بسمه تعالی

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

# تمرین دوم برنامه‌نویسی تجهیزات اینترنت اشیا

استاد: دکتر علی بهلولی

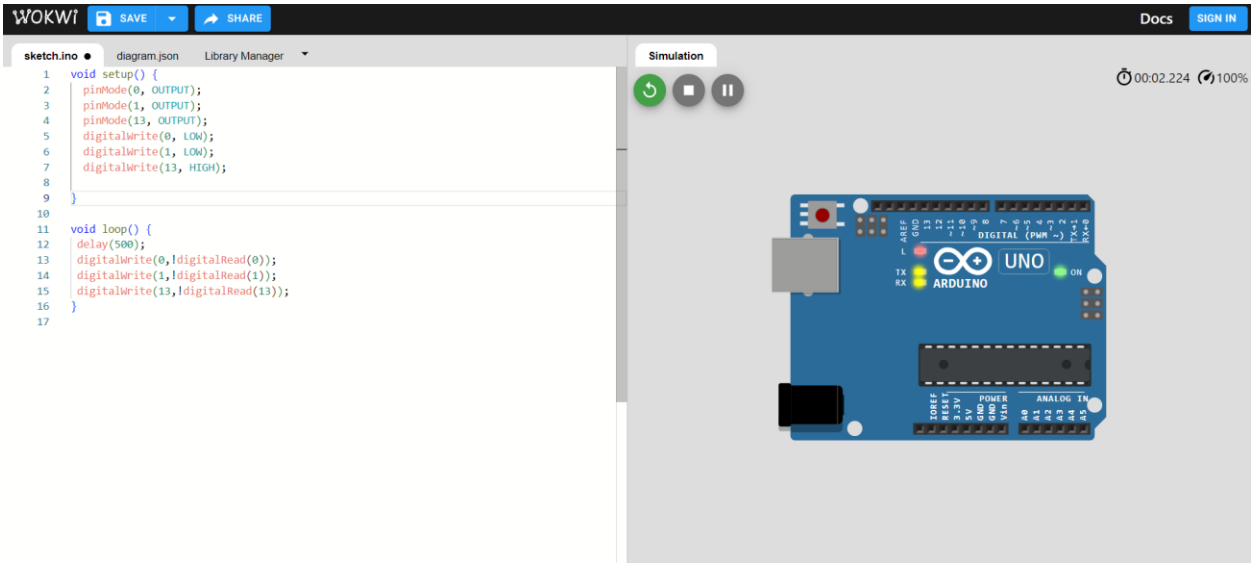
مهروالسادات نوحی

۹۹۳۶۱۳۰۶۱

بهار ۱۴۰۳

۱-برنامه ای بنویسید که سه Led روی بورد آردوینو به صورت همزمان با دوره تناوب یک ثانیه روشن و خاموش شوند. این برنامه را توسط شبیه ساز آنلاین زیر اجرا کنید. (اسکرین شات از محیط شبیه سازی را به عنوان پاسخ ارسال فرمایید.

https://wokwi.com/




```
void setup() {
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(13, OUTPUT);
  digitalWrite(0, LOW);
  digitalWrite(1, LOW);
  digitalWrite(13, HIGH);

}

void loop() {
 delay(500);
 digitalWrite(0,!digitalRead(0));
 digitalWrite(1,!digitalRead(1));
 digitalWrite(13,!digitalRead(13));
}
```

```
void setup() {
  pinMode(0, OUTPUT);
  pinMode(1, OUTPUT);
  pinMode(13, OUTPUT);

}

void loop() {
 digitalWrite(0, LOW);
 digitalWrite(1, LOW);
 digitalWrite(13, HIGH);
 delay(500);
 digitalWrite(0,HIGH);
 digitalWrite(1,HIGH);
 digitalWrite(13,LOW);
 delay(500);
}
```

هر دو کد خروجی یکسان دارند. با توجه به اینکه Led های متصل شده به پایه‌های صفر ویک، Active low هستند، برای روشن شدن باید صفر روی پایه ارسال شود و برای پایه ۱۳ باید ۱ فرستاده شود.

۲-با مطالعه مستندات این سایت، نحوه استفاده از ترمینال را (سریال مانیتور) را بیابید .

