

کد نامه

ویژه‌ی دانش‌جویان مبانی برنامه‌سازی نیم‌سال اول ۹۹-۰۰ دانشکده‌ی مهندسی و علم کامپیوتر دانشگاه صنعتی شریف



در این شماره از
کدنامه، می‌خوانید:



اشکال‌زدایی برنامه‌ها



ویژه‌نامه‌ی پایان‌ترم

اشکال‌زدایی برنامه‌ها

با امکاناتی که برای دیباگ برنامه‌ها در اختیار شما قرار دارد، آشنا شوید

قطعا در بسیاری از اوقات (همانند انجام تمرین‌ها و یا پروژه)، شرایطی پیش آمده است که فکر می‌کنید برنامه‌ی خود را کامل نوشته‌اید، اما نمره‌ی کامل را نگرفته و سپس بعد از بررسی، اشکال آن را نیز پیدا نمی‌کنید. مهارت پیدا کردن سریع اشکال، در پیش‌برد سریع‌تر تمرین‌ها و پروژه‌ی درس و نیز افزایش سرعت عمل در حل بخش عملی آزمون پایان‌ترم درس، نقش فراوانی خواهد داشت. با کدنامه همراه باشید.

14 بهمن

تمرین اشاره‌گر

مهلت ارسال تمرین اشاره‌گر (که مباحث آن در پایان‌ترم نیز حائز اهمیت اند)

6 بهمن

پایان‌ترم

آزمون پایان‌ترم درس مبانی برنامه‌سازی (گروه ۱)، ساعت ۳ عصر

3 بهمن

پروژه

مهلت ارسال نهایی فاز نخست پروژه درس مبانی برنامه‌سازی

رویدادهای پیش

روی درس مبانی

برنامه‌سازی،

در صورت عدم تغییر

اشکال زدایی برنامه‌ها

سید پارسا نشایی

Watches را فعال کنید. در پنجره‌ی باز شده و ذیل آیتیم **Local Variables** می‌توانید متغیرهای برنامه خود را به همراه مقادیر آن‌ها در آن لحظه مشاهده کنید. اگر متغیر مدنظر تان را مشاهده نمی‌کنید، در سطر خالی پس از سایر متغیرها، نام متغیر خود را وارد کرده و سپس اینتر بزنید. امکان نوشتن کدهای کوچک (به عنوان مثال * **quantity price** که یک محاسبات ریاضی انجام می‌دهد) نیز در این مکان وجود دارد. اطلاع از مقادیر متغیرها و عبارات در هر لحظه، مانند این است که پنجره‌ای رو به **RAM** کامپیوتر برایتان باز شده باشد!

ابزارهای **Continue** و **Run to Cursor**

ممکن است پس از مشاهده‌ی مقادیر متغیرها تصمیم گرفته باشید به اجرای برنامه ادامه دهید؛ برای انجام این کار، از **toolbar** دیباگینگ بالای صفحه، گزینه‌ی **Continue** را انتخاب کنید. اگر مایل هستید که اجرای برنامه از نقطه توقف تا یک خط به‌خصوص ادامه یابد، نشانگر تایپ را روی خط مدنظر قرار داده و سپس گزینه‌ی **Run to Cursor** را بزنید.

ابزارهای **Step In** و **Step Out** و **Step Over**

این سه ابزار مشهور، امکانات بسیار مفیدی برای اشکال‌زدایی کد در اختیار شما قرار می‌دهند. پس از توقف روی یک خط به‌خصوص، به کمک **Step Over**، می‌توانید خط کنونی را اجرا و به خط بعدی بپردازید. این ابزار، شما را از هربار قراردادن نشانگر ماوس بر روی خط بعد و سپس زدن گزینه‌ی **Run to Cursor** بی‌نیاز می‌کند.

اگر در خط کنونی، یک تابع را صدا زده‌اید و می‌خواهید کد درون آن تابع را نیز اشکال‌زدایی کنید، با زدن گزینه‌ی **Step In**، اجرای برنامه از درون تابع صدا زده شده ادامه می‌یابد. هرگاه کار شما با اشکال‌زدایی تابع جدید تمام شد، با زدن گزینه‌ی **Step Out**، می‌توانید اجرای برنامه را به تابع اصلی بازگردانید.

ویژه‌نامه‌ی پایان‌ترم

امتحان پایان‌ترم درس مبانی برنامه‌سازی (گروه ۱ - استاد فکوری) در هفته‌ی آینده، برگزار خواهد شد. در این مقاله تلاش می‌کنیم مروری بر مهم‌ترین مباحث برای مطالعه قبل از آزمون، داشته باشیم.

