

ماشین حساب

ساختمان داده‌ها - دکتر رضا رمضانی

دانشگاه اصفهان

پاییز 1403

—

فهرست

4.....	هدف‌ها
4.....	قابلیت‌ها
4.....	پیش‌نیازها
5.....	توصیف نیازمندی‌ها
5.....	ساختمان داده‌ها
5.....	ویژگی‌ها و عملکردها
5.....	پیاده‌سازی
6.....	ساختار و مدل‌های داده
7.....	کدنویسی تمیز
7.....	مدیریت خطا
8.....	پیچیدگی و بهینه‌سازی
8.....	ورودی خروجی
9.....	امتیازی
11.....	نکات تکمیلی

هدف‌ها

هدف‌های اصلی

- شبیه‌سازی عملکرد ماشین حساب در محاسبه‌ی عبارت‌های ریاضی و حل معادلات
- آشنایی با کاربرد ساختمان‌داده‌های استک و صف در ذخیره‌سازی و پردازش داده‌ها
- آشنایی با الگوریتم‌های پردازشی مرتبط با محاسبات ریاضی
- و ..

هدف‌های جانبی

- رعایت اصول ساختارمندی کد
- رعایت اصول تمیزی کد
- و ..

قابلیت‌ها

- محاسبه‌ی عبارت‌های ریاضی شامل عملگرها و عملوندهای متداول
- محاسبه‌ی مجموعه معادلات دارای وابستگی
- رعایت اولویت‌بندی در محاسبات عبارت‌ها و معادلات
- تشخیص خطاهای متداول

پیش نیازها

- آشنایی با مبانی ساختمان‌داده‌های استک و صف
- آشنایی با مبانی الگوریتم و پیاده‌سازی
- آشنایی با عبارت‌های پایه‌ی ریاضی و اولویت عملگرها

توصیف نیازمندی‌ها

پروژه‌ی حاضر به منظور طراحی سامانه‌ای برای حل مجموعه‌ای از معادلات وابسته‌ی ریاضی تعریف شده است. این معادلات شامل عملگرها و عملوندهای مشخص و با فرمت تعیین شده هستند که حل آن‌ها نیازمند توجه به اولویت در ترتیب و حل معادلات است. این برنامه مجموعه‌ی معادلات را به صورت متنی دریافت و نتیجه‌ی محاسبات یا پیام خطای مرتبط را خروجی می‌دهد. جزئیات بیشتر از نحوه‌ی عملکرد و پیاده‌سازی در ادامه بیان خواهد شد.

— 100 امتیاز

ساختمان داده‌ها

- پشته (Stack)
- صف (Queue)
- آرایه (Array)
- پیاده‌سازی ساختمان داده‌های پشته و صف الزامی بوده و استفاده از پیاده‌سازی‌های آماده‌ی آن در زبان‌های برنامه‌نویسی موجب کسر نمره‌ی این بخش خواهد شد.

— 100 امتیاز

ویژگی‌ها و عملکردها

- پشتیبانی از عملگرهای جمع (+)، تفریق (-)، ضرب (*)، تقسیم (/)، توان (^) و فاکتوریل (!)
- پشتیبانی از عملوندهای صحیح و اعشاری (به فرم بخش اعشاری، بخش صحیح)، عدد پی ($PI=3.14159$)، عدد نپر ($EN=2.71828$) و متغیرهای تک‌حرفی (A-Z a-z بدون توجه به بزرگ یا کوچک بودن حروف)
- پشتیبانی از محاسبات اولویت‌دار (با ترتیب اولویت پرانتز، فاکتوریل، توان، ضرب و تقسیم، جمع و تفریق)
- پشتیبانی از مجموعه معادلات وابسته به شکل «عبارت ریاضی=متغیر»

— 300 امتیاز

پیاده‌سازی

در این پروژه سه بخش پیاده‌سازی الگوریتمی مدنظر است:

— 150 امتیاز

محاسبه‌ی عبارت‌های ریاضی اولویت‌دار

- یکی از روش‌های محاسبه‌ی عبارت‌های ریاضی اولویت‌دار، تبدیل آن‌ها به فرم postfix و سپس محاسبه به کمک پشته است.

- استفاده از تابع‌های آماده‌ی زبان برنامه‌نویسی صرفاً جهت اجرای عملیات‌های پایه‌ی ریاضی (مانند توان) مجاز می‌باشد.

— 100 امتیاز

مدیریت وابستگی معادلات

- یکی از روش‌های مدیریت وابستگی معادلات، استفاده از صف برای اطمینان از پردازش هر معادله تنها پس از پردازش وابستگی‌های آن است.

— 50 امتیاز

ذخیره‌سازی و دسترسی سریع به نتایج

- یکی از روش‌های سریع دسترسی به داده‌ها در کمترین مرتبه‌ی زمانی، استفاده از آرایه و اندیس‌گذاری است.

