

پرنیان شاکریان

سیستم‌های عامل

گزارشکار تمرین اول

شماره دانشجویی ۹۹۴۰۰۰۶۴

در ابتدا کتابخانه مورد نظر را که `studio.h` می باشد را وارد می کنیم.

در ابتدای کار مقدار `count` به گونه ای که تعداد دستورهای است که از `user` دریافت خواهد کرد. سپس وارد `for` می شود.

که به ازای هر دستوری خود متن دستور را خواهد گرفت. اگر حرف اول `a` بود یعنی `add` می باشد پس

شرح به انجام عملیات `add` می کند و اگر `c` بود یعنی `end` را باید تمام کند و اگر `f` بود یعنی `for` است.

حالا نحوه کارکرد می بینیم. اگر `for` بود وارد تابع `loop` می شود و در تابع `loop` می آید یکبار `scan` می کنند تا مقدار `r`

تعداد `loop` که باید بخورد را بگیرد. سپس مقدار اول `sumset` را برابر مقدار `sumset` می آید مقدار `for` گرفته شده را در

`sumset` ضرب می کنند. هر دفعه که `for` می آید مقدار اول را برابر با `sumset` می آید مقدار `for` گرفته شده را در

بار قبلی 2×2 می شود و ۶ بار باید می خوانیم تا تمام دهیم به این ترتیب و ضرب می شود به این روش `sumset` حساب می شود.

مقدار `loopcount` برای این است که بهنیم به تعداد `loop` داشتیم. آنگاه `sets` برای این است که در آن مقدار

هر `for` مشخص می کنیم که چند مقدار است (یعنی مقدار `for` ها را داشته باشیم).

`loopcount * sets` را برابر به قرار دهیم که مقدار `loop` به تو `Array` که `loop` ها را در آن ذخیره کردیم.

در `end` وقتی به تابع رفته که باید `loopcount` یک عدد کم کرده. `sumset` را تقسیم بر مقدار آخرین `loop` که

داشتیم می کند مقدار آخرین `loop` را برابر قرار می ده تا نشان دهد مقدار `loop` ها کم شده و در نهایت در قسمت

`add` می آید که `add` بخورد به اندازه `sumset` به `result` ما اضافه می کنند.

در ابتدا کتابخانه های مورد نظر را include می کنیم. کتابخانه ی ما stdio.h می باشد. اگر مقدار روزهای Good را n در نظر بگیریم روزی شامل خط می باشد که خط اول n و خط دوم از آن به طول n که price را در تعداد روزهای n تا n در نظر خواهد گرفت. در خط اول `good finder int` در یک اعلان برای دادن یک نوع عدد صحیح به یک متغیر می توان استفاده کرد. در این الگوریتم ابتدا مقادیر `count` و `days` را از ورودی می خوانیم (در بالا `days` مقدار برقیع داده شده n مقدار می باشد) سپس در متغیرها می ریزیم.

سپس تابع `good finder` را روی آن `call` می کنیم که برای آن تابع برای هر روزی نگاه می کند که آیا روزهای بعد قیمت بک کمتر بوده و یا بیشتر بوده است. سپس بعد از چک کردن روزهایی که دارای این ویژگی هستند برای نوی `good finder` نفع خواهد کرد و سپس `return` خواهد کرد و همان را مستقیم چاپ خواهد کرد. در `for` اول ابتدا قبل آن یک `scanf` زده تا مقدار روزها را `count` کند و سپس در `for` وقتی انجام شد `++` می کند.

در `for` آخر برای حساب روزهای خوب `good days` و یک `for` تر در تو زده که در واقع می گوید اگر روز جدید از روز قبلی بهتر بود پس `++` را خارج کرده و بعد از ناب `break` می کند

`days[i] < days[j]`

در کد خود ابتدا کلمه مورد نظر خود را که `stio.h` است وارد می کنیم.

در ابتدا یک بقیع در مورد متغیرهای اولیه کد بگیریم. `lines` برای تعداد خط هایی است که می خوانیم بتوانیم. `names` آرایه

کارتری است که قرار است اسم هادر آن ذخیره شده ما هادی که گنیم. هادی اول کد اسم هادر. هادی دوم کارتری هادی

اسم هادی باشد. `Point` مقدار امتیاز هر کدام از این اسم هادی باشد. `index` در اصل برای ذخیره کردن `index` فرد برنده

است. و در `count` نیز تعداد افراد شرکت کننده را ذخیره می کنیم. `winner score` امتیاز فرد برنده است.

با شروع برنامه ابتدا تعداد خط هادر را می خواند. در شروع اول اسم هادی را می خواند و سپس به ازای تمام اسمی که در آرایه `names`

داریم در `string equal function` را یک می آید که آیا چنین اسم قبلاً بوده یا خیر. اگر بوده که به امتیاز آن اضافه می کند

و سپس چک می کند که اگر امتیاز آن الان بزرگتر از امتیاز `winner score` قبلی باشد قدر نام عنوان `winner` جدید می دهیم

خراشید کرد. یعنی هم `winner score` و هم `index` را به فرد جدید تغییر می دهد. این در حالی بود که اسم را قبلاً داشتیم

اگر ندانیم `for` را جلبرده بعد از اینکه به نتیجه برسد. دارد تابع `copy array` می شود و اسم را به آرایه

افزوده می کند و امتیاز آن را اضافه می کند و دوباره چک می کند که اگر امتیاز فرد جدید بیشتر از `winner score` بود آن را به

جدید تغییر می دهد و به `index` آن نیز تغییر می دهد. در آخر `count` را ++ می کند و تعداد نفرات را جمع می کند

در نهایت `Print` اسم فرد برنده را می نویسد. در آخر تابع `string equal` کارتری به کارتری `string` ها را چک می کند که

آیا با هم برابر و `copy array` نیز در آرایه کارتری را گرفته و یکی را در دیگری کپی می کند

در اینجا کتابخانه `std::vector` را داریم که می‌توانیم.

در تابع `main` ما طول رشته‌ی مورد نظر را `P` و `Q` در عدد داده شده هستند. در اینجا بخش دیگری عدد و دلی باید

به `P` و `Q` اضافه شود.

اگر بخش دیگری در دست نباشد مجموع دو عدد داده شده خود را در تابع `main` و وقتی عدد `P` بخش دیگری نباشد یعنی

از هر دو عدد فوق `P` و `Q` را در تابع `main` استفاده شده `P` و `Q` را در `P Count` و `Q Count`

نکته داشته‌ایم اگر به نقطه رسیدیم که $n < 0$ - مجموع `P Count` و `Q Count` را ۱ - می‌کنیم. سپس `return` می‌کنیم

$n = -(P + Q)$ - خود خواهیم داشت $n < 0$ then P و Q غیر قابل تقسیم n if

اگر هیچ یک از شرایط فوق برقرار نباشد تابع `splitting` با مقدار `P` و `Q` (در همان حوزان) که به n قابل تقسیم است `return` می‌کنیم

به تعداد `Q Count` و `P Count` `break` و `print` کرده.

هر دو شکستن به تعداد `P` و `Q` `char` را خوانده و `print` می‌کنیم.