



به نام خدا

## دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران

### مبانی کامپیوتر و برنامه نویسی



نیمسال اول 01-02

پروژه 2 (بخش نامه)

استاد: دکتر مرادی، دکتر هاشمی

این پروژه دارای بخش امتیازی می باشد. همچنین دقت کنید که برای این پروژه باید فلوچارت الگوریتم ارائه دهید که زمان تحویل آن با پروژه متفاوت می باشد که در بخش تحویل آمده است.

هدف این پروژه، استفاده از تابع، آرایه و Header File و نحوه کار با آن ها می باشد.

سهیل، نوید و پارسا پس از حل شدن مشکلات قبلی خود، حوصله شان سر رفته است و می خواهند تا یک بازی انجام دهند.

آن ها از شما کمک می خواهند تا یک بازی طراحی کرده و آن را در محیط خط فرمان<sup>1</sup> پیاده سازی نمایید. (برای اجرای برنامه در این محیط، کافست بعد از compile کردن کد توسط gcc، به پوشه ای که در آن فایل ذخیره شده رفته و بر روی فرمت خروجی .exe، دوبار کلیک کنید).

نام بازی بخش نامه است و قوانین مشخصی دارد. در ادامه، شرح بازی، قوانین و نحوه پیاده سازی آن توضیح داده شده است.

#### شرح بازی:

این بازی، یک بازی کارتی می باشد که دارای 4 بازیکن (2 تیم 2 نفره) است. نیاز است تا شما به عنوان بازیکن اول، این بازی را شروع کرده و 3 بازیکن دیگر (کامپیوتر)، طبق قوانین بازی خواهند کرد.

در این بازی 4 نوع کارت با علامت متفاوت وجود دارد که هر علامت نیز 11 عدد (1 تا 11) دارد (در مجموع 44 کارت در این بازی موجود است). این علامت ها عبارتند از: A B C D

در ابتدا تمامی کارت ها بُر می خورند. (برای بُر زدن کارت ها، همراه با صورت پروژه یک فایل C. نیز برای شما آپلود شده است که داخل آن تابعی وجود دارد که به شما 44 عدد رندوم بین 0 تا 43 خروجی می دهد که نحوه استفاده از آن در بخش نحوه پیاده سازی شرح داده شده است). سپس هر بازیکن 11 کارت دریافت می کند که این دریافت به صورت 2-4-5 می باشد. (ابتدا تمامی بازیکنان 5 کارت دریافت می کنند، سپس 4 کارت و نهایتاً 2 کارت) همچنین جهت چرخش بازی و پخش کردن کارت ها، به صورت ساعتگرد می باشد.

هنگامی که بازیکن اول (شما) 5 کارت ابتدایی خود را دریافت کرد، باید یکی از علامت هایی که در آن 5 کارت وجود دارد را به عنوان بخش نامه انتخاب کند. پس از انتخاب بخش نامه، سایر بازیکنان کارت های خود را دریافت می کنند.

در ادامه، بازیکن اول یکی از کارت های خود را بازی می کند، و بازیکنان دیگر به ترتیب کارت های خود را بازی می کنند. زمانی که هر چهار بازیکن، کارت های خود را بازی کنند، 4 کارت در صفحه بازی موجود است که با توجه به اعداد روی کارت، تیمی که بیشترین امتیاز را داشته باشد (با توجه به قوانین که جلوتر توضیح داده می شود) برنده دور می شود و بازیکنی که باعث پیروزی تیمش بوده، دور بعد را شروع می کند.

