



گروه کامپیوتر
دانشکده مهندسی

مبانی کامپیوتر و برنامه‌سازی

نیمسال اول ۹۷-۹۸

پروژه پایانی

مدرس: مصطفی نوری بایگی

زمان تحویل: ۱ بهمن ۱۳۹۶

REABSO

در این پروژه شما باید جدولی از حروف را از تصویر ورودی به دست بیاورید. سپس در میان حروف به دست آمده، یک سری لغات را پیدا کرده و در خروجی نمایش دهید. برای این کار لازم است تا سیستم، طرحی کلی از هر حرف داشته باشد.

۱ فازبندی پروژه

۱.۱ فاز ۱: برش عکس (Crop)

در این فاز باید برای یک عکس ورودی سفیدی‌های اطراف آیتام موجود در عکس را برش داده و عکس برش یافته را ذخیره کنید. به مثال زیر توجه کنید.



شکل ۱: عکس اصلی (سمت چپ) و عکس کراپ شده (عکس راست)

روش پیاده‌سازی این بخش به این صورت می‌باشد که ابتدا با استفاده از حلقه، نقطه بالایی آیتم موجود در عکس را شناسایی کرده و سپس با حرکت در جهت‌های راست، چپ و پایین، مرزهای آیتم را تشخیص دهید. سپس مقادیر موجود در این بخش از عکس را در آرایه جدید ریخته و ذخیره کنید.

۲.۱ فاز ۲: تغییر اندازه (Scale)

در این فاز برای یک عکس ورودی باید آن را به عکس جدید با طول و عرض جدید تبدیل کنید. به مثال زیر توجه کنید.



شکل ۲: عکس اصلی با اندازه 400×700 (سمت چپ) و عکس ری‌سایز شده با اندازه 500×500 (عکس راست)

روش پیاده‌سازی این بخش به این صورت است که باید تغییر اندازه را در دو جهت انجام داد. برای مثال ابتدا در جهت افقی و سپس در جهت عمودی. برای تغییر سایز در جهت افقی، لازم است تا نسبت (Ratio) عرض عکس جدید را به عکس اصلی محاسبه کنیم. برای مثال اگر عرض عکس اصلی برابر 300 و عرض عکس جدید برابر 500 باشد، نسبت عرض برابر $5/3$ خواهد شد. در نتیجه هر پیکسل در عکس اصلی باید تبدیل به $5/3$ پیکسل (یا به عبارتی 1 و $2/3$ پیکسل) در عکس جدید شود. در نتیجه پیکسل اول در عکس اصلی یک پیکسل در عکس جدید و $2/3$ از پیکسل بعدی از عکس جدید را دربر خواهد گرفت. به مثال‌های زیر توجه کنید.

1	1	1
*	*	*
*	*	*

1	$2/3 + 1/3$	1	$1/3 + 2/3$	1
#	#	#	#	#
#	#	#	#	#

شکل ۳: عکس اصلی (سمت چپ) و عکس ری‌سایز شده (عکس راست)؛ هر پیکسل به صورت یک رنگ مشخص شده و عدد مربوط به هر رنگ، درصد حضور پیکسل عکس اصلی در عکس جدید می‌باشد.

3	6	9
*	*	*
*	*	*

3	4	6	8	9
#	#	#	#	#
#	#	#	#	#

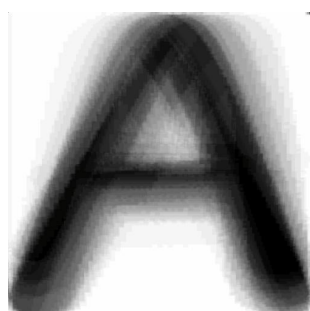
شکل ۴: عکس اصلی (سمت چپ) و عکس ری سائز شده (عکس راست)؛ مثال قبل همراه با عدد

۳.۱ فاز ۳: یادگیری (Learning)

در این فاز باید با استفاده از دیتاستی (مجموعه‌ای از چندین عکس) که در اختیار شما می‌گذاریم. یک طرح کلی از هر حرف ساخته و ذخیره کنید تا هر حرف برای ماشین قابل فهم باشد. به مثال‌های زیر توجه کنید.



شکل ۵: دیتاست مربوط به حرف A



شکل ۶: طرح کلی حرف A

همانطور که مشاهده می‌کنید خروجی یک طرح کلی از حرف A می‌باشد که از فایل‌های موجود در دیتاست به دست آمده است. روش پیاده‌سازی این بخش به این صورت می‌باشد که باید هر یک از فایل‌های موجود در دیتاست را خوانده، برش داده و به

یک اندازه مشخص تغییر اندازه دهید تا اندازه همه آن‌ها با هم برابر باشند. سپس با میانگین‌گیری پیکسل‌های تمامی عکس‌های موجود در دیتاست، طرح کلی به دست آمده را ذخیره کنید.

