به نام خدا

دستور کار کارگاه مبانی کامپیوتر و برنامه‌نویسی

جلسه هشتم

# حلقه‌ها در C

در این جلسه قصد داریم مروری بر حلقه‌ها در زبان C داشته باشیم.

۱. برنامه‌ای بنویسید که با گرفتن یک قرن میلادی به عنوان ورودی، تاریخ‎هایی که جمع روز و ماه در آن تاریخ میلادی برابر با سال (سال را به صورت دو رقمی در نظر بگیرید) آن تاریخ می‌گردد را برای قرن داده شده، پیدا کند.

مثلا برای قرن ۲۱ داریم:

21

19 1 2020

17 2 2019

15 3 2018

13 4 2017

11 5 2016

9 6 2015

7 7 2014

5 8 2013

3 9 2012

1 10 2011

۲. برنامه‌ای بنویسید که با دریافت عدد n از کاربر مقدار سری زیر را یکبار از راست به چپ و یکبار از چپ به راست محاسبه کند.

این برنامه را به ازای nهای مختلف اجرا کنید. آیا دو مقدار محاسبه شده یکسان می‌باشند؟ در مورد علت آن با مدرس آزمایشگاه خود گفت و گو نمایید.

۳. برنامه‌ای بنویسید که با دریافت عدد n از کاربر یک مثلث متساوی الساقین قائم الزاویه مشابه شکل زیر را رسم نمایید. اضلاع قائمه این مثلث شامل n ستاره می‌باشند.

ورودی:

5

خروجی:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

۴. می‌خواهیم فرجه‌ی nام عدد m را بدون استفاده از توابع کتابخانه‌ای math.h محاسبه نماییم.

به این منظور به ترتیب زیر عمل می‌کنیم:

* فرض می‌کنیم فرجه‌ی nام عدد m برابر با می‌باشد.

توان nام را با m مقایسه می‌کنیم.

* + اگر از m بزرگتر بود اینبار با ادامه می‌دهیم.
  + اگر از m کوچکتر بود اینبار با ادامه می‌دهیم.

این الگوریتم فرجه‌ی nام عدد m را به صورت صحیح می‌یابد. برنامه‌ای بنویسید که با دریافت n و m از کاربر این الگوریتم را شبیه‌سازی کند.

۵. برنامه‌ای بنویسید که با دریافت n عدد مرتب شده که با عدد صفر خاتمه پیدا می‌کنند، عدد k را از کاربر دریافت کرده و آن را در بین اعداد داده شده پیدا کرده و اندیس آن را خروجی دهد. به عنوان مثال:

ورودی:

1 2 3 4 5 6 0

3

خروجی:

3

الگوریتم شما چند مقایسه را برای پیدا کردن عدد k صورت می‌دهد؟ آیا مرتب بودن این مجموعه تاثیری در الگوریتم شما دارد؟

یکی از الگوریتم‌های معروف جهت جستجو در مجموعه‌های مرتب، الگوریتم جسجوی دودویی می‌باشد:

**function** binary\_search(A, n, T):

L := 0

R := n − 1

**while** L <= R:

m := floor((L + R) / 2)

**if** A[m] < T:

L := m + 1

**else if** A[m] > T:

R := m - 1

**else**:

**return** m

**return** unsuccessful

۶. همانطور که می‌دانید عدد اول، عددی می‌باشد که تنها بر خودش و یک بخش‌پذیر می باشد. فرض کنید عددهای چرت-اول عددهایی باشند که تنها بر خودشان، یک و یک عدد دیگر بخش‌پذیر باشند، مثلا عدد ۴ یک عدد چرت-اول می‌باشد زیرا بر خودش، یک و عدد ۲ بخش‌پذیر می‌باشد. برنامه‌ای بنویسید که یک عدد را دریافت کرده و مشخص کند این عدد چرت-اول می‌باشد یا خیر.

آیا بین این سوال و سوال ۴ ارتباطی می‌بینید؟ اگر این ارتباط را می‌بینید می‌توانید سوال اصلی در [اینجا](http://codeforces.com/problemset/problem/230/B) را پاسخ دهید. 😊

