



تمرین کامپیوتری دوم مبانی کامپیوتر و برنامه سازی



اساتید درس: دکتر هاشمی، دکتر مرادی

طراحان: آریا عازم، مهدی حاجی، مجید صادقی نژاد

مقدمه

سلام، در تمرین کامپیوتری دوم قرار است بازی محبوب **۲۰۴۸** را در محیط Terminal پیاده سازی کنید! چگونگی طراحی بازی و قوانین در ادامه توضیح داده شده است لطفا تمامی نکات تمیزی کد، که در کلاس بیان شده است را رعایت کنید همچنین شما فقط مجاز به استفاده از قابلیت هایی هستید که در درس تاکنون به شما تدریس شده است. **در انتهای این فایل می توانید نکات مهم مربوط به تحویل تمرین و همچنین نحوه نمره دهی را ملاحظه کنید.**



اهداف آموزشی

- ☐ استفاده از آرایه های چند بعدی به نحو مناسب
- ☐ استفاده از توابع برای افزایش خوانایی و سرعت و کارایی کد
- ☐ پیاده سازی یک محیط گرافیکی ساده در ترمینال



طراحی برنامه

با آغاز برنامه یک صفحه ی بازی که شامل چهار جز است نمایش داده می شود و منتظر می ماند تا شما ورودی خود را وارد کنید. این چهار جز و توضیحات آن ها در زیر آمده است:

```
*****
Your Score: 6
-----
| 2 |   |   |   |
|   |   |   |   |
|   | 4 |   |   |
|   |   |   |   |
-----
Move (L/R/U/D) or Quit (Q): █
```

• جدا کننده

پیش از هر بخشی باید خطی چاپ شود که هر مرحله را از مرحله های قبلی جدا کند. در اینجا این خط با استفاده از کاراکتر * کشیده شده است.

• نمایشگر امتیازات

دقیقا در بالای صفحه ی بازی باید امتیاز کاربر در مرحله ای که قرار دارد نمایش داده شود این امتیاز مجموع اعدادی است که در این مرحله از بازی در صفحه بازی وجود دارد.

• صفحه ی بازی

این صفحه که یک مربع ۴ در ۴ است قسمت اصلی برنامه ی شماسست که بازی ۲۰۴۸ در آن رخ می دهد. شیوه ی نمایش این صفحه عینا مشابه تصاویر بالا و تصاویر بعدی باید پیاده سازی شود . همانطور که در تصویر مشاهده می شود خانه هایی که خالی هستند در واقع شامل عدد صفر هستند اما عدد صفر در این بازی نمایش داده نمیشود قاعدتا صفحه ی بازی به صورت یک آرایه ی دو بعدی است که در هر مرحله با استفاده از یک تابع محتویات آن به طرز مناسبی در صفحه چاپ می شود. قوانین بازی ۲۰۴۸ در بخش بعدی توضیح داده خواهد شد.

• دریافت کننده ی حرکت بعدی

در زیر صفحه ی بازی یک عبارت چاپ می شود که نمایش دهنده ی راهنمای حرکات برای کاربر است در جلوی این عبارت و بدون رفتن به خط بعدی برنامه ی شما باید حرکت بعدی کاربر را بر اساس همان چیزی که در راهنما نشان داده است دریافت کند و با فشردن کلید enter توسط کاربر حرکت کاربر انجام شود و بازی به مرحله ی بعد برود. همچنین در صورتی که کاربر پیامی به غیر از حرکت قابل پذیرش که توسط راهنما نشان داده می شود وارد کند باید پیغام invalid move نمایش داده شود و مجددا بخش های همان مرحله بدون هیچ گونه تغییری در بازی چاپ شوند و در صورتی که کاربر عبارت Q را به عنوان ورودی وارد کرد بازی تمام شود.

ساختار برنامه به این صورت است که شما در هر مرحله از بازی یک بار این چهار جز را که محیط بازی نامیده می شوند را چاپ می کنید نیازی به پاک کردن مراحل قبلی از ترمینال نیست .

