

📄 مستندات پروژه نهایی: ماشین حساب مهندسی (C++ Procedural Project)

نام درس: مبانی برنامه‌نویسی

استاد: دکتر احسان خدنگی

موعد تحویل نهایی: ۱۴۰۴/۱۰/۲۸

۱. مقدمه و اهداف

هدف از این پروژه، پیاده‌سازی یک ماشین حساب مهندسی است که "منطق ریاضی" توابع را شبیه‌سازی می‌کند. در این پروژه دانشجوی باید بدون استفاده از ابزارهای آماده‌ی پیشرفته، الگوریتم‌های ریاضی، پردازش متن و مدیریت داده‌ها را پیاده‌سازی کند.

پروژه در ۴ فاز اجرایی توسعه داده می‌شود. پس از انجام هر فاز دانشجوی موظف است مستندات را به صورت مکتوب به دستیار آموزشی ارائه دهد و آمادگی هرگونه پرسش از کد های نوشته شده را داشته باشد.

پروژه تعریف شده شباهت قابل توجهی به سوالات امتحانی و کاربردهای برنامه نویسی در ریاضیات دارد به همین علت هر گونه کپی برداری از هوش مصنوعی غیر قابل پذیرش می‌باشد و در صورت شناسایی استفاده از هوش مصنوعی توسط دستیار آموزشی، دانشجوی ملزم به ارائه کامل مباحث و امتحان کتبی از بخش های مختلف پروژه به صورت جلسه حضوری می‌باشد.

در صورت بروز هر گونه اشکال در هر مبحثی مراجعه به مستندات سایت هایی مانند **w3schools**، کانال های یوتیوبی و کمک گرفتن از دوستان یا هوش مصنوعی صرفاً جهت ایده گرفتن بلا مانع است. برای دریافت راهنمایی های بیشتر نیز میتوانید به دستیاران آموزشی مراجعه کنید.

محیط توسعه و ارائه کد: برای توسعه میتوانید از محیط های استاندارد مانند Dev C++ یا VsCode استفاده کنید و فایل c++ را در کنار متن توضیحات(در قالب متن یا فایل pdf) برای دستیاران آموزشی ارسال کنید. در صورتی که به کامپیوتر دسترسی ندارید میتوانید از وبسایت onlinegdb.com استفاده کنید به طوری که هر بار برای توسعه فایل cpp را آپلود کرده پروژه را ادامه میدهید و در انتهای مرحله کد را دانلود میکنید.

۲. قوانین و محدودیت‌ها (Constraints)

۱. **ساختار برنامه:** استفاده از `class`، `struct` و `new/delete` (حافظه پویا) **ممنوع** است. کل پروژه باید با توابع (**Functions**) پیاده‌سازی شود.

۲. کتابخانه‌ها:

- استفاده از کتابخانه‌های `iostream` و `string` مجاز است.
- از کتابخانه `<cmath>` فقط و فقط برای توابع `pow` (توان)، `sqrt` (جذر) و `log` (لگاریتم) می‌توانید استفاده کنید.
- استفاده از توابع آماده مثلثاتی (`sin`، `cos`، ...) و عدد `M_PI` ممنوع است.

۳. **دقت محاسبات:** تمامی فرمول‌های بسط (Series) باید دقیقاً تا ۱۰ جمله محاسبه شوند.

۳. فازهای اجرایی پروژه

پروژه باید دارای یک منوی اصلی (switch-case) باشد که کاربر بتواند بین ۵ حالت زیر جابجا شود.

🚩 فاز ۱: هسته مرکزی و پردازش ورودی (Core Engine) – مهلت ارائه ۱۰/۱۵

این بخش وظیفه دارد ورودی‌های کاربر را تفسیر کرده و ثوابت ریاضی را تولید کند.

الف) تولید ثوابت ریاضی (Constants)

توابعی بنویسید که مقادیر زیر را با فرمول ریاضی محاسبه کنند:

۱. عدد نپر (e): مجموع ۱۰ جمله اول بسط زیر:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

۲. عدد پی (π): مجموع ۱۰ جمله اول بسط نیلاکانتا (Nilakantha):

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \dots$$

ب) سیستم ورودی هوشمند (Smart Parser)

شما باید تابعی به نام parse_input داشته باشید که یک رشته (string) بگیرد و خروجی double برگرداند.

الگوریتم:

- تشخیص کلمات کلیدی: اگر ورودی دقیقاً "pi" یا "e" بود، توابع بخش (الف) صدا زده شوند.
- اعتبارسنجی (Validation): اگر کلمه کلیدی نبود، کاراکترهای رشته چک شوند. فقط ارقام ۰ تا ۹، نقطه، و منفی – مجاز هستند. در صورت وجود کاراکتر غیرمجاز، خطا چاپ شود.
- تبدیل: اگر رشته معتبر بود، با دستور stod(string) به عدد تبدیل شود.

🚩 فاز ۲: مثلثات (Trigonometry) – مهلت ارائه ۱۰/۲۰

کاربر زاویه را بر حسب درجه وارد می‌کند.

