



تاریخ آزمون: یکشنبه ۱۳۹۰/۱۰/۲۵

وقت: ۱۵۰ دقیقه

آزمون پایان ترم گروه‌های ۱ تا ۹

توجه ۱: پاسخ هر سوال را در یک صفحه‌ی مجزا از پاسخنامه وارد نمایید.  
 توجه ۲: صورت سوال‌ها را در ابتدای آزمون مرور نموده، سوال‌های خود را در چهل دقیقه‌ی اول آزمون بپرسید. پس از گذشت این زمان به هیچ سوالی پاسخ گفته نخواهد شد.  
 توجه ۳: برای پاسخ به سوالات، به‌جز کتابخانه‌های ورودی/خروجی (C و C++)، مجاز به استفاده از کتابخانه‌ی دیگری نیستید.

## سوال ۱) سوال‌های کوتاه (۱۰ نمره)

الف. تفاوت بین اشاره‌گر ptr در دو تعریف زیر چیست؟ به طور مختصر توضیح دهید.

```
const int * ptr;
int * const ptr;
```

ب. خروجی برنامه‌ی زیر چیست؟

```
#include <iostream>
int test(int &b) {
    int a = 4;
    b = b < 10 ? b : 2 * b;
    return a + b;
}

int main() {
    int a = 10;
    test(a);
    std::cout << a;
    return 0;
}
```

## سؤال ۲: مقایسه‌ی دو رشته (۱۵ نمره)

تابع strcmp از کتابخانه‌ی استاندارد cstring را در نظر بگیرید. ورودی این تابع به ترتیب دو رشته‌ی s1 و s2 است که هر دو با کاراکتر '\0' خاتمه می‌یابند. اگر s1 برابر s2 بود، تابع مقدار صفر، اگر s1 کوچک‌تر (از نظر ترتیب حروف در جدول اسکی) از s2 بود، تابع مقدار -۱، و در غیر این صورت تابع مقدار +۱ برمی‌گرداند. این تابع را به شکل زیر پیاده‌سازی کنید:

```
int strcmp(const char *s1, const char *s2);
```

خروجی	s2	s1	مثال
1	Dorood	Salam	۱
0	ABC	ABC	۲
1	ABC	ABCDE	۳
-1	ABD	ABCDE	۴

### سوال ۳) معکوس آرایه (۲۰ نمره)

دنباله‌ای از اعداد صحیح که در یک آرایه ذخیره شده‌اند، داده شده است. می‌خواهیم تابعی به صورت بازگشتی بنویسیم که این اعداد را درجا معکوس نماید. این تابع را بصورت زیر پیاده‌سازی نمایید.

```
void reverse(int *start, int *end);
```

پارامترهای ورودی این تابع دو اشاره‌گر `start` و `end` هستند که `start` به آغاز دنباله‌ی مورد نظر در آرایه و `end` به عنصر پایانی دنباله اشاره می‌کند. مثلاً برای به دست آوردن معکوس اعداد موجود در آرایه‌ی `a` که به صورت `int a[10]` تعریف شده، تابع `reverse` به صورت زیر فراخوانی می‌شود:

```
reverse(a, a + 9);
```

به‌طور کلی با فراخوانی `reverse(a + i, a + j)`، عناصری از آرایه‌ی `a` که اندیسشان در بازه‌ی `[i, j]` قرار دارد، به عنوان یک دنباله در نظر گرفته شده، و معکوس عناصر موجود در آن دنباله به صورت بازگشتی محاسبه می‌شود.

توجه: در پیاده‌سازی این تابع نباید آرایه‌ی اضافی تعریف کنید.

مثال	محتویات آرایه قبل از فراخوانی تابع	محتویات آرایه پس از فراخوانی تابع
۱	1, 2, 3, 4, 5	5, 4, 3, 2, 1
۲	93, 23, 40, -261	-261, 40, 23, 93

### سوال ۴) ماتریس متقارن (۲۵ نمره)

تابعی بنویسید که یک آرایه‌ی دو بعدی و تعداد سطرها و ستون‌های آن را به عنوان ورودی تابع دریافت نموده و به عنوان خروجی، اندازه‌ی (تعداد سطرهای) بزرگ‌ترین زیر ماتریس متقارن آن را برگرداند. یک ماتریس (زیر ماتریس) `A` را متقارن گویند اگر و فقط اگر تعداد سطرها و ستون‌های آن برابر بوده و به ازای هر `i` و `j` متعلق به آن ماتریس (زیرماتریس)، `A[i,j]` برابر با `A[j,i]` باشد.

