارائه شده و برای فراخوانی $f(2,4)$ ، ترسیم

بفرمایید (پشته را از پایین به بالا ترسیم کنید و صرفاً در هر فریم، مقادیر خواسته شده – یعنی مقادیر متغیرهای x و y و مقدار برگشتی را

بنویسید و از نوشتن سایر اطلاعات اضافی خودداری کنید) (۱۰ امتیاز).

$f(2,1)$	$x=2$	$y=1$	returned value = 2
$f(2,2)$	$x=2$	$y=2$	returned value = $2*f(2,1)$
$f(2,3)$	$x=2$	$y=3$	returned value = $2*f(2,2)$
$f(2,4)$	$x=2$	$y=4$	returned value = $2*f(2,3)$

در صورت نوشتن نادرست هر کدام از مواردی که قرمز شده‌است، یک امتیاز کسر گردد. همچنین در صورتی که دانشجو پشته را ادامه داد،

صرفاً تا فراخوانی پایه‌ی اون (یعنی بالاترین سطر جدول) نمره داده شود.

بخش سوم : خرده کد نویسی (۳۰ امتیاز)

۷. برنامه‌ی زیر در برگیرنده‌ی تابعی است به نام `find If Exist` است؛ این تابع، یک آرایه (حاوی کلیدهایی از نوع عدد صحیح) را به همراه تعداد عناصر آن و یک کلید (از نوع عدد صحیح) دریافت کرده و آرایه را به منظور یافتن آن کلید جستجو می‌کند؛ در صورتی که کلید را بیابد، اندیس اولین خانه‌ای را که کلید در آن موجود است (از اندیس کوچک به بزرگ) باز می‌گرداند؛ در صورتی که کلید مذکور در آرایه موجود نباشد، عدد -۱ را باز می‌گرداند. تابع `find If Exist` را به گونه‌ای تکمیل کنید که برنامه‌ی زیر، این خواسته را بر آورده کند؛ کد خود را در کادر بنویسید و فرض کنید به جای کامنت خط ۱۵ جایگزین می‌شود (۱۰ امتیاز).

مثال : به ازای ورودی‌های ۲۳، ۱ و ۵۰۰ به ترتیب خروجی‌های ۱، ۴ و -۱ چاپ می‌شود (در هر بار اجرا فقط یک ورودی دریافت

شده و یک خروجی نمایش داده می‌شود).

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  | int findIfExists(int[],int,int);
4  | int main()
5  | {
6  |     int key;
7  |     scanf("%d", &key);
8  |     int keys[]={11,23,2,345,1,2,1};
9  |     printf("index is %d\n", findIfExists(keys,7,key));
10 |     return 0;
11 | }
12
13 | int findIfExists(int keys[], int size, int key){
14 |     int foundIndex=-1;
15 |     //Write Your Code Here.
16 |     return foundIndex;
17 | }
18 |
```