۴.۱ فاز ۴: تشخیص حرف

در این فاز باید یک عکس شامل یک حرف را به عنوان ورودی دریافت کنید و مشخص کنید که این حرف کدامیک از حروف یادگیری شده توسط برنامه است. روش پیاده‌سازی این بخش به این صورت می‌باشد که ابتدا عکس ورودی را برش داده و سپس به اندازه طرح‌های کلی حروف (فاز ۳) تبدیل می‌کنید. سپس تصویر ورودی را با طرح‌های کلی مقایسه می‌کنید و هر حرفی که مجموع اختلاف مقادیر پیکسل کمتری دارد را به عنوان حرف تشخیص داده شده انتخاب کنید.

۵.۱ فاز ۵: خواندن جدول (Read Table)

در این فاز باید یک عکس شامل جدولی از حروف به عنوان ورودی دریافت کنید و حروف موجود در جدول را در یک آرایه دوبعدی ذخیره کنید. به مثال زیر توجه کنید.

R	E	I	B
B	A	R	E
O	B	S	R
R	A	S	R

روش پیاده‌سازی این بخش به این صورت می‌باشد که ابتدا تعداد ستون‌ها و ردیف‌های جدول را به دست آورده و سپس عکس ورودی را بخش‌بندی می‌کنید؛ به طوری که در هر بخش تنها یک حرف موجود باشد. سپس با استفاده از تابعی که در فاز ۴ پیاده‌سازی کرده‌اید، حروف موجود در هر بخش را تشخیص داده و در یک آرایه دوبعدی ذخیره می‌کنید. همچنین تضمین می‌شود که عکس‌ها قابل بخش‌بندی هستند. یعنی بین هر ستون و ردیف از حروف یک خط وجود دارد که هیچ حرفی را قطع نمی‌کند (شکل زیر).

R	E	I	B
B	A	R	E
O	B	S	R
R	A	S	R

شکل ۷: نمونه جدول بخش‌بندی شده ورودی

۶.۱ فاز ۶: یافتن کلمه (Find Word)

در این فاز باید در آرایه دوبعدی به‌دست آمده از فاز ۴ جستجو کنید و تعداد و مکان کلمات موجود در فایل dictio-nary.txt را یافته و نمایش دهید. فرمت فایل دیکشنری به این صورت می‌باشد که در ابتدا تعداد کلمات، و در خطوط بعدی هر کلمه در یک خط خواهد آمد. خروجی شما باید شامل یک فایل متنی باشد که در آن هر کلمه به همراه تعداد یافت شده از آن کلمه، آمده باشد؛ همچنین یک عکس خروجی که مکان کلمه‌های پیدا شده را نمایش دهد. برای این کار می‌توانید از هر روشی که می‌خواهید استفاده کنید اما باید مکان کلمات پیدا شده به طور مشخص در عکس خروجی نمایش داده شود. به مثال زیر توجه کنید.

R	E	I	B
B	A	R	E
O	B	S	R
R	A	S	R

شکل ۸: نمایش کلمه ASR در عکس خروجی

۷.۱ فاز ۷: رابط گاربری گرافیکی

در این فاز باید یک رابط گرافیکی ساده برای این پروژه طراحی کنید. توجه داشته باشید که این رابط گرافیکی می تواند بسیار ساده باشد و تنها کافی است کارهای زیر را برای کاربر فراهم سازد.

- یادگیری حروف با استفاده از دیتاست
- دریافت آدرس عکس ورودی و پردازش آن
- نمایش خروجی (کلمات یافت شده و تعداد آن ها) و ذخیره فایل متنی و عکس خروجی

۸.۱ فاز ۸: خواندن و ذخیره فایل تصویری BMP

در این فاز باید توابع readBMP و saveBMP که تا کنون از طریق کتابخانه bmpio به آن ها دسترسی داشتید را خودتان پیاده سازی کنید.

۲ نکات تکمیلی

۱.۲ نحوه پیاده سازی

برنامه باید به زبان C نوشته شود. می توانید از آخرین استاندارد ارائه شده یعنی C11 استفاده کنید. (در صورتی که از نسخه های جدید کامپایلر استفاده می کنید، احتمالاً به صورت پیش فرض کد شما با همین استاندارد کامپایل می شود.)