تاکنون، برنامه‌های متعددی به زبان **C** نوشته‌اید، و قطعاً میان این تعداد برنامه، چندین برنامه وجود داشته که پس از ارسال پاسخ آن‌ها، نمره‌ی کامل را از داور خودکار کوئرا دریافت نکرده‌اید و مجبور شده‌اید در کد خود به دنبال اشکال کارتان بگردید، اما همه می‌دانیم که صرفاً با بررسی و نگاه کردن به کد نمی‌توان اشکال آن را یافت. احتمالاً هرگاه که رفع اشکال کد برای شما سخت شده است، از تعداد زیادی دستور **printf** استفاده کرده‌اید تا مقادیر متغیرهای مختلف را در هر لحظه چاپ کنید تا خطی که مشکل برنامه از آن ناشی می‌شود، پیدا شود؛ اما افزودن این تعداد دستور به برنامه، کاری بسیار دشوار و وقت‌گیر است. در این مطلب قصد داریم شما را با گوشه‌ای از امکاناتی که **Code::Blocks** برای اشکال‌زدایی یا **Debugging** بهتر کدتان در اختیار شما قرار می‌دهد، آشنا کنیم.

نکته‌ای مهم: برای در دسترس داشتن ابزارهای دیباگینگ، هنگام ساخت پروژه خود در **Code::Blocks** حتماً تیک مربوط به مورد **Create "Debug" Configuration** را فعال کرده و قبل از اجرای کد، از قسمت **Build target** در بالای صفحه، گزینه‌ی **Debug** را انتخاب کرده و سپس برنامه را با **mode** دیباگ اجرا کنید.

حال می‌خواهیم با برخی از ابزارهای مهم **Code::Blocks** برای کمک به دیباگ کد آشنا شویم:

قراردادن **Breakpoint**

ابزار **breakpoint** ابزاری برای متوقف کردن اجرای برنامه در یک نقطه مشخص است. یک **breakpoint** مشابه یک علامت توقف عمل می‌کند، به طوری که اجرای کد بلافاصله قبل از رسیدن به خطی که **breakpoint** روی آن قرار داده شده است، متوقف می‌شود. برای قراردادن **breakpoint** بر روی یک خط، کافی است بر روی فضای کوچک خالی سمت راست شماره خط کلیک کنید تا یک نقطه‌ی قرمز ظاهر شود. با زدن دوباره روی نقطه‌ی قرمز، **breakpoint** برداشته می‌شود. بعد از توقف برنامه روی یک **breakpoint** می‌توانید مقدار متغیرهای برنامه در آن لحظه را مشاهده کنید؛ برای این کار کافی است از منوی **Debug** و بخش **Debugging windows** گزینه‌ی

توجه: تمامی مطالب ارائه شده در این مقاله‌ی ویژه‌نامه، تنها برای دانش‌جویان گروه ۱ (استاد فکور) درس مبانی برنامه‌سازی دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف، صادق است.

آزمون آزمایشی

برای آشنایی هرچه بیش‌تر دانش‌جویان گروه یک با سوالات امتحانی، دو آزمون آزمایشی در کانال درس قرار داده شده است. توصیه می‌شود دانش‌جویان گرامی، ابتدا مطالعه و دوره‌ی کامل درس را انجام داده و سپس به حل آزمون‌های آزمایشی بپردازند. از دانش‌جویان انتظار می‌رود سوالات آزمون‌های آزمایشی و پاسخ آن‌ها را تحلیل کرده و برای پاسخ‌گویی به سوالات مشابه در آزمون اصلی، آماده باشند.

سایر منابع پیشنهادی برای مطالعه

مرجع رسمی درس، کتاب دیتل و دیتل است، بنابراین در صورت وجود زمان کافی، مطالعه‌ی سوالات پایانی هر فصل کتاب توصیه می‌شود (جهت یافتن پاسخ سوالات کتاب‌های مرجع دانشگاهی، می‌توانید به وبسایت‌هایی هم‌چون slader.com مراجعه کنید). سایت‌هایی چون GeeksForGeeks و هم‌چنین بانک سوالات مبانی برنامه‌نویسی و نیز سایر سوالات موجود در کوئرا، همگی برای تمرین بیش‌تر مناسب هستند. برای آمادگی هرچه بیش‌تر، می‌توانید سوالات سال‌های گذشته را از وبسایت cse.sharif.edu (بخش Courses) دریافت نمایید. از دانش‌جویان انتظار می‌رود بر مباحث پرسیده شده در تمرین‌های درس نیز تسلط لازم را داشته باشند. در صورتی که تمرینات را بدون تقلب انجام داده باشید، جای هیچ‌گونه نگرانی در خصوص عدم توانایی برای حل سوالات آزمون‌های درس، وجود نخواهد داشت.

