

کد سوئیسی (main args)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

بیاین برگردیم به تقریباً صدسال پیش، اون موقع که هیتلر و چرچیل و استالین داشتن دورهم میجنگیدن(!)، که الان به اسم جنگ جهانی دوم ازش یاد میکنیم!

اون موقع ارتش سوئیس، یه چاقو ی باحال داشتن که بهش میگفتن Offiziersmesser (تونستی بخونی اسمشو؟) تلفظ این کلمه خیلی سخت بود برای همین آمریکایی ها یه اسم خیلی ساده روش گذاشتن، بهش میگفتن "Swiss Army Knife". خدا خیرشون بده:)).

"چاقو ارتش سوئیس" یک چاقو جیبی همه کاره اس که یه عالمه وسیله های مختلف مثل تیغ، اهر چوب، قیچی، سیم بر، قوطی باز کن و ... تو خودش جا داده.



حالا بیاین برگردیم به زمان حال، که جنگ جهانی سوم شده و اصغر، سرباز شجاع و فداکار، میخواد از کشورش محافظت کنه و برای این کار به کمک ما نیاز داره. اون باید یه برنامه ای بنویسه که مثل چاقو ی همه کاره برای رشته ها (Strings) عمل کنه.

یعنی باید یه سری رشته رو از کاربر بگیره و با توجه به اینکه کاربر چه آپشنی رو انتخاب کرده یه سری کارا رو براش بکنه.

آپشن هایی که چاقو ی ما داره چیا هستن؟

۱. Palindrome (-p, --palindrome) :

ما به رشته ای میگیم palindrome که قرینه باشه، یعنی از هر طرفی بخونیش همون بشه!

مثلا رشته های "madam" و "racecar" و "bcb" پالیندروم هستند ولی رشته هایی مثل "milk" و "a bh hba" پالیندروم نیستند.

۲. Find (-f, --find):

دو تا رشته بگیره ، بینه رشته اول تو دومی هست یا نه؟ (توضیح بیشتر در مثال های پایین هست)

۳. Reverse (-r, --reverse):

یه رشته بگیره و برعکسش کنه . مثلا وقتی رشته "reverse123" رو میگیره ، رشته "321esrever" رو خروجی بده.

۴. Sort (-s, --sort) :

چنتا رشته بگیره و اونا رو برحسب الفبا مرتب کنه. مثلا وقتی رشته های "amain" و "fabric" و "bill" و "amnesia" رو میگیره باید این ترتیب رو خروجی بده: amain , amnesia , bill , fabric

۵. Concatenate (-c, --concatenate):

تمام رشته های ورودی را به هم می چسباند و یک رشته واحد درست میکند.

۶. Help (-h, --help):

اگر به برنامه این دستور داده شود برنامه تمام آپشن هایی (palindrome, find, ...) که دارد را برای کاربر چاپ میکند.

ورودی و خروجی

کاربر یکی از آپشن های بالا را قبل از اجرای برنامه انتخاب کرده و رشته های خود را به صورت زیر به برنامه میدهد. دقت کنید که آپشن هایی که کد ما داره همون موقع اجرای برنامه بهش داده میشن (نه در زمان اجرا !)

دقت کنید که هر کدام از آپشن ها بجز find میتوانند تعداد مختلفی رشته (string) ورودی داشته باشند.

مثال از help:

ورودی:

```
>>./code.out -h  
OR  
>>./code.out --help
```

خروجی:

```
-h, --help      prints all commands in terminal.  
-p, --palindrome checks if input string is palindrome.  
-c, --concatenate concatenates all the input strings.  
-s, --sort      sorts all of the input strings in alphabetic order.  
-f, --find      find a substring in a string.  
-r, --reverse    reverse all the input strings.
```

مثال از palindrome:

ورودی:

```
>>./code.out -p "aba" "madam" "car"  
OR  
>>./code.out --palindrome "aba" "madam" "car"
```

خروجی:

```
"aba" is palindrome.  
"madam" is palindrome.  
"car" is not palindrome.
```

مثال از sort:

ورودی:

```
>>./code.out -s "aba" "madam" "car"  
OR  
>>./code.out --sort "aba" "madam" "car"
```

خروجی:

```
aba  
car  
madam
```

مثال از Reverse:

ورودی:

```
>>./code.out -r "quera" "racecar" "car"  
OR  
>>./code.out --reverse "quera" "racecar" "car"
```

خروجی:

```
areuq  
racecar  
rac
```

مثال از find:

ورودی:

```
>>./code.out -f "happy" "I am happy"  
OR
```

```
>>./code.out --find "happy" "I am happy"
```

خروجی:

```
"happy" was found in "I am happy"
```

مثال از concatenate:

ورودی:

```
>>./code.out -c "I " "like " "cats."
```

OR

```
>>./code.out --concatenate "I " "like " "cats."
```

خروجی:

```
I like cats.
```

HANGMAN (functional programming)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

این لینک ، روند بازی رو کامل توضیح داده :

<https://www.youtube.com/watch?v=leW9ZotUVYo>

خب همونطور که دیدین ، اول یک نفر یک کلمه ای رو انتخاب میکنه ، بعد ما تعداد حروف این کلمه رو به بقیه نفرات نشون میدیم ، و اونا باید حدس بزنن چه حروفی در این کلمه وجود داره ، اما با هر حدس غلط ، "من" بازی ما یه قدم به "هنگ" نزدیک تر میشه (:))

اول بازی ، یک کلمه دلخواه رو به صورت main argument از کاربر میگیریم و در هر مرحله باید محتوایی که در فیلم یوتیوب ، روی تخته کشیده میشد رو نشون بدیم:

۱. چقدر از کلمه درست پر شده ،

۲. حدس های اشتباه چیا بودن و

۳. چقدر از آدم ما تا الان کشیده شده.

و در پایان بازی هم نشون میدیم که فرد برنده شده یا نه.

- حتما سعی کنین طراحی کدتون به صورت functional باشه.
- حواستون باشه که ، حروف بزرگ و کوچک هیچ فرقی باهم ندارن و اونارو باید یکی فرض کنین.

بخش امتیازی :

- میتونین یه گزینه داشته باشین که در هر مرحله از بازی ، کاربر بتونه کلمه رو به صورت یکجا وارد کنه و اگه درست حدس زده بود برنده بشه.

Hangman is a good game, It teaches you that by sayin the wrong words you could end someone's life.

دیبگ!

در این سوال یک مجموعه کد که از طریق [این لینک](#) قابل دانلود میباشد وجود دارد که شما باید ابتدا کد را با کمک دیباگ تصحیح و در نهایت خروجی آن ها را در فایل pdf قرار دهید در فایل پاسخ نهایی مشکلات هر تکه کد نیز باید مشخص شود. دقت شود که بعضی از کد ها صرفا جنبه ی آموزشی دارد و ممکن است هیچ غلطی در آن ها وجود نداشته باشد.

یکی از موارد زیر را به دلخواه انتخاب کنید و انجام دهید. (دیگر موارد نمره ی امتیازی دارد):

- نوشتن unit test برای تعدادی کد دلخواه با استفاده از توضیحات داده شده و کتابخانه های آماده مانند xUnit
- نوشتن assertion برای تعدادی کد دلخواه
- علاوه بر دیباگ کردن عادی دیباگ کردن به کمک پرینت کردن در فایل با فرمت توضیح داده شده در کلاس

گیت باگ (: git)

مهمی جدیداً رزومه ش رو برای کار توی یک شرکت فرستاده، سرپرست منابع انسانی شرکت با توجه به رزومه اون قبول کرده که یک هفته مهمتی براشون به صورت آزمایشی کار کنه تا بتونه متوجه بشه آیا مهمتی مهارت های لازم برای بودن تو کار تیمی و کار شرکتی رو داره یا نه.

برنامه نویسی ارشد شرکت برای بررسی کردن مهارت های مهمتی راجب گیت و گیت هاب براش چالش هایی گذاشته:

چالش اول:

باید که کد های یک مخزن توی گیت هاب رو بررسی و دیباگ کنه و کد اصلاح شده خودش رو با استفاده از مهارت هاش توی گیت برای صاحب مخزن ارسال کنه. [لینک ریپازیتوری](#)

چالش دوم:

باید که هر فرد با توجه به این فایل

قسمت مربوط به خودش رو ترجمه کنه و برای صاحب مخزن با استفاده از مهارت هاش ارسال کنه. فایلی که قراره این تغییرات رو روش انجام بدین از لینک زیر قابل دسترس هست. این فایل درواقع فایل مربوط به قسمت هشت کتاب کد تمیز هست.

[لینک فایل](#)

[لینک ریپازیتوری](#)

نکات قابل توجه:

- کد هایی که قراره دیباگ کنید درواقع کد های تمرین **دیباگ!** هفته پیش هستش و هدف استفاده از مهارت هاتون توی گیت و گیت هاب هست.
- برای چالش اول حتما طبق سوال جلسه قبل یکی از نکاتش رو هنگام دیباگ کردن در نظر بگیرید و استفاده کنید. (دیباگ با `assertion` یا `debug_print` و یا نوشتن `unit_test`)

- ریپو ها رو برای خودتون فورک کنید و اونجا کامیت هاتون رو انجام بدین و درنهایت pull request بفرسین و بعد از انجام این کار ها، نام و نام خونوادگی تون به همراه نام کاربری گیت هاب تون رو توی یک pdf بزارید و آپلود کنید.