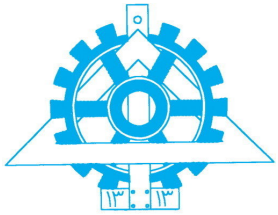
تمرین کامپیوتری سوم



دانشکده مهندسی برق و کامپیوتر

سیستم‌های عامل - پاییز ۱۴۰۱

استاد:
دکتر **مهدی کارگهی**

مهلت تحویل:
ساعت ۲۳:۵۹ روز شنبه ۲۷ آذر

مسئولان تمرین:
محمدامین باقرشاهی و **سینا نگارنده**

مقدمه

هدف از این تمرین آشنایی شما با مفاهیم اولیه **طراحی چندریسه‌ای**[1] یک مسئله است. در این تمرین شما به اعمال فیلترهایی روی تصاویر می‌پردازید. این تصاویر در فرمت 24 بیتی **بیت‌مپ**[2] هستند و کد نحوه خواندن این تصاویر به شما داده شده و شما باید اعمال فیلترها روی این تصاویر را در دو حالت سریال و موازی پیاده‌سازی کنید.

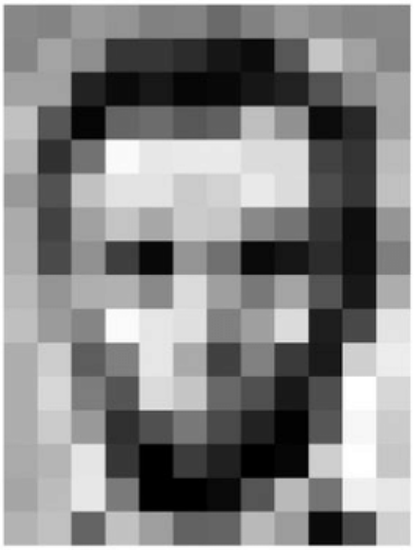
شرح تمرین

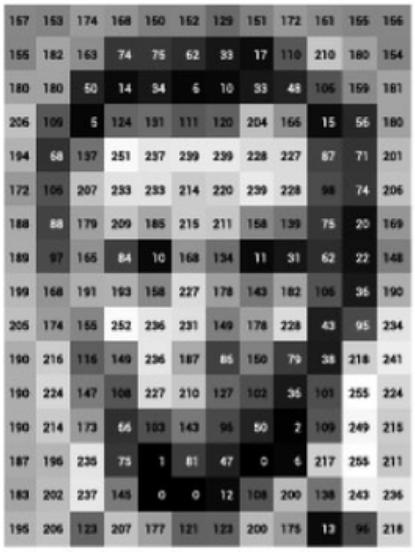
در این تمرین شما به اعمال فیلترهایی روی تصاویر می‌پردازید و پس از انجام مراحلی، نتیجه که تصویری تغییر یافته است، به دست می‌آید. در ابتدا برنامه شما اقدام به خواندن تصویر ورودی کرده و مقادیر سه کانال رنگی پیکسل‌های آن را در حافظه خود ذخیره می‌کند. پس از استخراج این اطلاعات، برنامه اقدام به اعمال مرحله به مرحله فیلترهای مورد نظر می‌کند. در این تمرین شما به دو روش این مسئله را پیاده‌سازی می‌کنید.

[1] Muti-Threaded Design
[2] Bitmap

خواندن تصاویر

کد این قسمت در فایلی با نام readImg.cpp در کنار این پرونده به شما ارائه شده است و شما باید این کد را تکمیل کنید. مقدار عددی هر پیکسل از تصویر در حالت RGB (مقادیر سه کانال رنگی قرمز، سبز و آبی) را باید در ساختمان داده‌ی دلخواه خود ذخیره کنید. از این





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

مقادیر در مراحل بعدی برای ایجاد تغییر در تصویر استفاده خواهید کرد. همچنین این مقادیر در بازه 0 تا 255 هستند.

عکس داده شده به صورت زیر است:

فیلتر آینه افقی

این فیلتر تصویر را به صورت افقی آینه می‌کند. برای اعمال این فیلتر، از رابطه زیر می‌توانید استفاده کنید.