توصیه می‌شود روند پیاده‌سازی را به ترتیب گفته شده پیش ببرید:

- ابتدا بخش مربوط به محاسبه‌ی عبارت‌های ریاضی را پیاده‌سازی نموده و از صحت عملکرد آن اطمینان حاصل کنید (بخش اول تست‌کیس‌ها)؛
- سپس برنامه‌ی خود را برای پشتیبانی از چند معادله گسترش دهید. این مرحله را ابتدا با فرض مرتب بودن معادلات از نظر وابستگی پیاده‌سازی و ارزیابی نموده (بخش دوم تست‌کیس‌ها)؛
- در مرحله‌ی بعد به مدیریت وابستگی‌های درهم پردازید (بخش سوم تست‌کیس‌ها).
- در هر مرحله به ذخیره‌سازی و دسترسی سریع به نتایج نیز توجه کنید.

— 50 امتیاز

ساختار و مدل‌های داده

- رعایت اصول طراحی معماری و سازماندهی کد از جمله اصول SOLID برای توسعه‌پذیری و بهبود کیفیت طراحی ضروری است.
- در طراحی معماری این پروژه می‌توانید از نمونه‌ی پیشنهادی در قالب نمودار کلاس استفاده نمایید. توجه داشته باشید که این نمونه کاملاً پیشنهادی و ناقص است؛ به‌گونه‌ای که تکمیل جزئیات و اعمال تغییرات لازم در آن هنگام پیاده‌سازی ضروری است.

— 50 امتیاز

کدنویسی تمیز

رعایت اصول کدنویسی تمیز برای فهم‌پذیری و بهبود کیفیت کد ضروری است.

-- 50 امتیاز

مدیریت خطا

خطاهای ممکن در این سیستم در سه بخش کلی طبقه‌بندی می‌شوند:

خطاهای ورودی

- Invalid Input

مثال:

$F=@+T$
 $\#=9-10$

- Invalid Format

مثال:

$R=(4!-3($
 $3-S=81*2$
 $d=4-*3$

خطاهای مربوط به محاسبه‌ها

- Arithmetic Error

مثال:

$D=C/0$
 $W=(-4)^{(1/2)}$

خطاهای مربوط به معادله‌ها

- Inconsistency

مثال:

$X=3+5$
 $y=x+7$
 $x=96*2$

- Circular Dependency

مثال:

$$x=3+4-Y$$

$$Y=X/90$$

- Not Defined Variable

مثال:

$$Z=x-4$$

$$y=(4+Z)*5.4$$

-- 50 امتیاز

پیچیدگی و بهینه‌سازی

- پیچیدگی زمانی و فضایی الگوریتم خود را در بدترین حالت، به ازای N معادله با فرض وجود وابستگی در بین آن‌ها محاسبه و به صورت مختصر توضیح دهید.
- بررسی کنید که چگونه می‌توان با بهینه‌سازی الگوریتم این پیچیدگی را به مراتب کمتر کاهش داد.

-- 300 امتیاز

ورودی خروجی

ورودی

در خط اول ورودی عدد n به معنای تعداد معادلات می‌آید و هر یک از n خط بعدی شامل یک معادله به فرم (عبارت ریاضی=متغیر) می‌باشد که در قالب یک رشته دریافت می‌شود.

- عبارت ریاضی می‌تواند شامل عملگرهای جمع (+)، تفریق (-)، ضرب (*)، تقسیم (/)، توان (^) و فاکتوریل (!)، عملوندهای اعداد صحیح، اعشاری (بخش اعشاری.بخش صحیح)، عدد پی (PI=3.14159)، عدد نپر (EN=2.71828) و متغیرها (a-z A-Z) بدون توجه به بزرگی یا کوچکی و پرانتز گذاری برای تعیین اولویت محاسبات باشد.

$$0 \leq n \leq 10$$

$$10^{-9} \leq \text{any number in equations} \leq 10^9$$

خروجی

در صورت قابل محاسبه بودن مقدار همه‌ی متغیرها و عدم بروز خطا در حین محاسبه، نتیجه را تا چهار رقم اعشار (در صورت اعشاری بودن) و در قالب مجموعه‌ای از متغیرها (حروف بزرگ) و مقادیر منتسب به آن‌ها به ترتیب حروف الفبا در خروجی چاپ کنید. در صورت بروز هرگونه خطا در هر یک از معادلات، تنها نام خطا را در خروجی چاپ کنید.

مثال:

Inp 1

$$x=4*(5+PI)$$

Outp X=32.5663

Inp 3

$$B=A+1$$

$$C=3+7$$

$$A=C*4$$

Outp A=40

$$B=41$$

$$C=10$$

Inp 3

$$A=90-4$$

$$B=C^5$$

$$C=7/0$$

Outp Arithmetic Error

-- 250 امتیاز

امتیازی

-- 60

پشتیبانی از تعریف عملیات جدید

- افزودن نماد و عبارت ریاضی متناظر بر اساس عملیات‌های پایه‌ی ماشین حساب؛
 - می‌توانید از الگوی زیر برای تعریف عملیات جدید استفاده کنید.