توجه: نظر به اهمیت پایان‌ترم، توجه به نکات بیان‌شده در کلاس حل تمرین ویژه‌ی آزمون پایان‌ترم، شدیداً توصیه می‌شود. هم‌چنین، دانش‌جویان موظف هستند اطلاعاتی‌ی آزمون را به طور کامل، مطالعه کنند.

مباحث اصلی مطرح شده در آزمون‌های پایان‌ترم

بخش نظری

تحلیل کدهای مربوط به اشاره‌گرها

برخی از نکات قابل توجه در سوالات این مبحث:

- شناخت کامل انواع ترکیبی از اشاره‌گرها به متغیر و توابع (در همین راستا، مشاهده‌ی فیلم کلاس‌های حل تمرین اشاره‌گرها، شدیداً توصیه می‌شود)
- شناخت تفاوت‌های پاس دادن متغیر و اشاره‌گر به توابع

- کسب مهارت در تحلیل عبارت‌هایی که از چندین * و & پشت سر هم تشکیل شده اند

- شناخت کارکرد اسم آرایه به عنوان اشاره‌گر و ارتباط میان آرایه‌ها، اشاره‌گرها و رشته‌ها

توجه: حل سوال تحلیل کد اشاره‌گرها از تمرینات داده شده، قبل از آزمون، شدیداً توصیه می‌شود.

تحلیل کدهای مربوط به struct ها

برخی از نکات قابل توجه در سوالات این مبحث:

- شناخت نحوه کار با استراکت‌ها (در همین راستا، مشاهده‌ی فیلم کلاس حل تمرین استراکت، شدیداً توصیه می‌شود)
- شناخت انواع ترکیبی به عنوان زیرمجموعه‌های استراکت
- شناخت معنی انتساب یک استراکت به دیگری

تحلیل کدهای مربوط به توابع بازگشتی

در این گونه سوالات، یک قطعه کد حاوی یک تابع بازگشتی داده شده و خروجی کد خواسته می‌شود. در پاسخ‌گویی به این سوالات، دقت کامل، نوشتن مقدار تمامی متغیرها در هر لحظه و توجه به حالات پایه، بسیار کارگشاست. توصیه می‌شود نمونه‌های مشابه را در آزمون آزمایشی، تحلیل و مشاهده کنید. مشاهده‌ی فیلم کلاس حل تمرین مربوط به توابع بازگشتی نیز توصیه می‌شود.

بخش عملی

سوالات تابع بازگشتی

در غالب سوالات این بخش، از شما خواسته می‌شود تا یک برنامه‌ی عادی را بدون استفاده از حلقه و به شکل یک تابع بازگشتی، پیاده‌سازی کنید. **توجه کنید** که در سوالات تابع بازگشتی، عدم وجود بازگشت در تابع (صدا زده شدن تابع از درون خودش) و یا استفاده از هر کلمه‌ی مربوط به حلقه یا دستورات پرشی (مانند `while`، `for` و نظایر آن)، باعث کسر نمره‌ی شدید در سوال خواهد شد. به تیپ‌های متداول زیر، توجه کنید:

- محاسبه‌ی دنباله‌هایی که جملات آن‌ها به شکل بازگشتی داده شده است؛ مانند دنباله‌ی فیبوناچی
- محاسبه‌ی توابع و عملیات ریاضی، بدون استفاده از `math.h`: مانند محاسبه‌ی ضرب دو عدد به صورت بازگشتی و به کمک جمع

- جداسازی رشته بر اساس یک کاراکتر خاص؛ مانند کاراکتر کاما (ویرگول)

- در نظر گرفتن یک یا چند رشته به عنوان آرایه و پیمایش روی آن

در حل سوالات رشته‌ها، مجاز به استفاده از هر کتابخانه‌ی دل‌خواهی هستید، مگر آن‌که در صورت سوال، استفاده از یک کتابخانه به طور خاص، منع شده باشد. شناخت و اشراف کامل بر کاربرد تمام توابع رشته‌ای مشهور موجود در کتابخانه‌ی `string.h` نیز برای آزمون ضروری است.