قوانین بازی ۲۰۴۸

پیشنهاد می شود بازی ۲۰۴۸ استاندارد را حداقل یک بار بازی کنید تمامی قوانین پیاده سازی این تمرین کامپیوتری دقیقاً مطابق قوانین بازی ۲۰۴۸ است. یک نسخه آنلاین از بازی در لینک زیر موجود است:

<https://play2048.co/>

با هر حرکت به چپ (L)، راست (R)، بالا (U)، یا پایین (D) تمام اعداد داخل صفحه بازی در صورتی که بتوانند حرکت می کنند این حرکت ها در جهت داده شده توسط کاربر به دو صورت است:

• حرکت تا افزوده شدن به عدد دیگر !

می دانیم در این بازی فقط اعداد توان های دو وجود دارند. اگر دو عدد یکسان در کنار هم قرار گرفته باشند یا بین آن ها هیچ عدد غیریکسان دیگری قرار نگرفته باشد در صورتی که با جهت داده شده توسط کاربر به هم برسند به هم افزوده می شوند و در جایگاه جدید به جای آن دو عدد حاصل جمع قرار می گیرد همچنین این حاصل جمع جدید تا جایی که به بن بست بخورد جلو می رود همچنین توجه داشته باشید که افزوده شدن ها در این بازی فقط بین دو عدد رخ می دهد و سه عدد یکسان در یک مرحله نمی توانند با هم جمع شوند ! زیرا از حالت توان دو خارج می شویم!

• حرکت تا رسیدن به بن بست !

اگر یک عدد نتواند به عدد دیگر افزوده شود تا جایی که به بن بست بخورد جلو می رود در این بازی بن بست ها شامل عدد غیر یکسان با عدد مورد نظر و یا دیواره های صفحه ی بازی می شوند

بعد از اعلام حرکت کاربر در صورتی که حرکت کاربر حرکت قابل انجام باشد (حداقل یک عدد توان حرکت در آن جهت را داشته باشد) حرکت کاربر انجام می شود و سپس یک عدد (۲ یا ۴) در یک مکان شانسی از صفحه ی بازی که خالی است افزوده می گردد و سپس به مرحله بعد می رویم و محیط بازی مجدداً چاپ می شود. همچنین در آغاز بازی دو عدد شانسی (۲ یا ۴) در دو مکان شانسی از صفحه اضافه می شوند.

در صورتی که حرکت کاربر قابل انجام نباشد بدون اضافه شدن عدد جدید به صفحه بازی یک بار دیگر محیط بازی به همان حالت قبل چاپ می شود.

اگر تمام صفحه بازی پر از عدد بشود و دیگر هیچ حرکتی امکان پذیر نباشد یکبار دیگر محیط بازی چاپ می شود اما به جای دریافت کننده ی حرکت بعدی عبارت GAME OVER چاپ می گردد و اجرای برنامه خاتمه می یابد.

در صورتی که کاربر بتواند یک عدد ۲۰۴۸ در صفحه بازی بسازد بازی تمام شده است بنابراین یکبار دیگر محیط بازی چاپ می شود اما به جای دریافت حرکت بعدی عبارت YOU WIN چاپ می گردد و اجرای برنامه خاتمه می یابد.