مثال	ورودی	خروجی
۱	1 2 1 2 2 1 1 2 3 2 2 1	3
۲	1 1 2 2 2 1 1 2 3 2 3 1	1

## سؤال ۵: کلاس اعداد بزرگ (۳۰ نمره)

می‌خواهیم کلاسی با نام `HugeInteger` برای ذخیره، نمایش و انجام برخی محاسبات بر روی اعداد بسیار بزرگ تعریف کنیم. همانگونه که می‌دانید اگر بخواهیم عددی با ۷۰ رقم را در حافظه ذخیره کنیم، هیچکدام از انواع داده‌ای که می‌شناسیم (مانند `int`، `float`، `double` و ...) نمی‌توانند این مقدار را ذخیره کنند. می‌خواهیم این مشکل را با استفاده از این کلاس حل کنیم. برای حل این مشکل از تکنیک استفاده از آرایه‌های بزرگ و ذخیره هر رقم عدد مورد نظر در یک خانه از آرایه، استفاده می‌کنیم.

الف - برای کلاس فوق، یک آرایه حاوی ۱۰۰ عنصر از نوع داده‌ای اعداد صحیح، و به صورت داده‌ی خصوصی (`private`) تعریف کنید (از این آرایه برای ذخیره‌ی یک عدد صحیح بسیار بزرگ استفاده کنید) (۳ نمره).

ب - در این کلاس باید چهار تابع عضو (متد) نیز به شرح زیر تعریف کنید. تابع‌های عضو همگی عمومی (`public`) بوده و باید خارج از کلاس قابل دسترسی باشند (۱۸ نمره).

```
void input (void);
```

تابع فوق وظیفه گرفتن یک عدد بزرگ (مثبت) از ورودی و ذخیره آن در آرایه را برعهده دارد. عدد ورودی بصورت یک رشته از ارقام 0 تا 9 با طول حداکثر ۱۰۰ توسط کاربر داده می‌شود. آرایه را به گونه‌ای مقداردهی کنید که اندیس کم ارزش‌تر حاوی رقم کم‌ارزش‌تر باشد. اگر عدد ورودی کمتر از ۱۰۰ رقم داشت، رقم‌های باقی‌مانده در آرایه با صفر پر می‌شوند.

```
void output (void);
```

تابع فوق وظیفه نمایش عدد بزرگ ذخیره شده در آرایه خصوصی را برعهده دارد. توجه داشته باشید که اعداد صفر سمت چپ عدد نباید چاپ شوند. مثلاً چاپ عدد بصورت 000120 غلط است و باید بصورت 120 چاپ شود.

```
int getDigit (int i);
```

تابع فوق وظیفه بازگرداندن رقم `ام` عدد بزرگ را برعهده دارد. کم ارزش‌ترین رقم عدد بزرگ، رقم صفرم آن محسوب می‌شود.

```
int compare (HugeInteger inputHI);
```

این تابع یک شی از نوع `HugeInteger` را به عنوان ورودی دریافت و به مقایسه عدد ورودی با عدد ذخیره شده می‌پردازد. اگر عدد ورودی بزرگتر بود مقدار 1، اگر عدد ورودی کوچکتر بود مقدار -1، و اگر هر دو عدد برابر بودند، مقدار صفر را بر می‌گرداند.

ج - در قسمت سازنده‌ی کلاس (`constructor`) به آرایه‌ی خصوصی مقدار اولیه داده و عناصر آن را برابر صفر قرار دهید (۴ نمره).

د - همچنین برنامه‌ای برای استفاده از این کلاس به شرح زیر بنویسید (۵ نمره).