الزامات:

- تبدیل واحد: ابتدا درجه را به رادیان تبدیل کنید ($rad = deg \times \frac{\pi}{180}$)
- فرمول‌ها: از بسط مک‌لورن برای sin و cos استفاده کنید (دقت ۱۰ جمله).
- تانژانت و کتانژانت: با تقسیم سینوس و کسینوس محاسبه شوند.
- مدیریت بی‌نهایت (Infinity):
 - اگر در محاسبه تانژانت، مخرج (کسینوس) بسیار نزدیک به صفر بود (مثلاً بین -۰.۰۰۰۱ و ۰.۰۰۰۱)، تقسیم انجام نشود و کلمه inf چاپ شود.
 - همین شرط برای کتانژانت (سینوس صفر) برقرار است.

🚩 فاز ۳: آمار و احتمالات (Statistics) – مهلت ارائه ۱۰/۲۴

در این بخش برنامه روی لیستی از اعداد کار می‌کند.

الزامات:

- آرایه: یک آرایه ثابت `double data[۱۰۰]` تعریف کنید. اگر کاربر بیش از ۱۰۰ عدد داشت، خطا دهید.
- مرتب‌سازی (Bubble Sort): قبل از محاسبه میانه، آرایه باید مرتب شود. الگوریتم مرتب‌سازی حبابی (دو حلقه تو در تو) را پیاده‌سازی کنید. استفاده از `std::sort` ممنوع است.
- محاسبات:

- مجموع (Sum)
- میانگین (Average)
- میانه (Median): اگر تعداد اعداد فرد است، عدد وسط؛ اگر زوج است، میانگین دو عدد وسط

🚩 فاز ۴: تحلیل اعداد (Number Analysis) – مهلت ارائه ۱۰/۲۸ (روز تحویل نهایی)

در این بخش کاربر یک عدد وارد می‌کند و برنامه باید ویژگی‌های آن را بررسی کند.

الزامات:

- بررسی صحیح بودن: چک کنید عدد اعشار نداشته باشد (مثلاً با کست کردن به `int` و مقایسه با عدد اصلی)
- زوج یا فرد (Even/Odd): با استفاده از عملگر باقیمانده `%`
- اول بودن (Prime): بررسی کنید عدد اول است یا خیر (حلقه از ۲ تا جذر عدد)

🌟 فاز ۵: چالش‌های امتیازی (Bonus Challenges) – مهلت ارائه ۱۱/۱

(انجام این بخش اختیاری است و نمره اضافه دارد)

چالش ۱: ضرب ماتریس‌ها (Matrix Multiplication)

امکانی فراهم کنید که کاربر درایه‌های دو ماتریس 3×3 را وارد کند و حاصل ضرب آن‌ها را ببیند.

- راهنمایی: نیاز به تعریف سه آرایه دو بعدی $[3][3]$ A , $[3][3]$ B , $[3][3]$ C دارید.
- الگوریتم: ضرب ماتریس نیازمند ۳ حلقه تو در تو است.

○ حلقه اول i (سطر ماتریس اول)

○ حلقه دوم j (ستون ماتریس دوم)

○ حلقه سوم k (برای ضرب و جمع کردن: $C[i][j] += A[i][k] * B[k][j]$)

چالش ۲: ریشه‌یابی عددی (N-th Root Finder)

تابعی بنویسید که بتواند ریشه n ام یک عدد را محاسبه کند (مثلاً ریشه سوم عدد ۲۷). استفاده از $\text{pow}(x, 1/n)$ ممنوع است.

راهنمایی الگوریتم (روش دو بخشی - Bisection):

باید از روش جستجوی بازه استفاده کنید. فرض کنید می‌خواهیم ریشه n ام عدد Target را پیدا کنیم:

۱. یک بازه حدس بزنید: $\text{Low} = 0$ و $\text{High} = \text{Target}$

۲. یک حلقه تکرار (مثلاً ۱۰۰ بار) ایجاد کنید.

۳. در هر تکرار:

○ وسط بازه را پیدا کنید: $\text{Mid} = (\text{Low} + \text{High}) / 2$

○ Mid را با یک حلقه ساده n ، for بار در خودش ضرب کنید تا Power بدست آید.

○ مقایسه:

▪ اگر Power از Target بیشتر بود، یعنی جواب در نیمه پایینی است ($\text{High} = \text{Mid} \leftarrow$)

▪ اگر Power از Target کمتر بود، یعنی جواب در نیمه بالایی است ($\text{Low} = \text{Mid} \leftarrow$)

۴. پس از پایان حلقه، Mid جواب تقریبی شماست.

 **بارم‌بندی (مجموع ۲۰ نمره)**

بخش پروژه	توضیحات	نمره
هوش مصنوعی ورودی	تابع parse_input، تشخیص e/pi و stod	۵
ریاضیات پایه	محاسبه دقیق فاکتوریل، e ، π و توابع مثلثاتی	۵
آمار و تحلیل	آرایه، مرتب‌سازی Bubble Sort، میانه و تشخیص عدد اول	۶
مدیریت خطا	جلوگیری از کرش برنامه و چاپ inf در مثلثات	۴
امتیاز ویژه	حل چالش‌های ماتریس و ریشه‌یابی	۴+