

سری سوم تمرینات

مهلت ارسال تا ساعت ۲۳:۵۹ جمعه ۱۴ دی ۱۳۹۷

دانشگاه صنعتی شریف - دانشجویان مهندسی برق و مکانیک

شماره درس: ۴۰۱۵۳      شماره گروه: ۴      تعداد واحد: ۳

مدرس: سعید امراللهی بیوکی

نیمسال اول (پاییز) ۹۸-۱۳۹۷

---

## ۱ عملیات ماتریس‌ها

دو ماتریس  $A_{5 \times 4}$  و  $B_{4 \times 5}$  را در نظر بگیرید و توابع زیر را پیاده‌سازی کنید.

الف) تابعی بنویسید که ماتریس  $A$  را در ماتریس  $B$  ضرب کند و ماتریس جواب را چاپ کند.

```
void mult(int A[5][4], int B[4][5]);
```

ب) تابعی بنویسید که سطر  $n$  ام ماتریس  $A$  را با ستون  $m$  ام ماتریس  $B$  عوض کند.

```
void change(int A[5][4], int B[4][5], int n, int m);
```

**ورودی:** در ابتدا مقادیر خانه‌های ماتریس‌های  $A$  و  $B$  داده می‌شود و سپس در خط بعدی عددهای  $n$  و  $m$  می‌آیند.

**خروجی:** ابتدا تابع `mult`، سپس تابع `change` و دوباره تابع `mult` صدا زده می‌شوند تا ماتریس‌های خروجی مورد نظر چاپ شوند.

**نمونه ورودی:**

```
-2 -2 -2 -2
-1 -1 -1 -1
0 0 0 0
1 1 1 1
2 2 2 2
-1 -1 -1 -1 -1
0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
1 2
```

**نمونه خروجی:**

```
-4 -4 -4 -4 -4
-2 -2 -2 -2 -2
0 0 0 0 0
2 2 2 2 2
4 4 4 4 4
6 -4 6 6 6
-2 8 -2 -2 -2
0 0 0 0 0
2 -8 2 2 2
4 -16 4 4 4
```

## ۲ کار با رشته‌نویسه‌ها

تابع زیر را پیاده‌سازی کنید:

```
int strncmp( const char *lhs, const char *rhs, size_t count);
```

این تابع حداکثر به اندازه count دو رشته‌نویسه lhs و rhs، را با هم مقایسه می‌کند. مقدار بازگشت این تابع اینگونه است:

- مقدار 1- در صورتیکه lhs از نظر لغتنامه‌ای (Lexicographically) قبل از rhs باشد.
  - مقدار 0 در صورتیکه lhs با rhs مساوی باشد.
  - مقدار 1 در صورتیکه lhs از نقطه نظر لغتنامه‌ای (Lexicographically) بعد از rhs باشد.
- \* دقت کنید که اگر طول رشته نویسه lhs یا rhs از count کمتر بود، مقایسه متوقف می‌گردد و نتیجه را برمی‌گرداند.

**ورودی:** در دو خط ابتدایی، دو رشته داده می‌شود و سپس در خط بعدی، مقدار count می‌آید.

**خروجی:** در تنها خط خروجی، خروجی این تابع را نمایش دهید.

**نمونه ورودی ۱:**

```
Koala  
Babai  
2
```

**نمونه خروجی ۱:**

```
1
```

**نمونه ورودی ۲:**

```
Art  
Artificial  
3
```

**نمونه خروجی ۲:**

```
0
```

### ۳ رمزنگاری رشته‌نویسه‌ها با روش سزار

می‌خواهیم آرایه‌ای از رشته‌نویسه‌ها را به روش سزار رمزنگاری کنیم. در این روش هر نویسه با عدد 13 جمع می‌شود. پرواضح است که گیرنده پیام رمز شده اگر بخواهد به پیام اصلی دسترسی پیدا کند باید از هر نویسه 13 واحد کم کند. برای این منظور دو تابع نیاز است؛ یکی encrypt برای رمزنگاری و دیگری decrypt برای رمزگشایی. این دو تابع به عنوان ورودی یک رشته‌نویسه را می‌گیرند و به ترتیب با افزودن و کاستن 13 واحد به/از رشته‌نویسه آن را رمزنگاری/رمزگشایی می‌کنند.