**سریال مانیتور چیست؟** سریال مانیتور یک ویژگی در محیط توسعه یکپارچه آردوینو (Arduino IDE) است که به ما امکان می‌دهد داده‌های متنی بین کامپیوتر و برد آردوینودریافت کنیم. این رابط یک ابزار قدرتمند برای دیباگ (اشکال‌زدایی) و کنترل پروژه‌های آردوینو ما می‌باشد. سریال مانیتور درواقع راهی برای ارسال/دریافت اطلاعات به/از کد آردوینو فراهم می‌کند. می‌توانیم از آن برای مشاهده پیام‌های دیباگ چاپ شده توسط برنامه یا ارسال دستوراتی که برنامه را کنترل می‌کنند، استفاده کنیم. وقتی ما کد خود را به برد آردوینو آپلود می‌کنیم، می‌توانیم از توابع ارتباط سریال که توسط زبان برنامه‌نویسی آردوینو ارائه شده‌اند برای ارسال و دریافت داده استفاده کنیم. مانیتور سریال این داده‌ها را به صورت زنده نمایش می‌دهد.

**راه‌اندازی سریال مانیتور**

در کد آردوینو، باید ارتباط سریال را با استفاده از ;Serial. Begin(baudRate) در تابع() setup تنظیم کنیم. پارامتر baudRate سرعت ارتباط را تعیین می‌کند که معمولاً روی ۹۶۰۰ بیت بر ثانیه تنظیم می‌شود.

```
void setup() {
  Serial.begin(115200); // Any baud rate should work
  Serial.println("Hello Arduino\n");
}

void loop() {
  // Do nothing...
}
```

- **ارسال داده به سریال مانیتور:** از () Serial.Print یا ()Serial.println برای ارسال داده از آردوینو به سریال مانیتور باید استفاده نمود.()Serial.Print داده را بدون کاراکتر خط جدید چاپ می‌کند، در حالی که ()Serial.println داده را با یک خط جدید چاپ می‌کند.

sketch.ino ● diagram.json ● Library Manager

```ino
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello, world!"); // چاپ "Hello, world!" با یک خط جدید
  delay(1000); // انتظار 1 ثانیه
}
```

```
Hello, world!
Hello, world!
Hello, world!
Hello, world!
Hello, world!
```

نکته: هردو آردوئینو Uno و Mega از پروتکل سریال (USART) پشتیبانی سخت افزاری دارند. سریال مانیتور به طور خودکار به پورت سریال سخت‌افزار متصل می‌شود و baud rate را تشخیص می‌دهد، بنابراین بدون هیچ‌گونه پیکربندی خاصی کار می‌کند.

نکته: می توانیم از کلاس سریال آردوینو برای تعامل با مانیتور سریال استفاده کنیم.

نکته: آردوینو مگا دارای چندین پورت سریال سخت افزاری است. می‌توانیم با پیکربندی پین‌های موجود در diagram.json، سریال مانیتور را به پورت سریال دیگری متصل کنیم. به عنوان مثال، برای اتصال Serial2 به مانیتور سریال، خطوط زیر را به بخش اتصالات در نمودار خود اضافه کنیم:

```
[ "mega:17", "$serialMonitor:TX", "" ],
[ "mega:16", "$serialMonitor:RX", "" ],
```

Mega را با شناسه واقعی قسمت wokwi-arduino-mega خود جایگزین کنیم.

توجه: باید `$serialMonitor:TX` را به پین `RX` پورت سریال و `$serialMonitor:RX` را به پین `TX` پورت سریال متصل کنیم.

می‌توانیم با افزودن بخش "SerialMonitor" به فایل diagram.json ، Serial Monitor را فعال کنیم.