```

for(int i=0; i<size; i++){
    if(keys[i]==key){
        foundIndex = i;
        break; //first found Element
    }
}
```

نوشتن صحیح حلقه : ۳ امتیاز

(بابت عدم درست نوشتن شرط حلقه و یا روش تغییر شمارنده،

۱ امتیاز کسر شود)

چک کردن صحیح برابری کلید (۳ امتیاز)

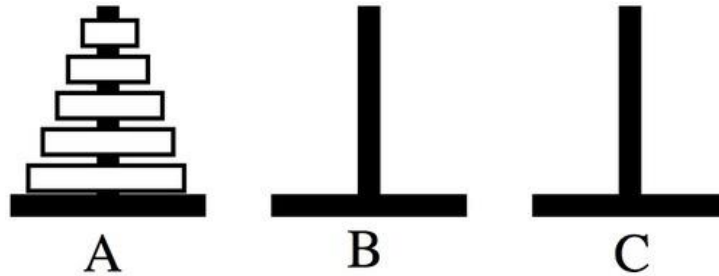
تغییر درست `flag` (۱ امتیاز)

وجود `break` (و یا هر روشی برای توقف حلقه در مواجهه با

اولین عنصر - (۱ امتیاز)

۸. (برج‌ها و دیسک‌ها – ۲۰ امتیاز) : فرض کنید سه برج داریم با نام‌های A و B و C. بر روی برج A تعدادی دیسک با سایزهای و به ترتیب

اندازه (از پایین به بالا و از بزرگ به کوچک) چیده شده‌اند.



برج‌ها و دیسک‌ها

می‌خواهیم یک برنامه به زبان C و به صورت بازگشتی بنویسیم که بتوانیم با در نظر گرفتن شرایط زیر، تمام دیسک‌ها را از برج A به برج C منتقل

نماییم : شرط ۱ : در هر لحظه فقط می‌توانید یک دیسک را از یک برج به برج دیگر منتقل کنید؛ شرط ۲ : هیچ دیسک بزرگی نباید بر روی

دیسک کوچک‌تری قرار گیرد؛ شرط ۳ : در هر جابجایی صرفاً دیسک بالایی از هر برج می‌تواند منتقل شود.

روش پیاده‌سازی شده : به منظور جابجایی تمامی دیسک‌ها از برج A به برج C لازم است از یکی از برج‌ها به عنوان برج کمکی استفاده کنیم و

جابجایی را به واسطه‌ی آن انجام دهیم. به عنوان نمونه در صفحه‌ی بعدی، مراحل جابجایی دیسک‌ها را از برج A به برج C (به کمک برج B) نشان

داده‌ایم. برای پیاده‌سازی این روش به صورت بازگشتی در زبان C، تابعی را به نام move Disk Tower تعریف کردیم که در آن تعداد دیسک‌ها،

نام برج مبدا (Source Tower)، نام برج کمکی (Aux Tower) و نام برج مقصد (Destination Tower) را به ترتیب به آرگومان آن می‌دهیم.

برای حل مسئله لازم است در بخش پایه، یک دیسک را از برج مبدا به برج مقصد منتقل نمایم و در بخش بازگشت موارد زیر صورت بگیرد :

✓ اول از همه، تمام دیسک‌ها (بجز بزرگ‌ترین دیسک) از برج A به برج B منتقل شوند (کادر ۱ در مثال شکل صفحه بعدی).

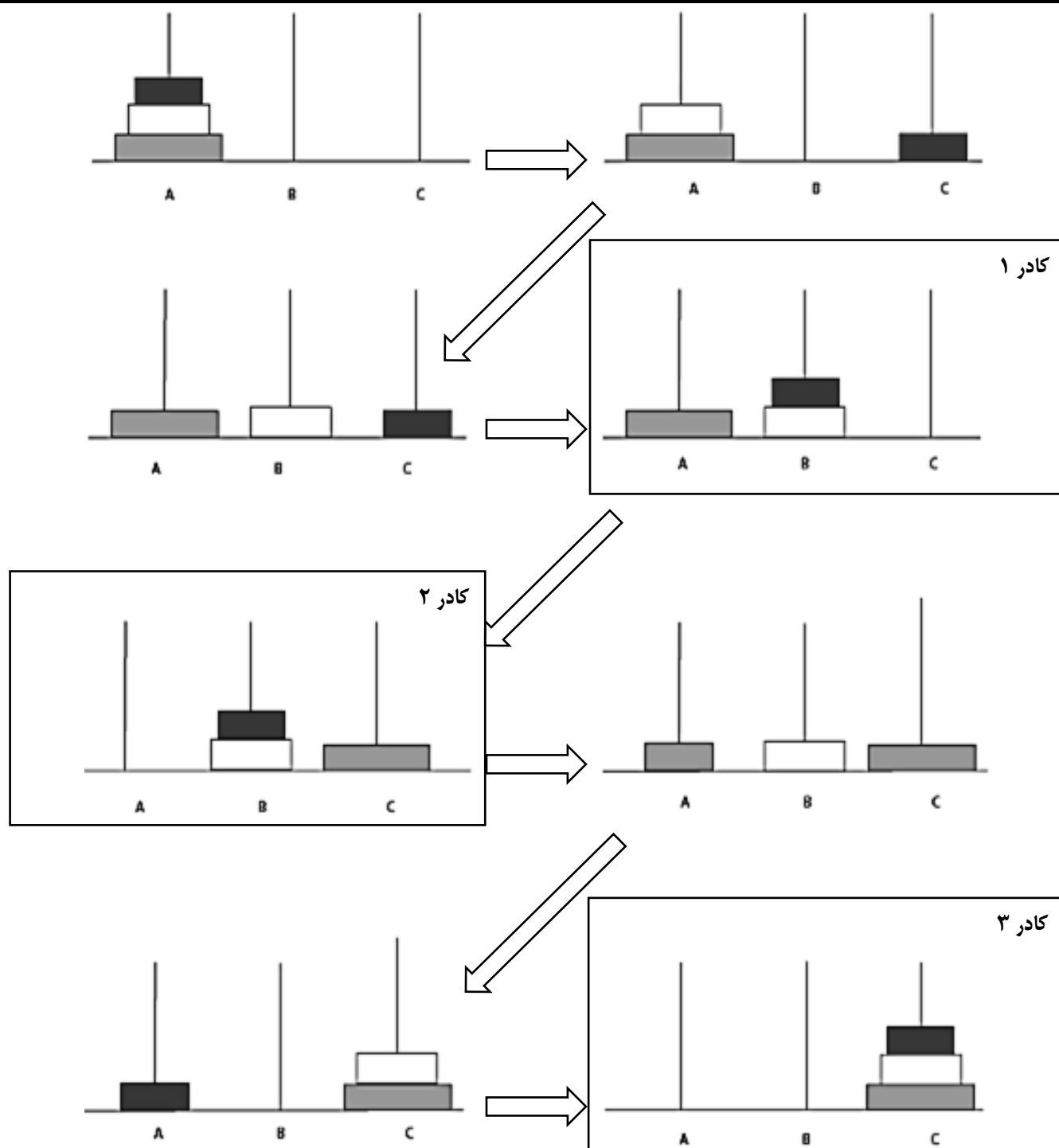
✓ سپس بزرگ‌ترین دیسک به برج مقصد (برج C) جابجا شود (کادر ۲ در مثال شکل صفحه بعدی).

✓ در نهایت تمامی دیسک‌های موجود در برج B به برج مقصد (برج C) جابجا شود (کادر ۳ در مثال شکل صفحه بعدی).

توجه کنید که در جابجایی‌ها، هر کدام از برج‌های A و B و C می‌توانند به عنوان برج کمکی (Aux Tower) مورد استفاده قرار گیرند.

شماره دانشجویی

نام و نام خانوادگی



مثالی از به کارگیری روش برای ۳ دیسک : به ترتیب از بالا به پایین - از چپ به راست

شماره دانشجویی

نام و نام خانوادگی

حال از شما می‌خواهیم برنامه‌ی زیر را به گونه‌ای تکمیل کنید که با دریافت تعداد دیسک‌ها، به ترتیب مراحل جابجایی آن‌ها را چاپ نماید.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     int numOfDisks=0;
7     printf("Enter Number of Disks : ");
8     scanf("%d", &numOfDisks);
9     moveDiskTower(numOfDisks, 'A', 'B', 'C');
10    return 0;
11 }
12
13 void moveDiskTower(int numberOfDisks, char sourceTower, char auxTower, char destinationTower){
14     if(numberOfDisks == 1){
15         //write Base Operation here
16     }else{
17         //write Recursion Part here
18     }
19 }
20 void printMovement(char source, char destination){
21     printf("%c -> %c\n", source, destination);
22 }
23
```