امید است با توجه به نکات فوق، بتوانید در آزمون پایان ترم مبانی برنامه‌سازی گروه ۱، نمره‌ای عالی کسب کنید.



کدنامه در ترمی که گذشت!

نظر به این حقیقت که ترم کنونی، اولین نیم‌سالی است که ایده‌ی «کدنامه» توسط دستیاران آموزشی درس مبانی برنامه‌سازی، اجرا می‌شود، خوش‌حال و خرسندیم که این ایده، مورد استقبال دانشجویان عزیز واقع شده و انگیزه‌ی ما را برای خدمت هرچه بیش‌تر، دو چندان کرده است. امیدواریم با این‌که کدنامه، یک ایده‌ی جدید بوده و قطعاً اشکالات فراوانی در مقام اجرا نیز داشته، توانسته باشد برای دانش‌جویان علاقه‌مند، مفید واقع شود. در این لحظه، لازم است از تمامی عوامل تیم محتوای آموزشی که در این ترم دشوار مجازی، دست به دست هم، برای پیش‌برد آموزش بهتر به دانش‌جویان نوورود، تلاش کرده‌اند، قدردانی کنیم.

مسئول تیم محتوای آموزشی و کدنامه: سید پارسا نشایی

نویسندگان گرانقدر تیم محتوای آموزشی و کدنامه:

- سید پارسا نشایی
- محمدمهدی برقی
- علیرضا حسین‌خانی
- متین داغیانی
- علی حاتمی

- محاسبات استقرایی‌گونه؛ مانند شمردن طول یک رشته با افزودن ۱ به طول رشته به جز کاراکتر اول آن و یا تبدیل مبنا (می‌توانید با مشاهده‌ی فیلم جلسه‌ی حل تمرین ویژه‌ی میان‌ترم، با نمونه‌هایی از این سوالات آشنا شوید)

سوالات آرایه‌ها

بسیاری از سوالاتی که به کمک آرایه‌ها حل می‌شوند، راه حلی بدون استفاده از آرایه نیز دارند، اما در آزمون لازم است محدودیت‌های سوال را به طور کامل رعایت کنید و در غیر این صورت، با کسر نمره‌ی شدید مواجه خواهید شد. به تیپ‌های متداول زیر، توجه کنید:

- محاسبات ماتریسی و انجام عملیات متداول ماتریسی (مانند ضرب و وارون ماتریسی) روی یک یا چند ماتریس که از ورودی دریافت می‌شوند. در محاسبات ماتریسی، لازم است از آرایه‌های دوبعدی استفاده کنید.

- انجام عملیات گوناگون روی آرایه؛ مانند انجام چرخش روی ترتیب عناصر آرایه یا مرتب‌سازی آن به روش دل‌خواه

- پیدا کردن یک عدد (`search` کردن) عدد در آرایه

- دریافت آرایه به طول نامعلوم؛ در این گونه سوالات، حد بالایی برای طول ورودی آرایه داده شده و می‌توانید آرایه‌ی خود را به طول حد بالایی مشخص شده در نظر گرفته، ولی در حلقه‌هایی که روی آرایه حرکت می‌کنند، سائز آرایه را همان عدد ورودی در نظر بگیرید.

سوالات رشته‌ها

سوالات رشته، خود از نوع سوالات آرایه هستند؛ با این تفاوت که آرایه‌های استفاده شده، از نوع کاراکتری خواهند بود. برای تسلط بر حل سوالات رشته در آزمون پایان‌ترم، علاوه بر حل نمونه سوالاتی که در شماره‌ی ۵ کدنامه، مورخ ۲۶ آذر ۱۳۹۹ قرار داده شده، توصیه می‌شود تیپ‌های متداول زیر را نیز بررسی کنید:

- انجام عملیات روی رشته، مانند چرخش، معکوس کردن و یا مرتب‌سازی یک رشته با تشکیل جایگشتی حروف آن

- چک کردن خواصی از رشته‌ها، مانند پالیندروم (متقارن) بودن

- تغییر برخی از کاراکترهای یک رشته؛ مانند تغییر تمام اعداد یک رشته به یک کاراکتر خاص و یا تغییر رشته از حروف بزرگ انگلیسی به حروف کوچک و یا بالعکس

- توانایی نوشتن توابعی که مشابه کارکرد توابع رشته‌ای مشهور موجود در کتابخانه‌های داخلی را داشته باشند، بدون استفاده از کتابخانه‌ها