در تصاویر زیر می توانید چند مرحله از بازی را ببینید (به ترتیب از راست به چپ):

```
*****
Your Score: 24
-----
|   |   | 2 |   |   |
| 4 |   |   |   |   |
| 8 |   |   |   |   |
| 2 |   |   | 4 | 4 |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 26
-----
|   |   |   |   |   |
| 4 |   |   |   | 2 |
| 8 |   |   |   |   |
| 2 | 2 |   | 4 | 4 |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): L
*****
Your Score: 28
-----
|   |   |   |   |   |
| 4 | 2 |   |   |   |
| 8 |   |   |   |   |
| 4 | 8 |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): L
*****
Your Score: 30
-----
|   |   | 2 |   |   |
| 4 | 4 |   |   |   |
| 8 |   |   |   |   |
| 4 | 8 |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): L
*****
Your Score: 34
-----
|   |   |   |   |   |
| 2 |   |   |   |   |
| 8 |   |   |   |   |
| 8 | 4 |   |   |   |
| 4 | 8 |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 38
-----
|   |   |   |   |   |
| 2 |   |   |   |   |
| 16| 4 | 4 |   |   |
| 4 | 8 |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): L
```

```
*****
Your Score: 8
-----
|   |   |   |   |   |
| 4 |   |   |   |   |
| 4 |   |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 12
-----
|   |   |   |   |   |
|   |   |   |   |   |
| 4 |   |   |   |   |
| 8 |   |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 12
-----
|   |   |   |   |   |
|   |   |   |   |   |
| 4 |   |   |   |   |
| 8 |   |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): U
*****
Your Score: 14
-----
|   |   |   |   |   |
| 4 |   |   |   |   |
| 8 |   |   |   |   |
| 2 |   |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 18
-----
|   |   |   | 4 |   |
| 4 |   |   |   |   |
| 8 |   |   |   |   |
| 2 |   |   |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
*****
Your Score: 22
-----
|   |   |   |   |   |
| 4 |   |   |   | 4 |
| 8 |   |   |   |   |
| 2 |   | 4 |   |   |
|   |   |   |   |   |
-----
Move (L/R/U/D) or Quit(Q): D
```

شیوه ی پیاده سازی

این تمرین کامپیوتری دارای دو فاز است که در فاز اول شما تعداد از قابلیت های بازی را پیاده سازی کرده و در فاز دوم به سراغ باقی قابلیت ها می روید.

فاز اول

در این فاز انتظار می رود که شما پنج قابلیت از بازی را در پنج تابع جداگانه که هر کدام وظیفه ی خاصی دارند پیاده سازی می کنید:

- تابع چاپ کننده ی صفحه ی بازی (printBoard)
- تابع مقدار دهی اولیه بازی (initializeBoard)
- تابع تولید کننده مقدار تصادفی (addRandom)
- تابع حرکت صفحه ی بازی به سمت چپ (moveLeft)
- تابع محاسبه امتیاز (calculateScore)

تابع چاپ کننده ی محیط بازی

این تابع صفحه ی بازی، بخش امتیازات و بخش دریافت ورودی را چاپ می کند، و منتظر دریافت ورودی می ماند سپس ورودی دریافتی از کاربر را به عنوان خروجی بر می گرداند. همچنین ورودی این تابع آرایه دو بعدی صفحه ی بازی ۲۰۴۸ است. تابع محاسبه ی امتیاز در داخل این تابع فراخوانی می شود و مقدار امتیاز را بر می گرداند

تابع مقدار دهی اولیه بازی

واضح است که صفحه ی بازی ۲۰۴۸ یک آرایه دو بعدی است که خانه هایی که خالی هستند در واقع دارای مقدار ۰ هستند اما این مقدار در هنگام چاپ صفحه ی بازی نمایش داده نمی شود. این تابع در آغاز بازی یک آرایه ی دو بعدی تحت عنوان صفحه بازی می گیرد که خانه های این آرایه خالی و دارای مقدار بی ارزش^۱ هستند سپس تمام خانه های صفحه ی بازی را با مقدار صفر پر می کند و پس از آن طبق قوانین بازی در دو خانه ی شانسی از صفحه ی بازی دو مقدار شانسی (۲ یا ۴) را قرار می دهد. ورودی این تابع آرایه ی صفحه ی بازی است و خروجی ندارد. همچنین قرار دادن مقدار شانسی در خانه ی شانسی در این تابع توسط تابع تولید کننده ی مقدار تصادفی صورت می گیرد.

