سوال ۱ : برنامه‌ای بنویسید که شرایط زیر را پیاده‌سازی کند:

حلقه اصلی:

- یک LED که به PORTC.0 متصل است، به صورت پیوسته با تأخیری به اندازه ۵ کلاک پردازنده روشن و خاموش (Toggle) شود .
- این فرآیند باید در همه لحظات اجرا شود، مگر اینکه وقفه‌ای اجرا شود.

وقفه‌ها:

INT0:

- وقتی کلیدی که به INT0 متصل است فشرده شود (لبه بالارونده)، LEDهای متصل به PORTD به ترتیب یکی‌یکی روشن شوند.
- وقتی کلید رها شود (لبه پایین‌رونده)، LEDها باید به ترتیب معکوس خاموش شوند. (مثلاً اگر چهار LED وجود دارد: LED1 → LED2 → LED3 → LED4 روشن شوند و در زمان رها شدن، LED4 → LED3 → LED2 → LED1 خاموش شوند.)

INT1:

- این وقفه در هر دو لبه (بالارونده و پایین‌رونده) فعال می‌شود.
- در لبه بالارونده (فشردن کلید): همه LEDهای PORTA به صورت همزمان روشن شوند.
- در لبه پایین‌رونده (رها کردن کلید): تمام LEDهای PORTA خاموش شوند.

سوال ۲:

حلقه اصلی:

- یک عدد 8 بیتی (بین 0 تا 255) در رجیستر R16 ذخیره شده است. این عدد به صورت پیوسته هر 1 ثانیه یک واحد افزایش می‌یابد.
- وقتی مقدار به 255 رسید، دوباره از 0 شروع شود.
- مقدار فعلی عدد به طور مداوم در آدرس 0x100 ذخیره شود.

وقفه‌ها:

INT4:

- این وقفه در لبه پایین‌رونده فعال می‌شود.
- وقتی کلیدی که به پین INT0 متصل است فشرده شود:
  - مقدار ذخیره‌شده در R16 باید در حافظه SRAM در آدرس 0x101 ذخیره شود.
  - مقدار R16 به عدد 50 تنظیم شود و شمارش از 50 ادامه یابد.

INT5:

- این وقفه در لبه بالارونده فعال می‌شود.
- وقتی کلیدی که به پین INT1 متصل است فشرده شود:
  - برنامه باید مقدار فعلی رجیستر R16 را به عدد 0 تنظیم کند.
  - مقدار ذخیره‌شده در حافظه آدرس 0x100 و 0x101 باید پاک شود (به 0 تنظیم شود).

سوال ۳:

حلقه اصلی:

- از آدرس 0x2500 الی 0x4000 را بخواند و اگر تعداد بیت های ۱ آن فرد بود آن را ۲ برابر کند و در غیر این صورت آن را ۸ برابر کرده و در نهایت نتیجه را در همان ادرس ذخیره کند.
- این فرآیند باید به طور مداوم اجرا شود، مگر زمانی که یک وقفه فعال شود.

وقفه‌ها:

INT7:

- وقتی کلیدی که به پین INT0 متصل است فشرده شود (لبه پایین‌رونده)، یک شمارنده نرم‌افزاری که از 0 شروع شده است، مقدار خود را یک واحد افزایش دهد.
- وقتی کلید رها شود (لبه بالارونده)، مقدار شمارنده روی LEDهای PORTD به صورت باینری نمایش داده شود.

(مثلاً اگر شمارنده مقدار 5 را نگه دارد، وضعیت باینری 0101 روی PORTD نمایش داده شود.)

INT3:

- این وقفه در هر دو لبه فعال می‌شود.
- لبه بالارونده (فشردن کلید): همه LEDهای PORTD خاموش شوند.
- لبه پایین‌رونده (رها کردن کلید): شمارنده باید به مقدار 0 ریست شود و تمامی LEDهای PORTD نیز خاموش شوند.

سوال ۴:

از شما خواسته شده تا یک فرایند اشتراک کانال به روش کدگذاری (CDMA) مختصر را پیاده سازی کنید. رویه ی کلی این پیاده سازی به شرح زیر است:

پنج کد پیش فرض در رجیسترهای R17 الی R21 ذخیره شده که نمایانگر کد های مورد استفاده در پروسه رمزنگاری و رمزگشایی است که فرض شده این کد ها بر هم متعامد هستند.

حلقه اصلی:

فاز Encode:

- در این فاز، محتوای آدرس حافظه 0x54F0 الی 0x54F5 خوانده می‌شود.
- هر مقدار خوانده‌شده در کد هشت‌بیتی ذخیره‌شده در رجیستر کد ضرب می‌شود به فرم زیر ضرب شود:

0x54F0 : R17

0x54F1 : R18

0x54F2 : R19

0x54F3 : R20

0x54F4 : R21

- نتیجه عملیات در آدرس 0x3232 به بعد به صورت متناظر ذخیره می‌شود.

فاز Decode:

- در این فاز، مقادیر ذخیره‌شده در آدرس‌های 0x3232 به بعد خوانده می‌شود.
- این مقادیر بر کد هشت‌بیتی در رجیستر متناظر تقسیم می‌شوند.
- نتایج تقسیم در آدرس اولیه ی خود (در مرحله Encode) ذخیره می‌شوند.

حالت پیش‌فرض:

- برنامه در حالت پیش‌فرض در فاز Encode قرار دارد.

وقفه‌ها:

INT2:

- این وقفه در لبه پایین‌رونده فعال می‌شود.
- عملکرد:

1. مقدار موجود در رجیستر های کد را شیفت به راست می دهد
2. بررسی می‌شود که آیا کد جدید با کد های قبلی متعامد (Orthogonal) است یا خیر.

   کد جدید و کد قبلی متعامد هستند اگر حاصل‌ضرب داخلی (Dot Product) آنها برابر صفر باشد. (برای محاسبه، هر بیت از کدها باید بیت به بیت ضرب شده و نتیجه جمع شود.)
3. اگر کد جدید متعامد باشد، مقدار جدید در رجیستر کد متناظر ذخیره می‌شود.
4. اگر متعامد نیست، مقدار رجیستر کد متناظر تغییری نمی‌کند.

INT6:

- این وقفه در لبه بالارونده فعال می‌شود.
- عملکرد:
  - فاز حلقه اصلی بین Encode و Decode را Toggle می کند.

  (اگر در فاز Encode هستیم، به فاز Decode تغییر می‌کند و بالعکس.)

سوال ۵: برنامه ی زیر برای وقفه ۶ نوشته شده که با فشرده شدن یک کلید فعال میشود:

```
EXT_INT6_ISR:
   PUSH    R16
   IN      R16,SREG
   PUSH    R16

   SBIC    PortB , 2
   JMP     LowerEdgeBefore

HighEdgeBefore:
   SBI     PortB , 2
   JMP     EXT_INT6_ISR_END

LowerEdgeBefore:
   CBI     PortB , 2

EXT_INT6_ISR_END:
   POP     R16
   OUT     SREG , R16
   POP     R16
   RET
```

بررسی نمایید که این کد بدون مشکل اجرا خواهد شد و آیا ممکن است در آینده مشکلاتی ایجاد کند؟

موفق باشید