برنامه‌ای بنویسید که تعدادی رشته‌نویسه را از فایل encrypted.txt بخواند و رشته‌ها رمزنگاری کرده و در فایل decrypted.txt بنویسد. سپس به عنوان آزمایش درستی تابع نوشته شده، پیام‌ها را از فایل decrypted.txt بخوانید، آنها را رمزگشایی کنید و در خروجی چاپ کنید.

**ورودی:** تنها یک فایل encrypted.txt کنار فایل برنامه قرار دارد.

**خروجی:** محتوای رمزگشایی شده که باید دقیقاً مشابه محتوای فایل encrypted.txt می‌باشد را چاپ کنید.

نمونه ورودی ( محتوای فایل encrypted.txt ):

```
This sentence is about to be encrypted.  
So is this.
```

نمونه خروجی ( محتوای فایل decrypted.txt ):

```
auvÇ-Çr{ür{pr-vÇ-no|éü-ü|-or-r{pλå}ürq;‡`|-vÇ-üuvÇ;
```

نمونه خروجی ( محتوی کنسول):

```
This sentence is about to be encrypted.  
So is this.
```

- توجه کنید فایل برنامه را فقط آپلود کنید و درستی پاسخ شما در تحویل حضوری مشخص می‌شود.

## ۴ حساب بانکی

می‌خواهیم مشخصات حساب بانکی تعدادی سپرده گذار در یک بانک فرضی را ویرایش کنیم. هر حساب بانکی شامل المان‌های زیر است:

- نام و نام خانوادگی سپرده گذار (char \* name)
  - شماره حساب (char \* account\_num)
  - موجودی حساب (به ریال) (int balance)
- این بانک ۴ عمل را پشتیبانی می‌کند:
- برداشت از حساب: مقدار برداشت درخواستی و شماره‌ی حساب را به عنوان ورودی می‌گیرد و باقیمانده‌ی موجودی حساب را نمایش می‌دهد. اگر مبلغ درخواستی بیشتر از موجودی حساب باشد، پیام Not Enough Balance نمایش داده می‌شود.
- ```
void withdraw( char * account_num, int amount);
```
- سپرده‌گذاری در حساب: مقدار سپرده درخواستی و شماره‌ی حساب را به عنوان ورودی می‌گیرد و باقیمانده‌ی موجودی حساب را باز می‌گرداند.
- ```
void deposit( char * account_num, int amount);
```
- انتقال وجه از یک حساب به حساب دیگر: شماره حساب‌های مبدأ و مقصد و مبلغ تراکنش را به عنوان ورودی می‌گیرد و موجودی حساب مبدأ را نمایش می‌دهد. اگر مبلغ درخواستی بیشتر از موجودی حساب مبدأ باشد، پیام Not Enough Balance نمایش داده می‌شود.
- ```
void transfer( char * src, char * dest, int amount);
```
- بازبینی موجودی حساب: شماره حساب را به عنوان ورودی می‌گیرد و موجودی حساب را نمایش می‌دهد.
- ```
void getBalance( char * account_num);
```
- در ابتدا یک استراکت account تعریف کنید که توابع ذکر شده را پاسخگو باشد. موجودی اولیه را برابر صفر در نظر بگیرید.

**ورودی:** در خط اول برنامه n و در n خط بعدی، نام و شماره حساب افراد می‌آید. در خط بعدی m و در m خط بعدی عملیات مورد نظر می‌آید (0 < m, n < 100).

**خروجی:** در خطوط مناسب به ترتیب عبارات مناسب توابع را نمایش دهید.

**نمونه ورودی:**

```
2
Kimia Mohamadi 73001
Arman Bahadori 35080
4
deposit 35080 100000
withdraw 73001 1000
transfer 35080 73001 60000
getBalance 73001
```

نمونه خروجی:

100000

Not Enough Balance

40000

60000