پروژه کامپیوتری چهارم

آرین باستانی 810100088

بخش اول:

تمرین 1_1:

```
chars = ['a' 'b' 'c' 'd' 'e' 'f' 'g' 'h' 'i'...
    'j' 'k' 'l' 'm' 'n' 'o' 'p' 'q' 'r' 's' 't' 'u' ...
    'v' 'w' 'x' 'y' 'z' ' ' '.' ',' '!' '"' ';'];

Mapset = cell(2,32);
char_id = 0:1:31;
binaryNums = dec2bin(char_id, 5);
for map_index=1:32
    Mapset{1,map_index} = chars(map_index);
    Mapset{2,map_index} = binaryNums(map_index, :);
end
```

	1	2	3
1	a	b	c
2	00000	00001	00010
3			

سوال 2_2:

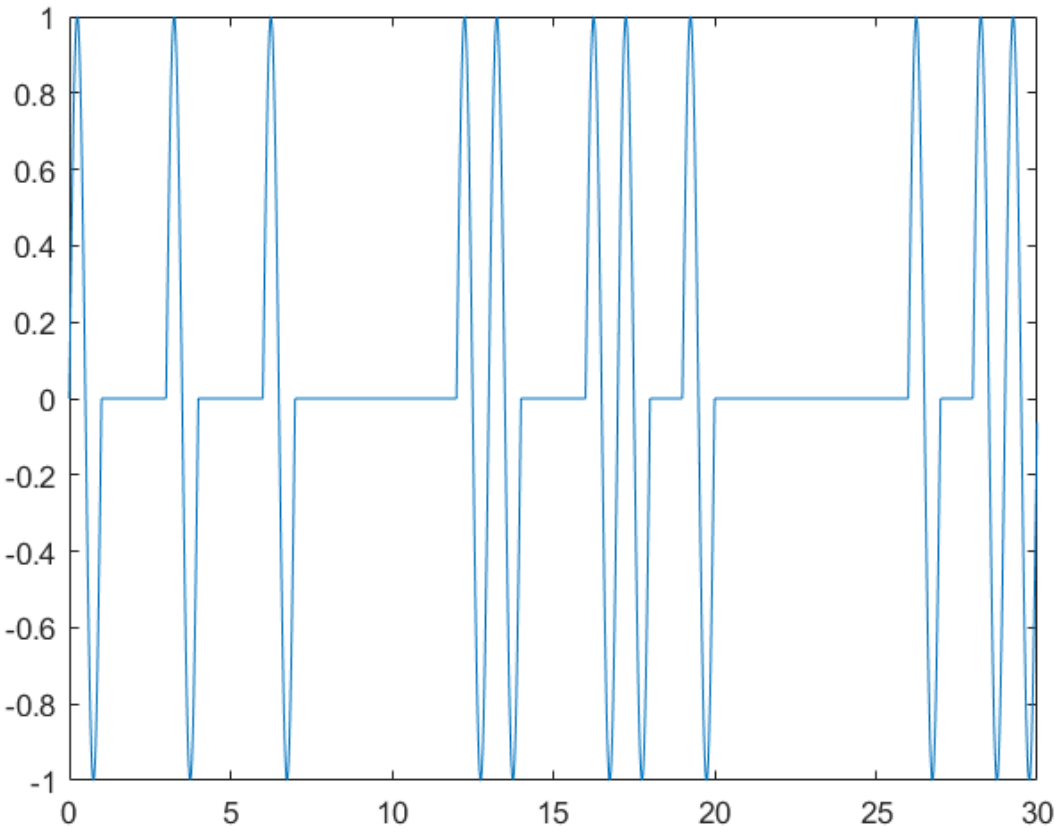
ابتدا با استفاده از دو حلقه ی تودرتو، عدد باینری هر کاراکتر در مسیج را پیدا کرده و سیو میکنیم.