الف) همان‌طور که مشاهده می‌کنید، در حالت پایه در صورتی که یک دیسک داشته باشیم، آن را مستقیم از مبدا به مقصد جابجا می‌کنیم. با کمک تابع print

movement، کدی را برای جایگزینی در خط ۱۴ (به جای کامنت) بنویسید (۵ امتیاز).

هر آزمون ۲ امتیاز دارد. `printMovement(sourceTower, destinationTower);`

ب) برای بخش بازگشت تابع (که به جای کامنت نوشته شده در خط ۱۷ و در بدنه‌ی else چاپ می‌شود)، به گونه‌ای از تابع move Disk Tower و به صورت

بازگشتی استفاده نمایید که مطابق با صورت مسئله، دیسک‌ها از ستون A به کمک ستون B به ستون C منتقل شوند (در صورت لزوم از تابع print movement

استفاده کنید) (۱۵ امتیاز). – به راه حل‌های غیر بازگشتی (حتی اگر کاملاً درست باشند) نمره‌ای تعلق نمی‌گیرد.

```
moveDiskTower(numberOfDisks-1, sourceTower, destinationTower, auxTower);
printMovement(sourceTower, destinationTower);
moveDiskTower(numberOfDisks-1, auxTower, sourceTower, destinationTower);
```

هر کدام از سه خط ۱ امتیاز دارد (فارغ از درستی – یعنی دانشجو متوجه شده باشد که در بدنه فراخوانی بازگشتی داریم)

درست نوشتن هر کدام از آزمون‌های فراخوانی بازگشتی ۱ امتیاز دارد.

درست نوشتن آزمون‌های خط میانی هر کدام ۲ امتیاز دارد.

در صورتی که شکل برنامه (در بالای صفحه) واضح نیست، در صفحه‌ی بعدی مجدداً نوشته شده‌است.

شماره دانشجویی

نام و نام خانوادگی

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int numOfDisks=0;
```

```
    printf("Enter Number of Disks : ");
```

```
    scanf("%d", &numOfDisks);
```

```
    moveDiskTower(numOfDisks,'A','B','C');
```

```
    return 0;
```

```
}
```

```
void moveDiskTower(int numberOfDisks, char sourceTower, char auxTower, char destinationTower){
```

```
    if(numberOfDisks == 1){
```

```
        //write Base Operation here
```

```
    }else{
```

```
        //write Recursion Part here
```

```
    }
```

```
}
```

```
void printMovement(char source, char destination){
```

```
    printf("%c -> %c\n", source, destination);
```

```
}
```

پایان آزمون

پیروز، شاد و تن درست باشید