



دانشکده‌ی مهندسی کامپیوتر

مبانی برنامه‌سازی کامپیوتر امتحان عملی زبان C

سید صالح اعتمادی *

تاریخ امتحان: ۳۰ دی ۱۳۹۹
زمان امتحان: ۱۵۰ دقیقه

*تشکر ویژه از آقای علی حیدری که نسخه اولیه این قالب را در ترم دوم سال تحصیلی ۹۷-۹۸ برای درس برنامه‌سازی پیشرفته تهیه کردند.

۱ آماده سازی

آماده سازی برای این امتحان همانند تمرین C می باشد. با این تفاوت که کلیه کدهای شما در داخل پوشه `c:\git\FC99001\CEXam` و شاخه `main` خواهد بود. نیاز به ایجاد پایپلاین جدیدی برای امتحان نیست.

۲ فعال سازی تست

سوال های امتحان بصورت تعدادی تست طراحی شده اند که لازم است تابع لازم برای پاس شدن تست را پیاده سازی کنید. همه تست ها `comment` شده و بستر `Catch2` با استفاده از نشانه `hide` برای رد کردن و عدم اجرای تست تنظیم شده است. ابتدا کامنت های مربوط به تست ها را یکی-یکی برداشته و از شناخته شدن تست توسط افزونه `Catch2 and Google Test Explorer` و `Test Explorer UI` در `VSCode` اطمینان حاصل کنید. برای این منظور ابتدا کامنت های مربوط به یک تست را بردارید. سپس تست را مطالعه کنید. نام تابع مورد تست و پارامترهای ورودی و نوع مقدار برگشتی تابع مورد تست را تشخیص دهید. سپس در فایل `cexam.h` تابع را با پارامترهای ورودی و مقدار برگشتی مناسب تعریف کنید (در این مرحله نیاز به پیاده سازی نیست). در صورت نیاز می توانید فایل `cexam_main.cpp`

را ایجاد کرده و تابع `main` را در آن پیاده سازی کرده و از درستی تعریف تابع خود مستقل از تست اطمینان حاصل نمایید. نهایتاً به منظور شناخته شدن تست ها توسط `VSCode` لازم است فایل تست بیلد شود. همچنین لازم است تنظیم `catch2TestExplorer.executables` متعلق به `Catch2 and Google Test Explorer` به گونه ای تنظیم شده باشد که فایل های اجرایی حاصل از بیلد فایل تست را پیدا کند (`cexam_test.exe`). این فرایند را تا برآشته شدن کلیه کامنت های تست ادامه دهید.

۳ فرستادن تست در Azure DevOps

پس از شناخته شدن کلیه تست ها در `VSCode` کد خود را `add/commit/push` کنید.

۴ پیاده سازی توابع

۱.۴ تست `q1_sum`

تابع `q1_sum` دو عدد صحیح به عنوان پارامتر دریافت کرده و جمع آنها را برمی گرداند.

۲.۴ تست `q2_sum`

تابع `q1_sum` دو عدد صحیح و آدرس یک عدد صحیح به عنوان پارامتر دریافت کرده و دو عدد صحیح را در محل آدرس صحیح قرار می دهد.

۳.۴ تست `q3_solve_equation`

تابع `q3_solve_equation` ضرایب یک معادله درجه به شکل $ax^2 + bx + c = 0$ را به عنوان پارامتر دریافت می کند و علاوه بر برگرداندن تعداد ریشه های حقیقی معادله، مقادیر ریشه ها را در محل آدرس پارامترهای چهارم و پنجم قرار می دهد. فرمول ریشه معادل درجه ۲ به شکل زیر است:

$$\frac{b \pm \sqrt{b^2 - 4ac}}{2a}$$

برای جزئیات بیشتر تست را مطالعه کنید.

۴.۴ تست `q4_add_equations`

تابع `q4_add_equations` ضرایب دو معادله درجه دو مانند سوال قبل را دریافت کرده و در محل آدرس های داده شده قرار می دهد. برای جزئیات بیشتر تست را مطالعه کنید.

۵.۴ تست q5_digit_count

تابع `q5_digit_count` یک عدد صحیح مثبت یا منفی به عنوان پارامتر دریافت کرده و تعداد رقم‌های آن در مبنای ۱۰ را برمی‌گرداند.

۶.۴ تست q6_number_to_string

تابع `q6_number_to_string` یک عدد صحیح مثبت یا منفی از ورودی دریافت کرده و معادل حرفی آن را به صورت رشته حرفی در آرایه کاراکتری داده شده قرار می‌دهد. راهنمایی همانطور که مثلاً عدد متناظر با کارکتر A عدد ۶۵ می‌باشد، عدد متناظر با کاراکتر صفر انگلیسی ۰ عدد ۴۸ می‌باشد و سپس عددهای ۱ تا ۹ به ترتیب از ۴۹ تا ۵۷ می‌باشند.