پیکربندی پیش فرض به صورت زیر می‌باشد:

```
"serialMonitor": {
  "display": "auto",
  "newline": "lf",
  "convertEol": false
}
```

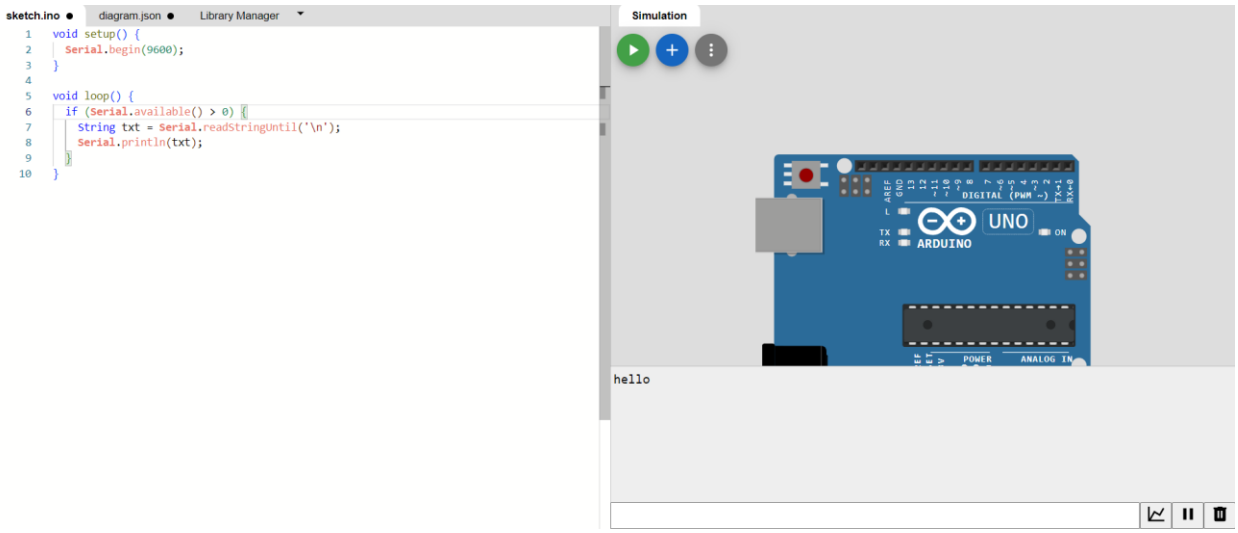

**Display#**

ویژگی display زمان/نحوه نمایش مانیتور سریال را پیکربندی می کند. مقادیر ممکن عبارتند از:

| Value | Description |
|---|---|
| auto | Display the Serial Monitor when there's some output (the default) |
| always | Always display the Serial Monitor when simulation starts |
| never | Never display the Serial Monitor |
| plotter | Display the Serial Plotter when simulation starts |
| terminal | Display a terminal (using XTerm.js) |

- **دریافت داده از سریال مانیتور**: می‌توانیم داده‌های ارسال شده از مانیتور سریال را نیز دریافت کنیم. از توابع **Serial.available()**, **Serial.read()** و توابع سریال دیگر برای مدیریت داده‌های ورودی استفاده کنیم.

هنگامی که یک خط متن را در سریال مانیتور وارد می‌کنیم، شبیه ساز آن متن را برای برنامه ما ارسال می‌کند. برنامه ما می تواند آن را با استفاده از Serial.read() و همچنین برخی از متدهای سریال دیگر بخواند.




نکته: به طور پیش‌فرض، شبیه‌ساز یک کاراکتر فید خط ("\n"، کد اسکی ۱۰) را به هر خط متنی که به برنامه ما ارسال می‌کند اضافه می‌کند. ما می توانیم از ویژگی newline برای تغییر این رفتار و پیکربندی یک دنباله متفاوت از کاراکترها استفاده کنیم:

| Value | Characters | ASCII codes | Description |
|---|---|---|---|
| lf | "\n" | 10 | Line feed (the default) |
| cr | "\r" | 13 | Carriage return |
| crlf | "\r\n" | 10 13 | Carraige return + linefeed |
| none | "" | | Don't append any characters to input lines |