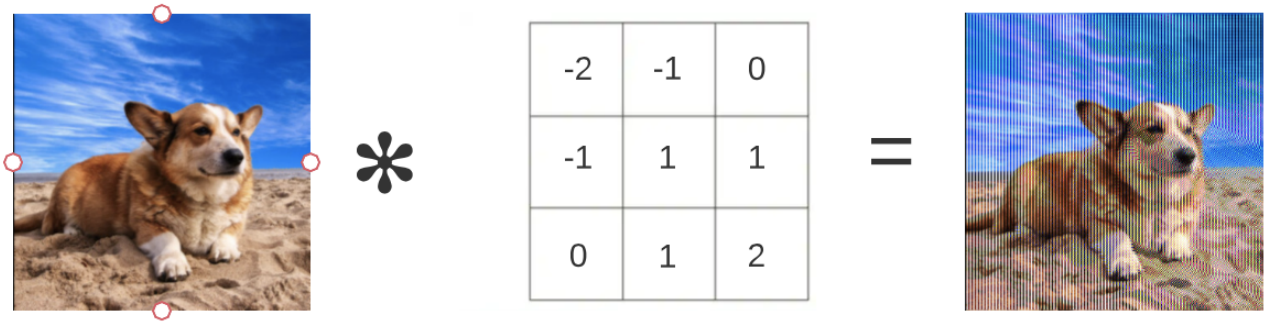
$$outputImage(x, y) = inputImage(-x, y)$$

حاصل این مرحله به شکل زیر است:

فیلتر شطرنجی

در این فیلتر هر یک از کانال های رنگی پیکسل ها در یک تصویر با مقدار کانولوشن ۹ پیکسل همسایه و خودش و کرنل زیر جایگزین می‌شود. این کار باعث ایجاد حالت شطرنجی در تصویر می‌شود. برای مثال تصویر زیر نشان دهنده یک کانال رنگی از تصویر است که در آن ورودی و خروجی فیلتر برای پیکسل مرکزی مشخص شده است.

1 2 3
4 5 6
7 8 9

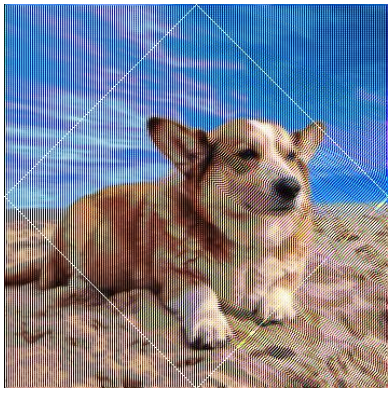


حاصل این مرحله (با انجام مراحل قبل) به شکل زیر است:

افزودن علامت لوزی روی تصویر

در این مرحله یک علامت لوزی روی تصویر ایجاد می‌شود. برای انجام این مرحله، باید چهار خط مشخص شده در تصویر زیر را رسم کنید. مقدار متناظر با رنگ سفید برابر با حداکثر مقدار ممکن برای هر پیکسل است.

حاصل این مرحله (با انجام مراحل قبل) به شکل زیر است:



پیاده‌سازی سری

در این بخش از تمرین شما به پیاده‌سازی **سری**[3] برنامه خواسته شده می‌پردازید. سعی کنید در این بخش از تمرین **بهترین پیاده‌سازی** که می‌توانید از لحاظ **زمان اجرا** انجام دهید؛ زیرا برای مقایسه عملکرد پیاده‌سازی چندریسه‌ای با سری، حالت سری باید در حالت بهینه پیاده‌سازی شده باشد. پس از این مرحله اعمالی که بیشترین زمان اجرا را به خود اختصاص داده‌اند شناسایی کنید.

[3] Serial

پیاده‌سازی چندریسه‌ای

در این بخش از تمرین به موازی‌سازی اعمال صورت گرفته در توابعی که در بخش قبل به عنوان **Hotspot**[4] از آنها یاد کردید می‌پردازید. خواندن ورودی و ذخیره‌سازی آن در حافظه از اعمال زمانگیر در بسیاری از برنامه‌هاست که احتمالا از توابع مربوط به آنها (درکنار سایر توابع) به عنوان Hotspot های برنامه یاد کرده‌اید. برای موازی‌سازی این بخش می‌توانید خواندن و ذخیره‌سازی مقادیر پیکسل‌های تصویر و اعمال فیلتر روی آنها را توسط چندین ریسه انجام دهید. بهترین ترکیب تعداد ریسه‌ها، نحوه تقسیم داده‌ها و مکانیزم‌های همگام‌سازی ریسه‌ها را باید بدست آورده و انتخاب های خود را توجیه کنید. در انتها، میزان تسریع پیاده‌سازی چندریسه‌ای به پیاده‌سازی سری را از رابطه زیر بدست آورده و طبق قالب خروجی که در ادامه آمده است، گزارش کنید:

$$speedup = \frac{serial\ execution\ time}{parallel\ execution\ time}$$

- **دقت کنید** که خروجی برنامه چندریسه‌ای شما باید عیناً مطابق با خروجی برنامه سری شما باشد.
- **توجه شود** که این بخش از تمرین باید به صورت **چندریسه‌ای** پیاده‌سازی گردد و سایر پیاده‌سازی‌ها قابل قبول **نیست**.
- **دقت شود** برای موازی‌سازی پروژه تنها مجاز به استفاده از کتابخانه **PThread** هستید و استفاده از کتابخانه های دیگر (بجز کتابخانه‌های پایه زبان C++) مجاز نیست.
- نام فایل اجرایی شما در هر دو حالت سری و موازی باید ImageFilters.out باشد.

ورودی و خروجی برنامه

برنامه شما باید نام فایل تصویر ورودی را از خط فرمان دریافت کند. نمونه اجرای برنامه با فرض اینکه تصویر ورودی با نام ut.bmp در کنار فایل اجرایی شما قرار گرفته است در زیر آمده است. خروجی گفته شده باید برای هر دو پیاده‌سازی سری و چندریسه، بعد از اجرای **هر کدام** باید به عنوان خروجی چاپ شود.

نمونه اجرا
./ImageFilters.out ut.bmp

[4] توابعی که در برنامه‌تان بیشترین زمان اجراها را به خود اختصاص می‌دهند

قالب و نمونه خروجی این اجرای برنامه در زیر آمده است که در آن باید زمان اجرای برنامه گزارش شود. همچنین علاوه بر خروجی خط فرمان، تصویر خروجی برنامه شما باید در فایلی با نام output.bmp در کنار فایل اجرایی شما نوشته شود و صحت آن بررسی می‌شود.

قالب خروجی
Execution Time: <execution_time>

نمونه خروجی
Execution Time: 2.12

توجه: نام فایل خروجی برنامه و متن قالب خروجی را از اینجا کپی نکنید و آن را تایپ کنید!

نکات تکمیلی

- تمام خروجیهای برنامه را در **جریان خروجی استاندارد**[5] چاپ کنید.
- **تضمین می‌شود** که ورودی‌هایی که به برنامه شما داده می‌شود صحیح هستند و نیازی به بررسی صحت ورودی توسط برنامه شما نیست.
- طراحی درست، **کارایی**[6] برنامه و شکستن برنامه به بخشهای کوچکتر تأثیر زیادی در نمرهی تمرین دارد.

نحوه تحویل

- **دقت کنید** که فایل آپلودی شما با نام OS_CA3_<SID>.zip حتما باید شامل **دو پوشه**[7] مجزا باشد که در یک پوشه پیاده سازی سری و در پوشه دیگر پیاده سازی موازی آورده شده است. دقت کنید که فایل zip شما شامل فولدر بیرونی نباشد و مستقیماً پس از unzip کردن آن، دو پوشه پیاده سازی سریال و موازی شما بدستآید. **تصویر ورودی و خروجی را در فایل آپلودی خود قرار ندهید.**
  - برای مثال، نمونه فایل مورد قبول در زیر آمده است:

[5] Standard Output Stream
[6] Performance
[7] Directory

```
OS_CA3_81019xxxx.zip
├── parallel
│   ├── main.cpp
│   └── makefile
└── serial
    ├── main.cpp
    └── makefile
```

- برنامهٔ شما باید در سیستم‌عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید دارای Makefile باشد. همچنین در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- نام فایل اجرایی شما که در کنار Makefile خود ساخته می‌شود باید ImageFilters.out باشد.
- نکته هایی که در جلسه توجیهی تمرین گفته می‌شود و یا در فرومهای مربوطه مطرح می‌شود بخشی از تمرین هستند؛ بنابراین به آنها توجه داشته باشید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.