



شرح پروژه :

در این پروژه قصد داریم یک برنامه کاربردی برای یادگیری بهتر مفاهیم ساختمان داده ها طراحی کنیم. این برنامه کاربردی دارای قسمت های زیر است :

- تبدیل فرم infix به postfix و بالعکس
- تبدیل فرم infix به prefix و بالعکس
- تبدیل فرم prefix به postfix و بالعکس

برنامه به این صورت اجرا می شود :

مرحله ۱: ابتدا یک منو برای کاربر نمایش داده می شود که دارای گزینه های زیر است :

۱. تبدیل infix به postfix
۲. تبدیل postfix به infix
۳. تبدیل infix به prefix
۴. تبدیل prefix به infix
۵. تبدیل postfix به prefix
۶. تبدیل prefix به postfix
۷. نمایش تاریخچه

مرحله ۲: بعد از انتخاب عدد مورد نظر هر بخش به صورت زیر ادامه داده می شود :

ورودی ۱ : برای تبدیل عبارت infix به postfix از الگوریتم حیاط راه آهن دیکسترا استفاده کنید. در این بخش ابتدا یک رشته دریافت می شود که هر عملوند یا عملگر با استفاده از یک اسپیس از هم جدا می شوند. بعد از دریافت رشته ورودی، باید خروجی در هر مرحله نمایش داده شود. منظور از خروجی نمایش خروجی به فرم RPN و پشته ی عملگرها می باشد. اگر رشته ورودی به فرم infix نبود باید با یک ارور مناسب به کاربر اطلاع داده شود و از کاربر خواسته شود تا دوباره ورودی را وارد کند و یا به منو بازگردد.

ورودی ۲، ۳، ۴، ۵، ۶ : در این بخش ها ورودی مانند بخش ۱ از کاربر دریافت شده و در صورت معتبر بودن ورودی، خروجی و پشته عملگرها را به کاربر نمایش می دهد و در غیر این صورت، با یک ارور مناسب، از کاربر ورودی جدید می خواهد و یا آن را به منو ارجاع می دهد.

ورودی ۷: در این بخش تاریخچه عملیات های انجام شده را نمایش می دهیم. بعد از انتخاب این بخش از کاربر خواسته می شود یکی از اعداد زیر را انتخاب کند :

۱. خروجی براساس مرتبه ی انجام شده هر عملیات به صورت صعودی باشد.

❖ توضیح: فرض کنید از ابتدای اجرای برنامه تا کنون قسمت های زیر به ترتیب انجام شده اند :

Infix to postfix

Postfix to prefix

Prefix to infix

Postfix to prefix

Postfix to prefix

Postfix to prefix

Prefix to postfix

Prefix to infix

Infix to prefix

خروجی تاریخچه بالا به صورت زیر است :

Postfix to prefix : 5

Prefix to infix : 2

Infix to postfix : 1

Infix to prefix : 1

! دقت کنید در رکورد های با مرتبه مساوی ، اولویت با عملیاتی است که زودتر انجام شده است.

۲. خروجی براساس مرتبه ی انجام شده هر عملیات به صورت نزولی باشد.

! این قسمت ، دقیقاً برعکس قسمت ۱ می باشد .

لینک مفید :

https://en.wikipedia.org/wiki/Shunting-yard_algorithm

بخش نمره اضافه :

★ می توانید الگوریتم های مرتب سازی که تاکنون آموخته اید را به منوی اصلی برنامه اضافه کنید.

★ پیاده سازی ال نمره اضافه دارد.

★ هرچقدر پیاده سازی بخش های مختلف پروژه ، بهینه تر باشد نمره اضافه بیشتری دریافت خواهید کرد.

نکات پروژه :

- ❖ رعایت اصول شیء گرایی لازم می باشد.
- ❖ می توانید پروژه را با زبان های برنامه نویسی C++ ، Python و یا Java پیاده سازی کنید.
- ❖ برای قسمت هایی که در درس ساختمان داده ها تدریس شده است (صف ، پشته ، مرتب سازی و...) استفاده از کتابخانه ها و توابع آماده مجاز نیست و باید خودتان آن ها را پیاده سازی کنید.
- ❖ پروژه به صورت تک نفره یا نهایت گروه های دو نفره قابل انجام است. (افرادی که به صورت تک نفره پروژه را انجام دهند در نهایت نمره شان با احتساب ضریب ۱۲۰٪ حساب خواهد شد.)
- ❖ در صورت مشاهده تقلب و اثبات آن، نمره ۱۰۰- برای گروه ها منظور خواهد شد.
- ❖ فقط یک فایل zip با نام DS-Midproj-StudentNumber.zip را در سامانه آموزش مجازی (VU) آپلود نمایید.
- ❖ هر دو عضو گروه باید پروژه را ارسال کنند.
- ❖ هنگام تحویل پروژه، هر دو عضو گروه باید تسلط کامل به تمام پروژه را داشته باشند.
- ❖ تاریخ تحویل پروژه تا ساعت ۲۳:۵۵ تاریخ ۱۴۰۰/۹/۳۰ می باشد .

با آرزوی موفقیت

تیم حل تمرین ساختمان داده ها