مثال:

Inp Advanced Mode/New Operation

$$A \# B = A*B-(A+B)$$

Outp Operation defined successfully.

Inp 1

$$X=2\#3$$

Outp X=1

-- 50 امتیاز

پشتیبانی از نمایش تاریخچه محاسبات

- گسترش قابلیت‌های ماشین حساب با نمایش وضعیت عبارت در حال محاسبه در هر مرحله؛
مثال:

Inp Advanced Mode/Show History

2

$$C=54^2/D$$

$$D=87*(32-17)$$

Outp $D=87*15$

$$D=1305$$

$$C=2916/1305$$

$$C=2.2344$$

-- 40 امتیاز

پشتیبانی از توابع ریاضی

- گسترش قابلیت‌های ماشین حساب با پشتیبانی از توابع ریاضی مانند: توابع مثلثاتی،
لگاریتم و ...؛
مثال:

Inp Advanced Mode/Special Functions

2

$$X=50*\sin 90$$

$$Y=\tan 32/x$$

Outp $X=50$

$$Y=0.6246$$

-- 30 امتیاز

پشتیبانی از نمایش توابع

- گسترش قابلیت‌های ماشین حساب با افزودن امکان رسم نمودار توابع در کنسول یا
رابط کاربری گرافیکی؛

-- 20 امتیاز

پشتیبانی از رابط کاربری گرافیکی

- گسترش قابلیت‌های ماشین حساب با پیاده‌سازی رابط کاربری گرافیکی به سبک
دلخواه؛

Inp Advanced Mode/Graphical UI

-- <50 امتیاز

سایر موارد

- گسترش قابلیت‌های ماشین حساب بسته به خلاقیت و سلیقه‌ی شخصی؛
- به منظور حفظ عملکرد حالت پایه‌ی ماشین حساب، هریک از پیاده‌سازی‌های امتیازی باید با وارد کردن دستور خاصی مثلا Advanced Mode اجرا شوند. در غیر اینصورت ماشین حساب از ورودی خروجی‌های گفته شده در بخش‌های قبلی پشتیبانی می‌کند. پیروی از این الگو برای قبولی تست‌کیس‌ها ضروری است.
- پیاده‌سازی موارد امتیازی تنها در صورت تکمیل بخش‌های اصلی پروژه می‌تواند امتیاز اضافی به همراه داشته باشد.

نکات تکمیلی

زبان پیاده‌سازی

- زبان‌های برنامه‌نویسی قابل ارزیابی در پروژه شامل C++ و java و C# می‌باشد.
- استفاده از فریم‌ورک زبان‌های برنامه‌نویسی نام‌برده مجاز است.

مهلت اتمام

- پروژه‌ی جاری در تاریخ 1403/09/06 ساعت 06:00:00 منتشر و حداکثر مهلت اتمام آن تا تاریخ 1403/09/17 ساعت 06:00:00 می‌باشد.

گروه‌بندی

- پیاده‌سازی و ارزیابی پروژه به‌صورت فردی انجام می‌شود.

بستر پیاده‌سازی

- توسعه و پیاده‌سازی پروژه، در بستر گیت‌هاب انجام می‌گیرد.
- با ورود به **لینک اساینمنت پروژه**، ریپوزیتوری مخصوص هر فرد به‌صورت خودکار ساخته می‌شود.
- روند توسعه‌ی پروژه در قالب کامیت‌های **متوالی و معنادار** روی برنچی به نام Calculator و تحلیل پیچیدگی آن روی **برنچ دیگری** بنام Analysis انجام می‌شود. توصیه می‌شود پیاده‌سازی بخش‌های امتیازی نیز روی برنچ مجزا انجام گردد.
- ساخت **برنچ‌های متعدد** و رعایت **اصول نوشتار صحیح متن کامیت** برای توسعه‌ی تمیزتر توصیه می‌شود.

نحوه‌ی ارزیابی

- ارزیابی عملکرد پروژه، به صورت تست‌کیسی در بستر گیت‌هاب و همچنین ارائه‌ی حضوری انجام می‌گیرد.
- ارزیابی ابتدایی پروژه به کمک تست‌کیس‌های طبقه‌بندی شده و با ابزار تست گیت‌هاب انجام می‌شود.
- ارزیابی نهایی در قالب ارائه‌های 20 دقیقه‌ای، به صورت حضوری و براساس **بارم‌بندی ذکرشده در بخش‌های مختلف همین مستند** انجام می‌شود.
- بخش‌های امتیازی پروژه شامل ارزیابی اولیه **نمی‌شود**.