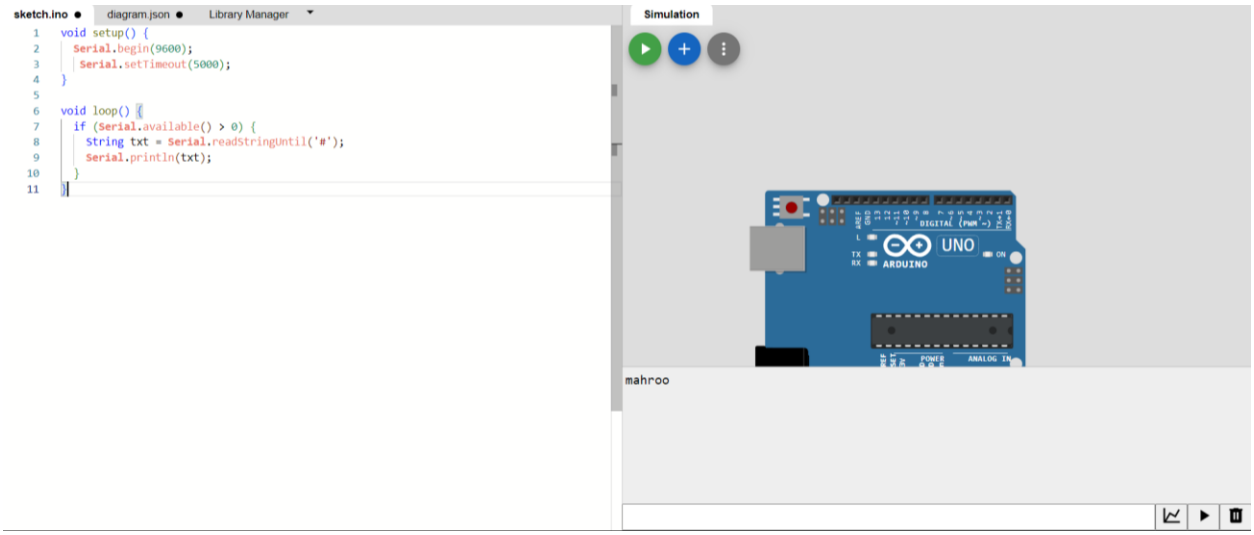
۳-سناریوهایی برای نوشتن برنامه تست جهت پاسخ به سوالات زیر در مورد تابع Serial.readStringUntil() طراحی و سپس تست کنید(اسکرین شات از اجرای برنامه ارسال نمایید)

الف) آیا این تابع از نوع Blocking است یا Non-Blocking ؟

ب) با نوشتن چند برنامه تست، نحوه عملکرد این تابع را به صورت کامل بدست آورید(فلوچارت رسم کنید)

**۳-الف)** اول بریم ببینیم این تابع چه کاری انجام می‌دهد: تابع Serial.readStringUntil() در آردوینو یک رشته ورودی را تا رسیدن به یک کاراکتر مشخص می‌خواند. یعنی این تابع تا زمانی که کاراکتر مورد نظر را در ورودی نیابد، دیگر عملیات برنامه انجام نخواهد شد و برنامه را در همان حالت متوقف نگه میدارد. به عبارت دیگر، این تابع تا دریافت کاراکتر مشخص شده، کنترل برنامه را در خود حبس میکند و برنامه را از انجام سایر عملیات باز میدارد. پس این تابع از نوع Blocking است. این تابع به صورت پیشفرض به مدت ۱۰۰۰ میلی ثانیه برای دریافت ورودی از پورت سریال بلاک می شود و پس از گذشت زمان گفته شده، ادامه برنامه اجرا خواهد شد. برای تغییر این مقدار پیشفرض، می‌توان از تابع Serial.setTimeout() برای تغییر این مدت زمان استفاده کرد و به میلی‌ثانیه، مدت زمان مورد نظر خود را در آرگومان ورودی این تابع قرار می‌دهیم.

در کد زیر، با مشخص کردن ۵۰۰۰ میلی‌ثانیه به عنوان timeout ، تابع readStringUntil به مدت ۵ ثانیه جهت دریافت ورودی، صبر می‌کند و با مشاهده کاراکتر # ، نتیجه را بر میگرداند. در صورت دریافت کاراکتر # در مدت زمان ۵ ثانیه و یا عدم دریافت ورودی، تابع از حالت Blocking خارج می‌شود و ادامه برنامه اجرا می‌شود. نکته این است که این تابع اگر ورودی را دریافت کند با حالت انتهایی # از بلاک خارج می‌شود ولی اگر دریافت نکند برنامه را به مدت تعیین شده بلاک می‌کند.



