



جمهوری اسلامی ایران
وزارت علوم و تحقیقات و فناوری
دانشگاه فنی و حرفه ای
دانشگاه فنی الزهرا (س) مشهد
تمارین درس مهندسی نرم افزار
آموزشکده فنی و حرفه ای دختران مشهد-الزهرا(س)

گروه : کامپیوتر – نرم افزار

مقطع : کارشناسی

استاد راهنما :

سرکار خانم مهناز زارع

نام دانشجو:

صباجانی

شماره دانشجویی :

۰۳۱۲۱۰۵۸۷۰۵۰۳۷

آبان ماه سال ۱۴۰۴

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فهرست

تمرین اول ۱

فهرست شکل‌ها

عکس 1-1 1-1

تمرین اول: الگوریتم بنویسید که یک لیست از رشته را گرفته و بر اساس طول رشته مرتب کند اگر هر رشته هم طول باشد، بر اساس حروف الفبا مرتب کنید.

۱. شروع الگوریتم
۲. یک لیست خالی تحت عنوان `S_list` ایجاد کنید.
۳. دریافت رشته از کاربر
- ۳-۱. هر رشته وارد شده به لیست اضافه می شود .
- ۳-۲. ورود رشته ها تا زمانی ادامه می یابد که کاربر اعلام کند (مثلاً وارد کردن رشته خالی یا علامت پایان).
۴. حذف رشته های تکراری: پس از ساخت لیست، تمام رشته های تکراری از لیست حذف می شوند تا هر رشته فقط یکبار وجود داشته باشد.
۵. نمایش لیست اولیه
۶. حلقه ای ایجاد میکنیم و متغیر ایجاد شده درون حلقه را `i` میگذاریم، برای هر عنصر `i` از لیست، رشته ای را برای مقایسه با سایر رشته ها انتخاب می کنیم.
۷. حلقه درونی ایجاد میکنیم و متغیر داخل آن را `j` قرار می دهیم، برای هر عنصر `j` بعد از `i`، مراحل زیر انجام می شود:
 - ۷-۱. مقایسه طول رشته ها اگر طول رشته `i` از `S_list` بزرگ تر از طول رشته `j` ام `S_list` باشد، جای دو رشته تبادیل می شود.
 - ۷-۲. مقایسه حروف الفبا اگر دو رشته طول برابر داشته باشند، براساس حروف الفبا مقایسه و در صورت نیاز جابه جا می شوند.
۸. تکرار مقایسه ها حلقه ها تا پایان لیست ادامه می یابند تا هیچ جابه جایی لازم نباشد و لیست کاملاً مرتب شود.
۹. لیست نهایی که بر اساس طول و در صورت تساوی طول بر اساس حروف الفبا مرتب شده است، نمایش داده می شود.
۱۰. پایان الگوریتم

شبه کد یا (Pseudo Code) این الگوریتم از دیدگاه بنده

شروع

گام ۱: دریافت لیست به صورت پویا و حذف تکراری لحظه‌ای لیست $\leftarrow S\text{-list} = []$ در حالی که کاربر رشته وارد می‌کند: اگر رشته وارد شده خالی است، توقف.

گام ۲: بررسی تکراری اگر رشته در لیست وجود ندارد آنگاه اضافه کردن رشته به لیست و گرنه نمایش پیغام "این رشته قبلاً وارد شده است"

گام ۳: نمایش لیست اولیه: نمایش لیست

گام ۴: مرتب‌سازی برای i از ۱ تا طول(لیست) - ۱ انجام بده :

برای j از $i + 1$ تا طول(لیست) انجام بده :

اگر طول (لیست $[i]$) $<$ طول (لیست $[j]$) آنگاه

جابه‌جا کردن لیست $S\text{-list}[i]$ و لیست $S\text{-list}[j]$

اگر طول برابر بود، ترتیب الفبا را بررسی کن

اگر طول (لیست $[i]$) $==$ طول (لیست $[j]$) آنگاه اگر لیست $S\text{-list}[i]$ بعد

لیست $S\text{-list}[j]$ در الفبا بیاید آنگاه جابه‌جا کردن لیست $S\text{-list}[i]$ و لیست $S\text{-list}[j]$ انجام میشود.

گام ۵: لیست نهایی که بر اساس طول و در صورت تساوی طول بر اساس حروف الفبا مرتب شده است، نمایش داده می‌شود.

پایان الگوریتم

علت اینکه از شبه کد استفاده کردم برای این بود که من از الگوریتم برای استفاده کردم که مغزم وادار بشه مسئله رو کامل بفهمم که چه کاری باید انجام بدم از شبه کد برای این استفاده کردم که الگوریتمم به کد نویسی واقعی نزدیک تر باشه تا موقع کد نویسی ذهنم راحت تر باشه.