Scrabble

Problem set

1 مسئله‌ای برای حل کردن

در بازی scrabble، بازیکنان با ساختن کلمات امتیاز می‌گیرند که این امتیازات با جمع کردن ارزش هر حرف به دست می‌آیند.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	3	3	2	1	4	2	4	1	8	5	1	3	1	1	3	10	1	1	1	1	4	4	8	4	10

برای مثال، اگر ما بخواهیم به کلمه “CODE” امتیاز دهیم، با در نظر گرفتن اینکه ارزش C سه امتیاز، ارزش O یک امتیاز، ارزش D دو امتیاز و ارزش E یک امتیاز است، امتیاز "CODE" را 7 اعلام میکنیم.

در فایلی به اسم `scrabble.c` در فولدری به اسم `scrabble`، برنامه‌ای در C بنویسید که برنده بازی Scrabble را اعلام کند. برنامه شما باید دوبار درخواست ورودی کند: یکبار از بازیکن اول (Player 1) کلمه‌اش را میگیرد و یکبار از بازیکن دوم (Player 2)؛ سپس با توجه به اینکه کلمه کدام بازیکن امتیاز بیشتری گرفته است، یا باید Player 1 wins! را چاپ کند یا Player 2 wins! را و در صورت برابری Tie! را چاپ نماید.

```
#include <ctype.h>
#include <cs50.h>

#include <stdio.h>

#include <string.h>


int main(void)

{


}
```

توصیه

کمی کد بنویسید که میدانید کامپایل میشود

دقت کنید که چند `header` جدید را `include` کرده‌ایم تا به توابعی که ممکن است در حل این مسئله به شما کمک کند دسترسی داشته باشید.

```
#include <ctype.h>  #include <cs50.h>

#include <stdio.h>

#include <string.h>


int main(void)

{

    // Prompt the user for two words


    // Compute the score of each word


    // Print the winner

}
```

قبل از نوشتن کد بیشتر، مقداری شبه کد بنویسید

اگر نمیدانید چگونه این مسئله را حل کنید، آن را به چند مسئله کوچک‌تر تقسیم کنید:

1. از کاربر دوبار ورودی بگیرید.
2. امتیاز هر کلمه را محاسبه کنید.
3. برنده را چاپ کنید.

بیایید مقداری شبه کد به عنوان کامنت بنویسیم تا این مسایل را به ما یادآوری کنند:

در بعضی از تمرین‌ها، مثل این، ممکن است راهنمایی‌هایی داده شود که شما را به کل راه‌حل رهنمون شود. با اینکه شما مجاز به استفاده از این راهنمایی‌ها هستید، اما توصیه اکید ما این است که حتما اول خودتان تلاش کنید! بقیه مسائل در این مجموعه تمرین‌ها اینگونه نیستند و معمولا این نوع مسائل صرفا یک نسخه دست‌گرمی برای مسائل بزرگتری هستند که بعدا باید با آنها دست و پنجه نرم کنید.

```
#include <ctype.h>  #include <cs50.h>

#include <stdio.h>

#include <string.h>


int main(void)

{

    // Prompt the user for two words

    string word1 = get_string("Player 1: ");

    string word2 = get_string("Player 2: ");


    // Compute the score of each word


    // Print the winner

}
```

شبه کد ها را به کد تبدیل کنید

اول به این فکر کنیم که چطور میتوانیم از کاربر دوبار ورودی بگیریم. می‌توانید از get_string که تابعی در کتابخانه CS50 است برای گرفتن یک رشته استفاده کنید

```c
#include <ctype.h> #include <cs50.h>

#include <stdio.h>

#include <string.h>


int compute_score(string word);


int main(void)

{

    // Prompt the user for two words

    string word1 = get_string("Player 1: ");

    string word2 = get_string("Player 2: ");


    // Compute the score of each word

    int score1 = compute_score(word1);

    int score2 = compute_score(word2);


    // Print the winner

}


int compute_score(string word)

{

    // Compute and return score for word

}
```

سپس به این فکر کنیم که چطور میتوانیم امتیاز هر کلمه را محاسبه کنیم. از آنجایی که الگوریتم امتیازدهی برای هر دو کلمه به شکل یکسان عمل می‌کند، فرصت خوبی برای ایجاد انتزاع دارید! دراینجا یک تابع به اسم `compute_score` میسازیم که یک رشته با عنوان `word` میگیرد و یک عدد صحیح به عنوان امتیاز کلمه برمیگرداند.

```
#include <ctype.h> #include <cs50.h>

#include <stdio.h>

#include <string.h>


// Points assigned to each letter of the alphabet

int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4, 8, 4, 10};

int compute_score(string word);


int main(void)

{

    // Prompt the user for two words

    string word1 = get_string("Player 1: ");

    string word2 = get_string("Player 2: ");


    // Compute the score of each word

    int score1 = compute_score(word1);

    int score2 = compute_score(word2);

    // Print the winner

}


int compute_score(string word)

{

    // Compute and return score for word

}
```

حالا به پیاده سازی `compute_score` میپردازیم. برای این کار شما باید ارزش هر حرف در کلمات را بدانید. میتوانید هر حرف و ارزش متناظر آن را با یک آرایه به هم مرتبط کنید. به یک آرایه 26 تایی از اعداد صحیح با عنوان `POINTS` فکر کنید که در آن، اولین عدد، ارزش حرف A، عدد دوم ارزش B و الی آخر باشد. در زبان C، با تعریف و مقداردهی اولیه این آرایه خارج از تابع، میتوانید آن را در دسترس تمام توابع قرار دهید؛ از جمله `compute_score`.