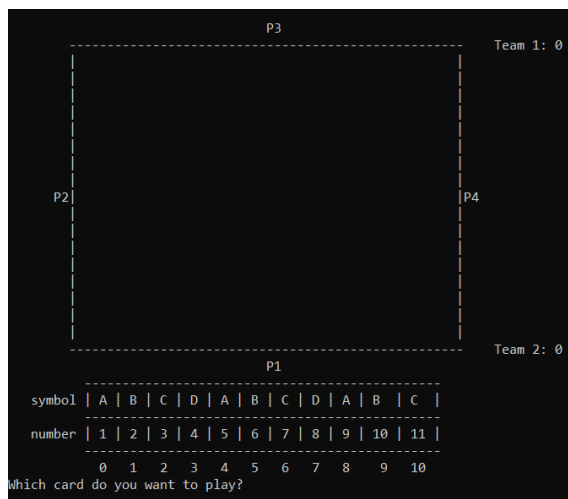
<sup>1</sup> Command Line

## قوانین بازی:

- در هر دور، بازیکنان باید کارتی را بازی کنند که هم علامت کارت شروع می‌باشد. بازیکنان دیگر (کامپیوتر)، باید بزرگترین عدد کارت هم علامت کارت شروع را بازی کنند. (مگر در مواردی که جلوتر شرح داده خواهد شد)
- در صورتی که بازیکنان دیگر (کامپیوتر)، علامت کارت شروع را نداشته باشند، باید طبق الگوریتم زیر بازی کنند:
  1. بازیکن باید بزرگترین عدد بخش نامه را که در دست خود دارد را بازی کند.
  2. در صورتی که بازیکن کارت با علامت بخش نامه ندارد، باید کوچکترین عدد هر علامت دیگری را که در دست خود دارد را بازی کند.
- تنها بازیکن اول می‌تواند در صورت نداشتن علامت کارت در حال بازی، هر کارت دیگری با هر علامت و عدد دلخواهی را بازی کند.
- در صورتی که بازیکن دیگری به غیر از بازیکن اول قرار باشد بازی را شروع کند، نیاز است طبق الگوریتم زیر کارت خود را بازی کند:
  1. کارتی که بزرگترین عدد را دارد و علامت آن بخش نامه نیست را بازی کند.
  2. اگر تمامی کارت‌های بازیکن بخش نامه است، باید بزرگترین عدد آن را بازی کند.
- در هر دور، تیمی برنده است که یکی از اعضای تیم، کارت ارزشمندتری نسبت به سایرین بازی کرده باشد. ارزش کارت‌ها به شرح زیر است:
  1. اگر علامت تمام کارت‌های صفحه بازی یکسان بود، کارتی ارزشمندتر است که عدد بزرگتری نسبت به سایر کارت‌های صفحه بازی داشته باشد.
  2. اگر در بین کارت‌های صفحه بازی علامت بخش نامه وجود داشت، کارتی ارزشمندتر است که علامت آن بخش نامه بوده و عدد بزرگتری داشته باشد.
- اولین تیمی که به امتیاز 6 برسد، برنده بازی خواهد بود.

## نحوه پیاده سازی:

در ابتدا نیاز است تا یک صفحه بازی طراحی شود تا بازیکنان بتوانند کارت های خود را در آن بازی کنند. صفحه بازی باید به شکل زیر باشد:

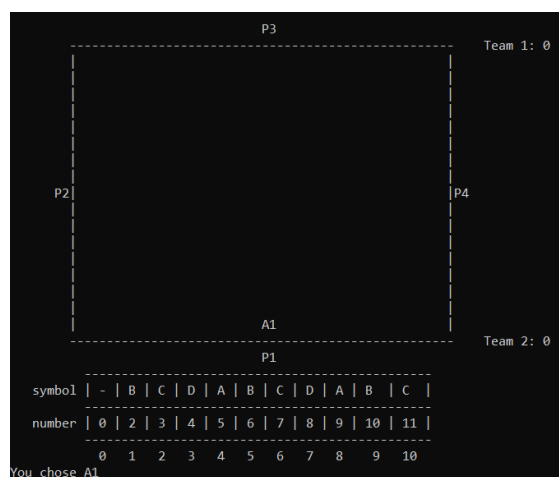


تیم اول بازیکن اول و سوم، و تیم دوم بازیکن دوم و چهارم هستند.

