

## تمرین سری پنجم مبانی برنامه‌نویسی نیم‌سال اول ۹۶-۹۷

به سه نکته توجه کنید

الف) زمان تحویل تا قبل از ۲۲:۰۰ روز پنجشنبه ۹ آذر ماه است.

ب) پاسخ مسائل را در سایت Quera ارسال نمایید.

ج) در این تمرینات تا حد ممکن برنامه‌های خود را به توابع مختلف (هر چند تابعی کوچک) بشکنید.

۱-

الف) تابع `reg_index` اعداد `n` و `m` و `dir` را به عنوان ورودی می‌گیرد که در آن `n` یک عدد صحیح مثبت چند رقمی، `m` یک رقم و `dir` یا `+1` یا `-1` است.

```
int dig_index(int n, int m, int dir)
```

خروجی این تابع یک عدد است که مشخص میکند رقم `m` چندمین رقم از سمت راست (اگر `dir = +1` باشد) یا سمت چپ (اگر `dir = -1` باشد) عدد `n` است. اگر رقم `m` در `n` نباشد ۰ تولید می‌شود. برای مثال

```
dig_index(1000, 0, -1) → 2
```

```
dig_index(1000, 0, 1) → 1
```

```
dig_index(1000, 2, 1) → 0
```

تعریف این تابع را بنویسید.

ب) با استفاده از این تابع، تعریف تابع زیر را بنویسید که یک عدد را می‌گیرد و اگر عدد متقارن باشد ۱ و در غیر این صورت ۰ تولید می‌کند

```
int sym(int n)
```

ج) يك برنامه بنویسید که يك عدد صحیح مثبت را بگیرد و تعیین کند که متقارن است یا نه؟ (۱۵ نمره)

فرمت ورودی:

<n>

فرمت خروجی:

<True>/<False>

۲- تابعی به اسم `max_factor` بنویسید که عدد صحیح `n` را بگیرد و بزرگترین عامل اول آن را برگرداند. تابع دیگری به اسم `all_factors` بنویسید که عدد صحیح `n` را بگیرد و با استفاده از تابع `max_factor`

همه عامل‌هاي اول عدد  $n$  را چاپ کند. تعداد دفعات تکرار هر عامل آن نیز مشخص شود.  
 برنامه‌اي بنویسید که عدد  $m$  را از کاربر بگیرد و برای همه اعداد 1 تا  $m$  تابع `all_factor` را فراخوانی کند.  
 (۱۵ نمره)  
 (۱۰ نمره اضافي در صورتي که `all_factors` به صورت بازگشتي باشد).

فرمت ورودی:

<m>

فرمت خروجی:

1: <factor>^<repeat> \* <factor>^<repeat> ...

2: <factor>^<repeat> \* <factor>^<repeat> ...

...

m: <factor>^<repeat> \* <factor>^<repeat> ...

۳- ترکیب  $m$  از  $n$  را می‌توان با رابطه بازگشتي به صورت زیر نوشت (۲۰ نمره)

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$$

الف) شرط (شرط‌هاي) مناسب پایه برای اتمام رابطه بازگشتي را تعیین کنید.

ب) تابع بازگشتي برای پیاده‌سازي ترکیب  $m$  از  $n$  را بنویسید:

```
int combination(int n, int m)
```

ج) برنامه‌ای بنویسید که عدد  $n$  و  $m$  را از کاربر گرفته و ترکیب  $m$  از  $n$  و تعداد دفعات فراخوانی تابع `combination` را محاسبه کند.

۴- برای حدس Goldbach طبق الگوریتم شرح داده شده در کلاس توابع زیر را پیدا سازی کنید. (۲۰ نمره)

```
int is_prime(int n);
```

```
int next_prime_number(int n);
```

```
int check_Goldbach(int n);
```

با استفاده از این توابع، تابع `main` را بنویسید که يك عدد را از کاربر بگیرد و این حدس را برای تمام اعداد زوج بزرگتر از دو و کوچکتر از این عدد بررسی کرده و دو مقدار عدد اول مورد نظر را چاپ کند.

فرمت ورودی:

<n>

فرمت خروجی:

Even number: <e>  
Prime 1: <p>  
Prime 2: <p>  
Even number: <e>  
Prime 1: <p>  
Prime 2: <p>

...

## ۵- بازی حدس زدن اعداد: (۳۰ نمره)

در تمرین هدف این است که بازی ساده حدس اعداد نوشته شود. این برنامه به این صورت عمل می‌کند که در ابتدا نوبت کامپیوتر است بنابراین برنامه عددی بین ۰ تا ۱۰۰۰ را در نظر می‌گیرد. کاربر سعی می‌کند این عدد را پیدا کند. به ازای هر عددی که کاربر حدس می‌زند برنامه مشخص می‌کند که آیا این حدس از عدد مد نظر بزرگتر یا کوچکتر است. زمانی که کاربر عدد مد نظر کامپیوتر را به درستی حدس زد، نوبت کاربر می‌شود و این بار کامپیوتر سعی می‌کند عدد را حدس بزند و کاربر برای هر حدس کامپیوتر، بزرگی یا کوچکی آنرا مشخص می‌کند. با حدس درست برنامه نوبت کامپیوتر می‌شود و این بازی ادامه می‌یابد تا جایی که کاربر سه بار عدد ۰ را پشت سر هم وارد کند. زمانی که بازی تمام شد در انتها برنامه تعداد حدس‌های غلطی که کاربر و کامپیوتر در کل بازی داشته‌اند را چاپ می‌کند. برنامه خود را طوری بنویسید که **به طور متوسط کمترین مقایسه‌ها** را داشته باشد.

**اختیاری.** برنامه کامپیوتری در بازی تقلب می‌کند!!!. برنامه هر بار جواب درست را در خصوص کوچک یا بزرگ بودن حدس کاربر نمی‌دهد! بلکه یک مقدار  $x$  وجود دارد که نشان دهنده احتمال تقلب کامپیوتر است. مثلاً اگر  $x=0$  باشد یعنی هیچ تقلبی در کار نیست.  $x=0.5$  به معنی غلط بودن جواب‌های کامپیوتر در ۵۰٪ موارد است. برنامه بازی را به نحوی تغییر دهید که در ابتدای بازی مقدار  $x$  را از کاربر بگیرد و در ادامه به همان میزان در بازی تقلب کند.