

باسمه تعالی

## الف) پروژه فکر بکر

برنامه‌ای بنویسید که یک عدد تصادفی  $m$  رقمی با رقم‌های غیر تکراری تولید کند ( $m=4,5,6,7$ ) و از کاربر بخواهد که آن را حدس بزند. به ازای هر حدس کاربر، کامپیوتر اعلام کند که چند رقم از حدس کاربر در عدد تولید شده در مکان صحیح وجود دارد و چند رقم وجود دارد ولی در مکانی غیر از مکان واقعی. مثلاً اگر کامپیوتر عدد 05128 را تولید کرده باشد و کاربر عدد 02975 را حدس بزند، یک رقم (صفر) در مکان صحیح قرار دارد و دو رقم (۵ و ۲) در مکانی غیر از مکان واقعی قرار دارد. برنامه تا جایی ادامه می‌یابد که کاربر به جواب صحیح برسد یا اینکه تعداد حدس‌ها از ۱۲ بیشتر شود. لازم است برنامه شما در نهایت بر اساس تعداد حدس و میزان سختی بازی (مقدار متغیر  $m$  که در ابتدا از کاربر پرسیده می‌شود)، امتیازی به کاربر دهد (فرمول دلخواهی را در نظر بگیرید) و اگر کاربر رکوردشکنی کرده باشد، نام او را بپرسد و در فایل ثبت کند.

برنامه باید دارای بخشی باشد که نام و امتیاز فرد رکوردشکن را نشان دهد.

نمره اضافی:

امتیازدهی بر اساس زمان هم باشد.

نام سه نفر برتر را در فایل نگه دارد.

## ب) پروژه تشخیص چهره

فرض کنید اطلاعات مربوط به یک تصویر، در یک فایل باینری قرار گرفته است. برنامه‌ای بنویسید که نام یک فایل را از کاربر بپرسد و تعیین کند که آیا در این تصویر، چهره‌ای وجود دارد یا خیر. در صورت مثبت بودن جواب، مکان تقریبی چهره‌ها را تعیین کنید.

برای این منظور باید از الگوریتم‌های تشخیص چهره استفاده کنید. یک الگوریتم ساده برای این کار در ادامه توضیح داده می‌شود. کافی است مراحل توضیح داده شده را قدم به قدم انجام دهید تا شما هم بتوانید یک برنامه تشخیص چهره داشته باشید!

توضیح مقدماتی:

در دنیای کامپیوتر، برای بیان کردن رنگ‌ها از چند سیستم مختلف استفاده می‌شود. معروفترین آن‌ها سیستم RGB است. در این سیستم، رنگ هر پیکسل با سه عدد یک بایتی (سه عدد از ۰ تا ۲۵۵) مشخص می‌شود. این سه عدد به ترتیب بیان‌کننده میزان رنگ قرمز، سبز و آبی در تشکیل رنگ نهایی است. مثلاً سه‌تایی (150,0,130) نشانگر رنگی است که در آن اثری از سبز وجود ندارد و ترکیبی از قرمز و آبی با نسبت مشخص است که احتمالاً چیزی شبیه به بنفش خواهد بود. دو سیستم دیگر بیان رنگ‌ها، یکی نرمالایز شده RGB است (یا همان rg) و دیگری HSL.

ساختار فایل

شما برای تشخیص چهره به اطلاعات مربوط به هر پیکسل احتیاج دارید و باید هم از سیستم HSL و هم rg استفاده کنید. به همین علت فرض کنید فایلی از نوع ساختار زیر در اختیار داریم:

```
struct pixel {  
    int H,S,L;  
    int r,g;  
}
```

درون فایل تعداد زیادی از این ساختارها وجود دارد (برای هر پیکسل یک ساختار). شما باید تمام پیکسل‌ها را به نوبت بررسی کنید و در صورتی که پیکسل‌ها شرایط خاصی داشته باشند احتمالاً نشانگر یک تصویر خواهد بود.

الگوریتم تشخیص چهره بدین ترتیب است که شما باید تمام پیکسل‌های موجود در تصویر را بررسی کنید و پیکسل‌هایی را که می‌توانند تصویر چهره باشند را مشخص کنید. این کار در یک فرآیند چند مرحله‌ای انجام می‌شود و هر بار بر اساس برخی معیارها، شما تعدادی از پیکسل‌های زائد را کنار می‌گذارید (سیاه می‌کنید) تا در نهایت فقط پیکسل‌هایی باقی بمانند که نشانگر یک چهره هستند.

مراحل انجام پروژه:

۱- نام فایل حاوی اطلاعات  $rg$  و  $HSL$  را از کاربر بپرسید.

۲- اطلاعات مربوطه را از فایل استخراج کنید تا مشخصات تمام پیکسل‌ها معلوم شود.

۳- هر پیکسل را بر اساس فرمول‌های مرحله اول موجود در [اسلاید](#) بررسی کنید (محدوده پارامتر  $H$ ، مقدار  $F1$ ،  $F2$  و  $W$ ).

۵- یک تصویر سیاه و سفید رسم کنید که در آن پیکسل‌های نامرتب همگی به رنگ سیاه باشند و پیکسل‌هایی که احتمالاً نشانگر یک چهره هستند به رنگ سفید نمایش داده شود.

تا اینجا مرحله اول پروژه شما انجام شده است. برای آنکه نتیجه دقیق‌تری به دست بیاورید مراحل زیر را ادامه دهید.

۶- با توجه به مبحث شماره‌گذاری که در اسلایدهای توضیح آمده است، تعداد بخش‌هایی از تصویر را که ممکن است نشانگر یک چهره باشد، تعیین کنید. (مجموعه پیکسل‌های سفید متصل به هم را به عنوان یک شیء در نظر بگیرید - ربطی به مفهوم برنامه‌نویسی شیء‌گرا ندارد! معنی فارسی شیء منظور است! - سپس تعداد اشیاء را بشمارید)

۷- تعداد چهره‌های بالقوه (تعداد اشایء مرحله قبل) را چاپ کنید. همچنین برای هر شیء مشخص کنید که اگر بخواهیم کادری دور آن در نظر بگیریم، طول، عرض و مساحت کادر چقدر خواهد بود.

مرحله دوم پروژه شما هم به پایان رسید. هم‌اکنون شما تعداد چهره‌های بالقوه را به دست آورده‌اید. برای آنکه ببینید از این میان کدام‌ها واقعا چهره هستند و کدام‌ها نه، لازم است مراحل زیر را دنبال کنید.

۸- تست‌های شماره ۴ تا ۶ موجود در اسلایدها را انجام دهید تا گزینه‌های غیرمرتبط را شناسایی کنید.

۹- دور گزینه‌های باقی‌مانده، کادری رسم کنید.

کار شما اینجا به پایان می‌رسد. تبریک می‌گوییم. شما یک برنامه **Face Detection** نوشته‌اید.

در صورت تمایل می‌توانید تست شماره ۷ اسلایدها را هم انجام دهید تا الگوریتم شما بهتر شود.