برای پیاده سازی `compute_score` اول سعی کنید ارزش هر حرف را در `word` پیدا کنید.

- به یاد داشته باشید که برای پیدا کردن کاراکتر nام در یک رشته مانند `s`، میتوانید از `s[n]` استفاده کنید. پس برای مثال، `word[0]`، اولین کاراکتر `word` را میدهد.
- به یاد داشته باشید که کامپیوترها از کدهای ASCII برای نشان دادن کاراکترها استفاده می‌کنند. این استاندارد هر کاراکتر را با یک عدد نشان می‌دهد.
- همچنین به یاد داشته باشید که index صفر `POINTS`، `POINTS[0]`، ارزش A را به شما می‌دهد. به این فکر کنید که چطور میتوانید نمایش عددی "A" را به index ارزش آن تبدیل کنید. این موضوع در مورد "a" چه میشود؟ پس باید برای حروف کوچک و بزرگ، تبدیلات متفاوتی انجام دهید. توابع `isupper` و `islower` میتوانند به کمک شما بیایند.
- به کاراکترهایی که حرف نیستند باید امتیاز صفر تعلق بگیرد. مثلا "!"، صفر امتیاز می‌گیرد.

اگر بتوانید ارزش یک کاراکتر در `word` را محاسبه کنید، آنگاه می‌توانید با استفاده از یک حلقه، همین کار را برای سایر کاراکترها نیز انجام دهید. بعد از کمی تلاش خودتان، می‌توانید از راهنمایی زیر نیز استفاده کنید!

```
#include <ctype.h> #include <cs50.h>

#include <stdio.h>

#include <string.h>


// Points assigned to each letter of the alphabet

int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4, 8, 4, 10};


int compute_score(string word);


int main(void)

{

    // Prompt the user for two words

    string word1 = get_string("Player 1: ");

    string word2 = get_string("Player 2: ");


    // Compute the score of each word

    int score1 = compute_score(word1);

    int score2 = compute_score(word2);


    // Print the winner

}


int compute_score(string word)

{

    // Keep track of score
```

```
int score = 0;

    // Compute score for each character
    for (int i = 0, len = strlen(word); i < len; i++)
    {
        if (isupper(word[i]))
        {
            score += POINTS[word[i] - 'A'];
        }
        else if (islower(word[i]))
        {
            score += POINTS[word[i] - 'a'];
        }
    }

    return score;
}
```

در آخر، آخرین مرحله شبه کد خود را پیاده سازی کنید: چاپ کردن برنده. به یاد داشته باشید که دستور if میتواند برای بررسی درستی یک شرط استفاده شود. همچنین میتوانید از else if و else برای بررسی شرطهای مجزای دیگر استفاده کنید.

```
if (/* Player 1 wins */)  {
    // ...
}
else if (/* Player 2 wins */)
{
    // ...
}
else
{
    // ...
}
```

بعد از اینکه خودتان تلاش کردید که کارهای فوق را انجام دهید، میتوانید به راهنمایی زیر نیز نگاهی بیندازید (یا حتی راه حل کامل!).

```c
#include <ctype.h> #include <cs50.h>

#include <stdio.h>

#include <string.h>

// Points assigned to each letter of the alphabet

int POINTS[] = {1, 3, 3, 2, 1, 4, 2, 4, 1, 8, 5, 1, 3, 1, 1, 3, 10, 1, 1, 1, 1, 4, 4, 8, 4, 10};

int compute_score(string word);

int main(void)

{

    // Prompt the user for two words

    string word1 = get_string("Player 1: ");

    string word2 = get_string("Player 2: ");


    // Compute the score of each word

    int score1 = compute_score(word1);

    int score2 = compute_score(word2);


    // Print the winner

    if (score1 > score2)

    {

        printf("Player 1 wins!\n");

    }

 else if (score1 < score2)

    {

        printf("Player 2 wins!\n");

    }
```

```
else
    {
        printf("Tie!\n");
    }
}
int compute_score(string word)
{
    // Keep track of score
    int score = 0;

    // Compute score for each character
    for (int i = 0, len = strlen(word); i < len; i++)
    {
        if (isupper(word[i]))
        {
            score += POINTS[word[i] - 'A'];
        }
        else if (islower(word[i]))
        {
            score += POINTS[word[i] - 'a'];
        }
    }
    return score;
}
```

برنامه شما باید مثل این مثال‌ها عمل کند:

```
$ ./scrabble Player 1: Question?

Player 2: Question!

Tie!
```

```
$ ./scrabble Player 1: red

Player 2: wheelbarrow

Player 2 wins!
```

```
$ ./scrabble Player 1: COMPUTER

Player 2: science

Player 1 wins!
```

```
$ ./scrabble Player 1: Scrabble

Player 2: wiNNeR

Player 1 wins!
```

درستی

در ترمینال، دستور زیر را اجرا کنید تا درستی کد خود را ارزیابی کنید

```
check50 cs50/problems/2025/x/scrabble
```

استایل

در ترمینال، دستور زیر را وارد کرده تا طراحی کد خود را ارزیابی کنید.

```
style50 scrabble.c
```

با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
submit50 cs50/problems/2025/x/scrabble
```

CS50x Iran

Harvard's Computer Science 50x Iran