¹ Garbage value

تابع تولید کننده ی مقدار تصادفی

این تابع، آرایه ی صفحه ی بازی را به عنوان ورودی گرفته و یک مقدار تصادفی (۲ یا ۴) در یکی از خانه های خالی (یا همان دارای مقدار صفر) به صورت تصادفی می گذارد و خروجی خاصی ندارد. برای تولید مقدار تصادفی شما نیاز به استفاده از دو کتابخانه ی `<stdlib.h>` , `<time.h>` دارید در ابتدای تابع main خود تابع `srand(time(0));`

را فراخوانی کنید. سپس در هر کجای کد که نیاز به تولید مقدار تصادفی داشتید می توانید از تابع: `rand();`

استفاده کنید. این تابع برای شما یک مقدار تصادفی `int` تولید می کند. حال چگونه می توان یک مقدار تصادفی `int` را به صفر یا یک تبدیل کرد و سپس آن را به دو یا چهار تبدیل کرد؟ همچنین شما باید از این تابع برای تعیین خانه ی تصادفی خالی ای که می خواهید (۲ یا ۴) را در آن بگذارید استفاده کنید. برای اطلاعات بیشتر در مورد دو تابع `rand` , `srand` می توانید به لینک های زیر مراجعه کنید:

https://www.w3schools.com/c/ref_stdlib_rand.php

https://www.w3schools.com/c/ref_stdlib_srand.php

تابع حرکت صفحه ی بازی به سمت چپ

در این فاز شما باید فقط حرکت به سمت چپ صفحه ی بازی، را در یک تابع پیاده سازی کنید. تابع به این صورت عمل می کند که به عنوان ورودی صفحه ی بازی را گرفته و حرکت به سمت چپ را روی اعداد داخل صفحه ی بازی، بر اساس قوانین بازی ۲۰۴۸، پیاده سازی می کند همچنین در صورتی که امکان حرکت به سمت چپ میسر بود (یعنی اتفاقی در صفحه ی بازی رخ داد) عدد یک را باز می گرداند و در غیر این صورت عدد صفر را باز می گرداند.

تابع محاسبه امتیاز

این تابع به عنوان ورودی صفحه ی بازی را دریافت می کند و سپس تمام آن آرایه را پیمایش کرده و اعداد داخل آن را با هم جمع کرده و نتیجه را به عنوان خروجی باز می گرداند.

فاز دوم

پس از اتمام این فاز از شما انتظار می رود که پروژه تمام شده باشد و بازی 2048 قابلیت تست و بازی داشته باشد! در انتهای این فاز، بازی ما باید تمام عملکردهای تشریح شده در فازهای اول و دوم را در خود داشته باشد!

تابع های حرکت صفحه ی بازی به سمت راست، بالا و پایین

این توابع هر کدام دقیقا همانند تابع حرکت به سمت چپ که در فاز قبل پیاده سازی شد عمل می کنند و همچنین شباهت بسیار زیاده به تابع حرکت به سمت چپ دارند.

تابع بررسی باخت در بازی

این تابع صفحه ی بازی را به عنوان ورودی می گیرد و در صورتی که دیگر هیچ حرکتی در صفحه ی بازی ممکن نباشد و تمام صفحه ی بازی پر شده باشد مقدار یک را بر می گرداند در غیر این صورت مقدار صفر را بر می گرداند. طبیعتا این تابع پس از انجام هر حرکت توسط کاربر فراخوانی می شود.

تابع بررسی برنده شدن در بازی

این تابع صفحه بازی را به عنوان ورودی می گیرد و در صورتی که عدد ۲۰۴۸ در آن یافت شود ۱ را بر می گرداند و در غیر این صورت عدد ۰ را بر می گرداند. طبیعتا این تابع پس از انجام هر حرکت توسط کاربر فراخوانی می شود.

استفاده از توابع و تکمیل بازی

در نهایت پس از پیاده سازی این توابع با استفاده از تمام آن ها تابع main را بنویسید به طوری که بازی قابل بازی باشد. همچنین شما مجاز هستید هر تابع دیگری را نیز تعریف کنید به شرط آن که در درس مفاهیم آن را آموخته باشید.