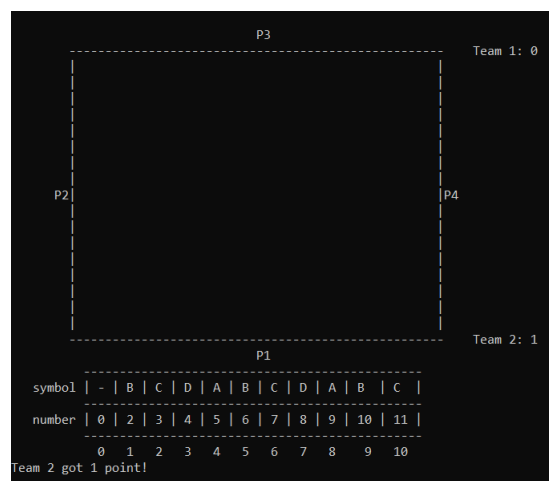
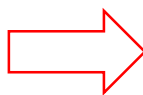
برای انتخاب یک کارت، بازیکن باید با استفاده از یک عدد بین 0 تا 10 که نشان دهنده ایندکس کارت‌های دست خود می‌باشد، کارت موردنظر را انتخاب کند. (اعداد 0 تا 10 زیر سطر number در شکل)

برای ورودی گرفتن نیاز است تا در هر مرحله، پیغام مناسبی نمایش داده شود و پس از ورودی گرفتن نیز، پیغام مناسب نمایش داده شود.

هنگامی که بازیکن کارت موردنظر را انتخاب کرد، لازم است تا همانند شکل زیر، صفحه بازی آپدیت شود:



هنگامی که یک دور از بازی بگذرد، نیاز است تا امتیاز هر تیم با توجه به بخش قوانین آپدیت شود. برای مثال به شکل زیر توجه کنید:



در انتها هر تیمی که برنده شود باید با نمایش یک پیغام مناسب، برنده مشخص شود.

- شما می‌توانید ابعاد زمین را تغییر دهید اما باید دقت کنید که ابعاد صفحه و محل قرارگیری آن‌ها دقیقاً مطابق شکل داده شده باشد.

- نمایش پیام های مناسب در ورودی و بعد از آن، می تواند **ابتکاری** باشد و نیازی نیست دقیقا مانند نمونه باشند.
  - همچنین پیام ها باید به یکی از دو زبان انگلیسی یا فینگلیش نوشته شود.
  - بعد از ورودی گرفتن نیاز است تا دستور **Enter** وارد شود و تا زمانی که **Enter** زده نشده، بازی نباید پیش برود.
- برای ذخیره کردن کارت ها نیاز است تا از آرایه دو بعدی استفاده نمایید. به طوریکه باید آرایه ای به صورت `arr[4][11]` برای هر بازیکن داشته باشید. می توانید به هرنحوی که مایل هستید، بعد اول این آرایه را متناظر با هر علامت درنظر بگیرید. برای مثال ممکن است `arr[1]` را برای علامت B یا C درنظر بگیرید.
- در شروع پروژه آرایه ای تحت عنوان `cards[4][11]` در نظر بگیرید که به صورت زیر پر شده باشد. (نام این آرایه می تواند متفاوت باشد).

`[1, 2, 3, ..., 11], [1, 2, 3, ..., 11], [1, 2, ..., 11], [1, ..., 11]`

این آرایه تمامی کارت های شما را نشان می دهد.

برای بُر زدن کارت ها، از تابع `shuffle_card` که همراه با صورت پروژه برای شما آپلود شده است، می توانید استفاده کنید. نحوه استفاده از این تابع به صورت زیر می باشد (برای استفاده از این تابع به کتابخانه های `stdio.h`, `stdlib.h`, `time.h` نیاز دارید).

```
void shuffle_card(int* card) {
    int i, r, temp;
    for (temp = 0, i = 0; temp < 44; i++, temp++)
        card[temp] = i;
    srand(time(NULL));
    for (i = 43; i > 0; i--) {
        r = rand() % i;
        temp = card[i];
        card[i] = card[r];
        card[r] = temp;
    }
}

int main() {
    int card[44];
    shuffle_card(card);
}
```

لطفا در تابع پارامتری را تغییر ندهید! همچنین برای این تابع نیاز به نوشتن **header file** جدا نیست و میتوانید به طور مستقیم و با کپی پیست کردن آن در قسمت های مختلف کد خود از آن استفاده کنید.