**تست بلوکینگ بودن تابع Serial.readStringUntil()**

برای بررسی نهایی اینکه آیا این تابع Blocking است، می‌توانیم از طریق مانیتور سریال دستورات "on" و "off"را وارد کنیم و مشاهده کنیم که آیا برنامه تا زمان دریافت ورودی متوقف می‌شود یا خیر.

sketch.ino ● diagram.json ● Library Manager

```
1  void setup() {
2    pinMode(LED_BUILTIN, OUTPUT);
3    Serial.begin(9600);
4    Serial.setTimeout(5000);
5  }
6
7  void loop() {
8    String str = Serial.readStringUntil('\n');
9    if (str == "on") {
10     digitalWrite(LED_BUILTIN, HIGH);
11   }
12   if (str == "off") {
13     digitalWrite(LED_BUILTIN, LOW);
14   }
15 }
```

- **اگر LED روشن و خاموش شود**: این به معنای آن است که تابع **Serial.readStringUntil()** تا دریافت ورودی منتظر می‌ماند و سپس اجرای بقیه کد را ادامه می‌دهد. بنابراین تابع از نوع Blocking است.

و درست روشن و خاموش گشت پس Blocking است.

**۳-ب)**

برای بدست آوردن نحوه عملکرد تابع **Serial.readStringUntil()**، می‌توانیم چند برنامه تست بنویسیم که شرایط مختلف را بررسی کنند.

**برنامه تست ۱: بررسی بلوکینگ (Blocking) بودن تابع**

این برنامه تست می‌کند که آیا **Serial.readStringUntil()** تا زمانی که یک کاراکتر جدید دریافت کند، بلوک می‌شود یا خیر.

sketch.ino ● diagram.json ● Library Manager

```ino
void setup() {
  Serial.begin(9600);
  Serial.println("Type something and press Enter:");
  Serial.setTimeout(5000);
  unsigned long startTime = millis();
}

void loop() {
  unsigned long startTime = millis();
  String input = Serial.readStringUntil('\n');
  unsigned long endTime = millis();
  unsigned long duration = endTime - startTime;

  Serial.print("Received: ");
  Serial.println(input);
  Serial.print("Duration: ");
  Serial.print(duration);
  Serial.println(" ms");

  while (1);
}
```

**برنامه تست ۲: بررسی رفتار تابع بدون ورودی سریال**

این برنامه تست می‌کند که آیا **Serial.readStringUntil()** بدون ورودی سریال بلوک می‌شود یا خیر.

sketch.ino ● diagram.json ● Library Manager

```ino
void setup() {
  Serial.begin(9600);
  Serial.println("Waiting for input (no input will be given):");
}

void loop() {
  unsigned long startTime = millis();
  String input = Serial.readStringUntil('\n');
  unsigned long endTime = millis();
  unsigned long duration = endTime - startTime;

  Serial.print("Received: ");
  Serial.println(input);
  Serial.print("Duration: ");
  Serial.print(duration);
  Serial.println(" ms");

  while (1);
}
```

**برنامه تست ۳: بررسی رفتار تابع با ورودی سریال با تأخیر**

این برنامه تست می‌کند که آیا **Serial.readStringUntil()** با تأخیر ورودی سریال بلوک می‌شود یا خیر.

sketch.ino ● diagram.json ● Library Manager

```
void setup() {
  Serial.begin(9600);
  Serial.println("Type something after 5 seconds:");

  delay(5000);
}

void loop() {
  unsigned long startTime = millis();
  String input = Serial.readStringUntil('\n');
  unsigned long endTime = millis();
  unsigned long duration = endTime - startTime;

  Serial.print("Received: ");
  Serial.println(input);
  Serial.print("Duration: ");
  Serial.print(duration);
  Serial.println(" ms");

  while (1);
}
```

**نتیجه‌گیری:**

با اجرای این سه برنامه، می‌توانید عملکرد تابع **Serial.readStringUntil()** را بررسی کنیم:

۱. **آیا تابع بلوک می‌شود تا ورودی دریافت کند؟**

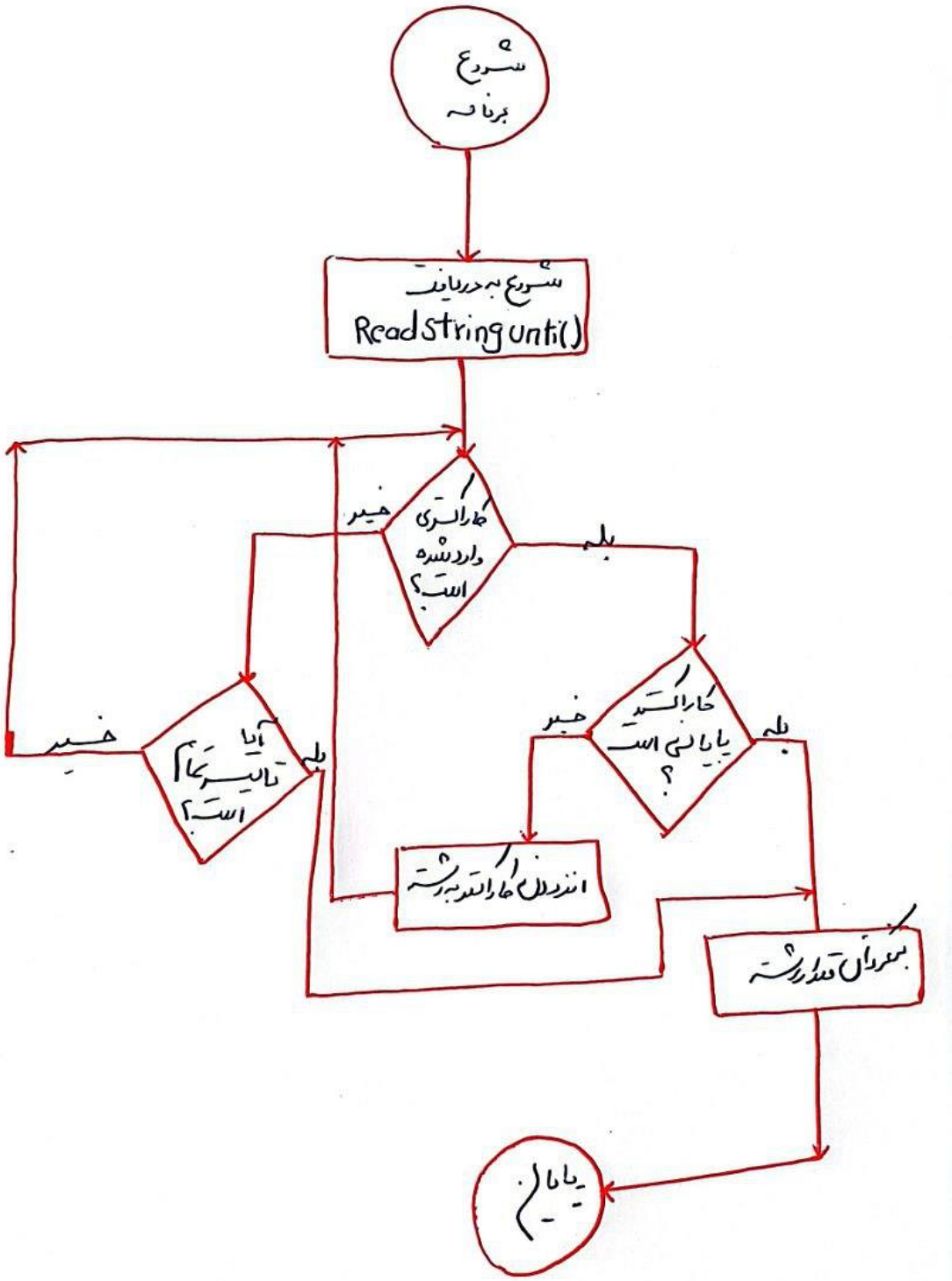
- اگر تابع تا زمانی که ورودی دریافت کند بلوک شود، مدت زمان اجرای کد افزایش می‌یابد.

۲. **رفتار تابع بدون ورودی سریال:**

- اگر ورودی سریال دریافت نشود، تابع همچنان بلوک خواهد شد و مدت زمان ست شده را سپری می‌کند.

۳. **رفتار تابع با ورودی سریال با تأخیر:**

- اگر ورودی سریال با تأخیر ارسال شود، مدت زمان اجرا باید شامل زمان تأخیر باشد.


شروع برنامه
شروع به دریافت ReadStringUntil()
کاراکتری وارد شده است؟
خیر
بله
کاراکتر یایانی است؟
خیر
بله
آیا تا الیسرتام است؟
خیر
بله
انتقال کاراکتر به پشته
برگرداندن کلمه از پشته
پایان