۷.۴ تست q7_equation_to_string

تابع `q7_equation_to_string` ضرایب یک معادله درجه دو مانند سوال‌های قبل را به عنوان ورودی دریافت کرده و رشته حرفی متناظر با آن را در آرایه کارکتر داده شده قرار می‌دهد. این سوال تا حدی وقت گیر است. بهتر است آنرا برای آخر کار بگذارید.

۸.۴ تست q8_fill_array

تابع `q8_fill_array` یک آرایه اعداد صحیح، طول آرایه و یک عدد صحیح به عنوان پارامتر دریافت کرده و عدد صحیح را در تمام عناصر آرایه قرار می‌دهد.

۹.۴ تست q9_array_copy

تابع `q9_array_copy` دو آرایه هم اندازه و طول آنها را به عنوان پارامتر دریافت کرده و عناصر آرایه اول را در آرایه دوم کپی می‌کند.

۱۰.۴ تست q10_max_odd_numbers

تابع `q10_max_odd_numbers` یک آرایه اعداد صحیح و طول آن را به عنوان پارامتر دریافت کرده و از بین اعداد فرد بزرگترین آنها را برمی‌گرداند.

۱۱.۴ تست q10_max_odd_positions

تابع `q10_max_odd_positions` یک آرایه اعداد صحیح و طول آن را به عنوان پارامتر دریافت کرده و از بین اعداد در اندیس فرد بزرگترین آنها را برمی‌گرداند.

۱۲.۴ تست q12_reverse_odd_positions

تابع `q12_reverse_odd_positions` یک رشته حرفی به عنوان پارامتر دریافت کرده و عناصر آن که در اندیس‌های فرد هستند را برعکس می‌کنند. اندیس‌های زوج نباید تغییر کنند. برای جزئیات بیشتر به تست مراجعه کنید.

۱۳.۴ تست q13_matrix_compare

تابع `q13_matrix_compare` دو ماتریس هم اندازه به صورت jagged array (آرایه‌ای از آرایه‌ها) و ابعاد آن را به عنوان پارامتر دریافت کرده و در صورت تساوی آنها true برمی‌گرداند.

۱۴.۴ تست q14_matrix_add

تابع `q14_matrix_add` دو ماتریس مانند سوال قبل به عنوان پارامتر دریافت کرده و مقادیر ماتریس اول و دوم را به هم اضافه کرده و در یک ماتریس جدید برمی‌گرداند.

۱۵.۴ تست q15_create_complex_number

تابع `q15_create_complex_number` مقادیر موهومی و حقیقی یک عدد مختلط را به عنوان پارامتر دریافت کرده و یک `struct` با عناصر متناظر `i` و `r` درست کرده و برمی‌گرداند. برای پیاده‌سازی این تابع لازم است ابتدا نوع `struct` با نام `complex_number` را تعریف کنید.

۱۶.۴ تست q16_add_complex_number

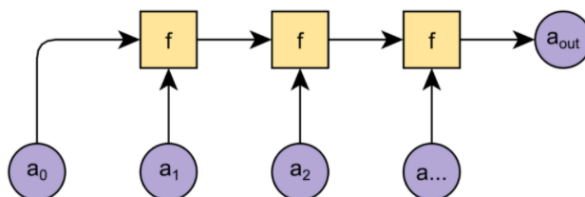
تابع `q16_add_complex_number` آدرس دو `struct` از نوع `complex_number` را به عنوان پارامتر دریافت کرده و جمع آنها را در پارامتر اول قرار می‌دهد. دقت کنید که محتوای `struct` دوم نباید تغییر کند.

۱۷.۴ تست q17_view_memory

تابع `get_cn_ptr1` و `get_cn_ptr2` را بگونه‌ای تعریف کنید که تست پاس شود. مطالعه و فهم تست جزو سوال است. در صورت نیاز با استفاده از دستور `-exec x/32bx <address>` می‌توانید حافظه را مشاهده کنید.

۱۸.۴ تست q18_aggregate

تابع `q18_aggregate` یک آرایه و یک اشاره‌گر به تابع به عنوان پارامتر دریافت کرده و عناصر آرایه را توسط تابع داده شده جمع‌بندی یا `aggregate` می‌کند. جمع‌بندی به این شکل است که عنصر اول و دوم آرایه به عنوان ورودی تابع داده شده و یک مقدار خروجی به دست می‌آید. حال این مقدار خروجی به همراه عنصر سوم آرایه به تابع داده می‌شود و مقدار خروجی جدید به دست می‌آید. این روند تا عنصر آخر آرایه ادامه پیدا می‌کند و خروجی نهایی برگردانده می‌شود. در شکل زیر عناصر آرایه a_0, a_1, \dots, a_n اشاره‌گر تابع `f` و خروجی تابع `a_out` می‌باشد.



شکل ۱

۵ ارسال

اگر موفق به پاس شدن تستی نشدید دستور مربوط به عدم اجرای تست را قبل از تست باقی بگذارید. پس از پیاده‌سازی توابع و پاس شدن تست‌هایی که فرصت کردین، نوبت به ارسال آنها می‌رسد. مثل قبل تغییرات را در شاخه `main add/commit/push` کنید.