بخش امتیازی 😊

• بازگشت یک مرحله ای (Back)

در صورتی که کاربر ورودی B را وارد کند به این معنی است که قصد دارد یک مرحله بازگشت و مجددا بازی را ادامه دهد در این حالت عدد شانسی اضافه شده حذف می گردد و شرایط صفحه به قبل از آخرین حرکت باز می گردد و محیط بازی مجددا چاپ می گردد.

• شروع مجدد (New Start)

در صورتی که کاربر ورودی N را بدهد بازی کاملاً از نو آغاز می شود توجه داشته باشید که در این حالت برنامه شما نباید خاتمه یابد و فقط بازی باید مجدداً شروع شود همچنین نیازی به پاک کردن مراحل قبلی چاپ شده نیست.

```
*****
Your Score: 6
-----
|   |   |   |   |
| 2 | 4 |   |   |
|   |   |   |   |
-----
Move (L/R/U/D) or Quit, Back, New Start(Q, B, N):
```

نکات مهم 😊

- تمامی موارد موجود در صورت تمرین باید در محیط گرافیکی موجود باشند و محیط گرافیکی مشابه تصاویر بالا پیاده سازی شود .
- استفاده از متغیرهای Global به هیچ وجه مجاز نمی باشد و در صورت نیاز باید داده ها را در قالب آرگومان به توابع انتقال دهید. به توابعی که با استفاده از متغیر Global نوشته شده باشند، نمره صفر تعلق می گیرد.
- در صورت وجود هرگونه سوالی، می توانید با کلیک بر روی نام دستیاران آموزشی که نام آن ها در ابتدای این فایل آمده است از طریق ایمیل در ارتباط باشید.
- فایل برنامه ی خود را در یک فایل فشرده با فرمت "zip" و با نام "CA2-SID.zip" قرار دهید که SID همان شماره ی دانشجویی شماست. برای مثال اگر شماره ی دانشجویی شما 810103555 باشد باید نام فایل خود را "CA2-810103555.zip" قرار دهید و آن را در قسمت در نظر گرفته شده در صفحه درس در سامانه ایلرن آپلود نمایید.
- فایل zip شما باید فقط شامل یک فایل C باشد، با نام:
"game.c"
- برنامه های شما باید با زبان برنامه نویسی C نوشته شود و استفاده از دیگر زبان های برنامه نویسی مجاز نیست.
- **مهلت تحویل فاز دوم این تمرین تا ساعت ۲۳:۵۹ روز شنبه ۸ ام دی است.**
- شما در طول این ترم چهار روز مهلت اضافه برای تمرین کامپیوتری دارید که می توانید آن ها را بین تمرین کامپیوتری به انتخاب خودتان تقسیم کنید در صورتی که از مهلت اضافی خود برای یک تمرین استفاده کنید می توانید آن تمرین را بدون کسر نمره، با تاخیر تحویل دهید. همچنین توجه داشته باشید که تمرین کامپیوتری با بیش از دو روز تاخیر تحویل گرفته نمی شوند.

- تمارین کامپیوتری برای یادگیری برنامه نویسی و مباحث مطرح شده در کلاس طراحی می‌شوند و انجام آن‌ها به صورت انفرادی خواهد بود. همچنین در صورت شباهت میان دو پروژه (که به وسیله ی نرم افزارهای مربوطه چک می‌شود.) برای هر دو نفر نمره صفر در نظر گرفته خواهد شد.

نحوه ی نمره دهی 😊

- ☐ نام گذاری مناسب متغیر ها و توابع در کد (۱۵ نمره)
- ☐ مرتب بودن و تمیزی کد و محیط گرافیکی بازی (۳۰ نمره)
- ☐ عدم وجود خطای کامپایل، اخطار (۲۰ نمره)
- ☐ کارایی و عملکرد بازی (۳۵ نمره)
- ☐ بخش امتیازی (۲۰ نمره)

😊 موفق باشید