پس از صدا زدن تابع؛ آرایه `card`، شامل اعدادی درهم ریخته از 0 تا 43 می باشد. شما می توانید با محاسبه باقیمانده و خارج قسمت هر عدد بر 11، کارت مورد نظر خود را بیابید. برای مثال:

$$\frac{25}{11} = 2, \quad 25 \% 11 = 3$$

این نشان می دهد که کارت موردنظر، خانه `cards[2][3]` می باشد که در اول پروژه تعریف شده است.

## امتیازی:

احتمالا متوجه شدید که دیگر بازیکنان (کامپیوتر) در این بازی ماهر نیستند! در این بخش می‌خواهیم با افزودن یکسری قوانین بیشتر، آن‌ها را در بازی بهتر کنیم و همچنین بتوانیم بازی را بیشتر از یک دست بازی کنیم. این قسمت دو بخش دارد.

### بخش اول (تسلیم):

در گام اول نیاز است تا بتوانیم بازی را در 5 ست بازی کنیم. هر ست از بازی زمانی به اتمام می‌رسد که یک تیم به امتیاز 6 برسد و یک امتیاز ست کسب می‌کند. در این صورت نیاز است تا جدول نمره دهی تغییر کرده و علاوه بر امتیاز هر ست، امتیاز مجموع که تعداد ست برده شده است را نیز نمایش دهد. برای اینکار می‌توانید از ابتکار خود استفاده کنید و جدول امتیازات را با توجه به سلیقه خود تغییر دهید. قوانین اضافه شده برای این بخش به شرح زیر است:

- ممکن است در یک ست، یک تیم به امتیاز 6 برسد ولی تیم حریف در امتیاز 0 بماند. در این صورت تیمی که امتیاز آن 0 است تسلیم شده و به تیم حریف به جای 1 امتیاز در امتیاز ست‌ها، 2 امتیاز تعلق می‌گیرد.
- بعد از اتمام هر ست، امتیازات هر ست، برای هر دو تیم باید صفر شود اما امتیاز تعداد ست برده شده نباید تغییر کند.

### بخش دوم (بازیکن زیرک):

در این بخش می‌خواهیم بازی را جذاب‌تر کرده و با بازیکنان ماهرتری بازی کنیم. برای این منظور لازم است تا بازیکنان (کامپیوتر) با توجه به شرایط موجود در صفحه زمین و کارت بازی شده توسط هم تیمی خود، بازی کنند. قوانین اضافه شده برای این بخش به شرح زیر است:

- هنگامی که نوبت بازیکنی (کامپیوتر) باشد، در صورتی که در صفحه بازی، کارت هم تیمی او بالاتر از بقیه کارت‌ها است، باید کوچکترین عدد از آن علامت را بازی کند. (فارغ از آنکه بازیکن دیگر تیم حریف ممکن است عدد بالاتری از آن علامت را بازی کند). اگر کارتی از آن علامت در دست خود نداشت، باید با علامتی غیر از بخش نامه و با کوچکترین عدد ممکن بازی کند و اگر غیر از بخش نامه علامت دیگری نداشت، کوچکترین عدد بخش نامه را بازی کند.
- در صورتی که بازیکن دیگری به غیر از بازیکن اول قرار باشد بازی را شروع کند، اگر در دست خود، عدد 10 یا 11 از هر علامتی را نداشت، باید کوچکترین عدد از یک علامت (به جز بخش نامه) را بازی کند.