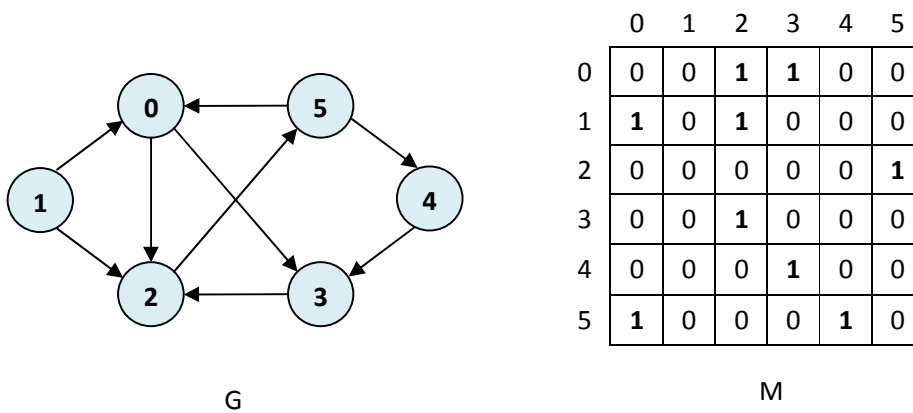
دو عدد بزرگ تعریف کنید. مقدار آنها را از ورودی بگیرید. آنها را مقایسه نموده و عدد بزرگتر و کوچکتر را به ترتیب چاپ نمایید. در صورت مساوی بودن دو عدد، در خروجی اعلام نمایید که دو عدد مساوی هستند و تنها یکی از آنها را چاپ نمایید.

توجه: این کلاس فقط برای اعداد صحیح مثبت تعریف شود و اعداد منفی یا اعشاری مد نظر نیستند.

«سوال ویژه دانشجویان مهندسی کامپیوتر و مهندسی برق»

سوال ۶) یافتن مسیر در گراف (۲۵ نمره)

گراف جهت‌دار  $G$  با  $n$  رأس را در نظر بگیرید. فرض کنید رأس‌های گراف با عددهای  $0$  تا  $n-1$  شماره‌گذاری شده‌اند. می‌توان این گراف را به وسیله‌ی یک ماتریس مجاورت  $M$  به صورت زیر نمایش داد: اگر از رأس  $i$  به  $j$  یک یال در گراف  $G$  وجود داشته باشد، قرار می‌دهیم  $M[i][j] = 1$ ، در غیر این صورت  $M[i][j] = 0$ . به طور مثال، در شکل زیر ماتریس مجاورت  $M$  مربوط به گراف جهت‌دار  $G$  نمایش داده شده است.



تابعی بنویسید که با دریافت ماتریس مجاورت یک گراف جهت‌دار  $G$ ، و شماره‌ی دو رأس  $u$  و  $v$  در این گراف، تعیین کند که آیا مسیری از رأس  $u$  به رأس  $v$  در گراف  $G$  وجود دارد یا خیر. به طور مثال، در گراف بالا از رأس  $0$  به رأس  $4$  مسیر وجود دارد، ولی از رأس  $5$  به رأس  $1$  مسیری وجود ندارد. این تابع را به صورت زیر پیاده‌سازی کنید:

```
bool path_exists(int M[][MAX_NODES], int n, int u, int v);
```

پارامترهای ورودی این تابع عبارتند از ماتریس مجاورت  $M$  (که هر سطر آن  $MAX\_NODES$  عنصر دارد)، تعداد رأس‌های گراف،  $n$ ، با شرط  $0 \leq u, v \leq n-1$  و شماره‌ی دو رأس  $u$  و  $v$  در این گراف با شرط  $0 \leq u, v \leq n-1$ .

خروجی این تابع، در صورت وجود مسیر از رأس  $u$  به رأس  $v$  در این گراف، برابر  $true$  و در غیر این صورت  $false$  خواهد بود.

نکته: فرض کنید ماتریس  $M$  بیرون تابع به صورت زیر تعریف شده است:

```
#define MAX_NODES 100
int M[MAX_NODES][MAX_NODES];
```

«موفق باشید»