برای پیاده سازی رابط گرافیکی برنامه (GUI) می توانید از کتابخانه GDI+^۱ برای سیستم عامل Windows و از کتابخانه Allegro^۲ برای هر سه سیستم عامل Windows، macOS و Linux استفاده کنید.

در زمان تحویل پروژه، پروژه شما روی یک سیستم ثالث کامپایل، اجرا و تست می شود. بنابراین همه فایل های مورد نیاز برنامه را در کنار آن قرار داده و از آدرس دهی absolute برای فایل ها استفاده نکنید.

^۱ [GDI+ - Windows Application|Microsoft Docs](#)

^۲ [Allegro - A game programming library](#)

۲.۲ بارم‌بندی پروژه

نمره پروژه از ۱۰۰ واحد محاسبه می‌شود و نمره‌ی اضافه به این مقدار افزوده خواهد شد. توضیحات بیشتر در خصوص بارم هر قسمت در جدول ۱ آمده است. البته ممکن است پس از بررسی پروژه‌ها، با توجه به عملکرد دانشجویان این بارم‌بندی مقداری تغییری کند. علاوه بر این موارد، پارامترهای مربوط به سنجش کیفیت کد و برنامه‌نویسی نیز در نمره‌ی شما تأثیر قابل توجهی دارد. در طول ترم با بعضی از روش‌های کدنویسی مرتب و صحیح آشنا شده اید و لازم است آن‌ها را در زمان کدنویسی در نظر بگیرید. برای داشتن برنامه‌ای با کیفیت بالاتر، سعی کنید به‌طور بهینه از توابع استفاده کرده و حداقل دستورات را در بدنه تابع main قرار دهید؛ کد را همیشه مرتب کرده و از توضیح (comment) برای افزایش خوانایی برنامه استفاده کنید.

جدول ۱: خلاصه بارم‌بندی پروژه

عنوان	میزان نمره
کامپایل و اجرا شدن برنامه	الزامی
فاز ۱	۱۰
فاز ۲	۲۵
فاز ۳	۱۰
فاز ۴	۱۰
فاز ۵	۲۰
فاز ۶	۵
فاز ۷	۲۰
فاز ۸	+۲۰

۳.۲ نمره‌ی اضافی

همانطور که در بخش قبلی گفته شد، هر کدام از قابلیت‌های مطرح شده در ارزیابی پروژه نمره‌ی مخصوص خود را دارد. با این وجود، لیستی از قابلیت‌های اضافه برای دانشجویانی که علاقه‌مند به کسب نمره‌ی بیشتر از پروژه هستند ارائه می‌شود. نمره‌ی اضافی فقط محدود به این لیست نیست و می‌توانید قابلیت‌های دیگری که تأثیر مثبت در طراحی و اجرای برنامه دارند را با هماهنگی تیم حل‌تمرین پیاده‌سازی کنید.

استفاده از **Git**: برای کنترل بهتر سورس‌کد برنامه و نسخه‌های مختلف آن می‌توانید از برنامه Git استفاده کنید. امروزه از گیت

و برنامه‌های مشابه تقریباً در همه‌ی ابعاد صنعت گسترده نرم‌افزار استفاده می‌شود و آشنایی با آن و فراگیری نحوه استفاده از آن می‌تواند در آینده کاری شما نقش مهمی ایفا کند. جالب است بدانید سایت معروف GitHub^۳ بر پایه همین برنامه و در واقع برای خدمات‌دهی به کاربران گیت طراحی شده است.

۴.۲ برون‌سپاری

برون‌سپاری^۴ روشی است که در آن پروژه تعریف شده و سپس برای پیاده‌سازی، به افراد دیگری سپرده می‌شود. این روش در صنعت و شرکت‌های کوچک و بزرگ، موارد استفاده‌ی خاص خود را دارد و اگر به‌درستی استفاده شود، نتایج مطلوبی را به همراه خواهد داشت. اما با در نظر گرفتن اینکه پروژه فعلی مربوط به یکی از دروس تخصصی دانشگاه است، دانشجویان باید پروژه را شخصاً انجام دهند و در غیر این صورت نمره‌ی پروژه را کسب نمی‌کنند.

۵.۲ تحویل پروژه

تحویل پروژه به صورت حضوری و طبق جدول زمان‌بندی خواهد بود. جزئیات بیشتر در آینده اعلام می‌شود.

با آرزوی موفقیت؛

تیم حل تمرین مبانی کامپیوتر و برنامه‌سازی، پاییز و زمستان ۱۳۹۷

^۳ GitHub

^۴ Outsourcing - Wikipedia