• چند نکته... :

1. استفاده از هرگونه توابع C++ ممنوع می‌باشد و نمره ای به آن تعلق نمی‌گیرد.
2. پروژه باید نسبت به ورودی‌های نامعتبر، مقاوم بوده و در صورت وارد کردن ورودی نامعتبر، عبارت مناسبی را نمایش دهد. توجه شود که در صورت وارد کردن ورودی نامعتبر، برنامه نباید به اتمام برسد و پس از نمایش عبارت مناسب، باید دوباره ورودی بگیرد و بازی پیش برود.
3. بهتر است هر کدام از مراحل مختلف بازی را در تابع‌های جداگانه بنویسید و از آنها استفاده کنید. برای مثال می‌توانید یک تابع برای نمایش صفحه‌ی بازی بنویسید و در مرحله‌ی نمایش صفحه، تنها آن را صدا بزنید.
4. بهتر است که از یک حلقه‌ی کنترلی کلی برای تکرار مراحل کلی بازی استفاده کنید.
5. برای کارت‌های هر بازیکن حین بازی، از یک آرایه‌ی دو بعدی استفاده کنید.
6. توجه داشته باشید که شما باید در این پروژه، توابع خود را در فایل‌های جداگانه با هدر فایل‌های مخصوص خود بنویسید. شیوه‌ی تقسیم‌بندی توابع به خودتان مربوط است و هر روش قابل قبولی پذیرفتنی است.

• شیوه‌ی نمره دهی:

عنوان	نمره
نامگذاری مناسب و اصولی متغیرها	5
استفاده از تمام ورودی‌های تابع در آن و نامگذاری مناسب توابع	5
عدم وجود قطعه کد تکراری	5
نمایش درست صفحه‌ی بازی در command line	5
دریافت صحیح ورودی در command line	5
استفاده از آرایه‌ی دوبعدی	5
رسیدگی به خطاها	10
نمایش پیام‌های مناسب در هر مرحله از بازی	5
تغییر درست نوبت ها برای شروع بازی	5
پیاده سازی الگوریتم درست برای سایر بازیکنان به شکل صحیح	10
کامنت گذاری مناسب در هر جایی که نیاز به مستندسازی دارد	5
استفاده از header file	5
فلوچارت الگوریتم روند پروژه	10
تست و اجرای برنامه	20
امتیازی (بخش اول)	10
امتیازی (بخش دوم)	20
مجموع	130

## تحويل:

1. تنها فایل‌های با فرمت ".c" و ".h" را در یک فایل زیپ با فرمت "zip" و با نام CA2-SID.zip قرار دهید که SID همان شماره‌ی دانشجویی شماست. برای مثال اگر شماره‌ی دانشجویی شما 810101000 باشد، باید نام فایل خود را CA2-810101000.zip قرار دهید و آن را در قسمت در نظر گرفته شده در صفحه درس در سامانه ایلرن آپلود نمایید.
2. برنامه‌های شما باید با زبان برنامه نویسی C نوشته شود و استفاده از دیگر زبان‌های برنامه نویسی مجاز نیست.
3. لازم است تا برای این پروژه، **فلوچارت الگوریتم** روند پروژه نیز طراحی شده و آپلود گردد. مهلت آپلود فلوچارت تا **ساعت 23:55 جمعه 4 آذر** می‌باشد و بدون تاخیر می‌باشد. برای بخش امتیازی نیازی به فلوچارت نمی‌باشد.
4. مهلت آپلود پروژه تا **ساعت 23:55 جمعه 11 آذر** است. تأخیر در سه روز نخست به ازای هر روز ۱۰ درصد و در سه روز دوم به ازای هر روز ۱۵ درصد جریمه خواهد داشت. پس از این شش روز، به هیچ وجه نمره‌ای در نظر گرفته نخواهد شد. همچنین توجه کنید که با توجه به تاخیر در نظر گرفته شده به هیچ وجه امکان تمدید پروژه‌ها وجود ندارد.
5. **این پروژه دارای تحويل حضوری می‌باشد.** لازم است تا در زمان تحويل، بر پروژه و کد خود، تسلط کامل داشته باشید. پروژه‌ها 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 ی صفر در نظر گرفته خواهد شد.
6. در صورت وجود هرگونه سوال میتوانید پرسشهای خود را در فروم درس (در بخش مربوط به این پروژه) مطرح نمایید و یا از طریق ایمیل یا تلگرام با [سهیل](#)، [نوید](#)، [پارسا](#) در ارتباط باشید.