سپس در یک حلقه با قدم های به اندازه سرعت خواسته شده، سیگنال متناظر با هر قسمت را با فرمول داده شده تولید کرده و به جواب اضافه میکنیم که البته باید دقت شود چون که به ازای هر کاراکتر، 5 بیت داریم بنابراین طول آرایه در 5 ضرب میشود.

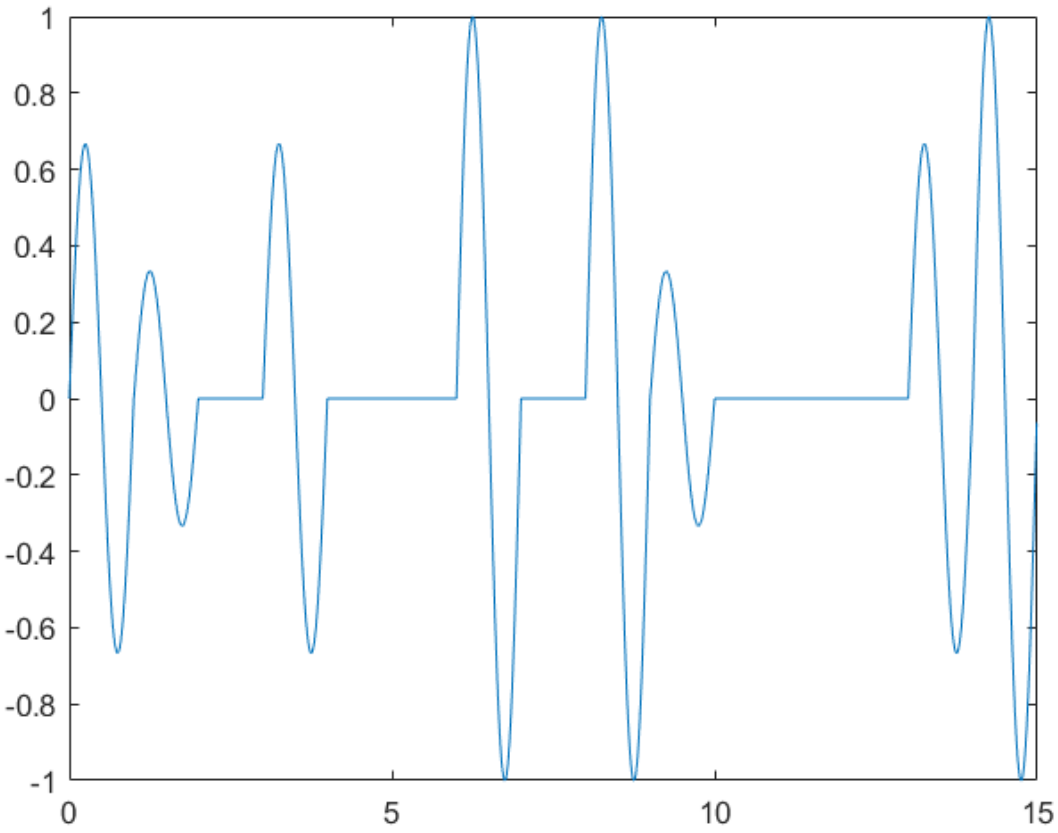
```
function codded_massg = coding_amp(massg, speed, Mapset)
    len = strlength(massg);
    step = 1 / (2 ^ speed - 1);
    first_code = [];
    codded_massg = [];
    for i = 1 : len
        for j = 1 : 32
            if Mapset{1, j} == massg(1, i)
                first_code = [first_code Mapset{2, j}];
                break;
            end
        end
    end
    t = 0 : 0.01 : 0.99;
    for i = 1 : speed : len*5
        current_code = [];
        current_code = first_code(i:i+speed-1);
        current = step * bin2dec(current_code) * sin(2 * pi * t);
        codded_massg = [codded_massg current];
    end
end
```

سوال 3_1:

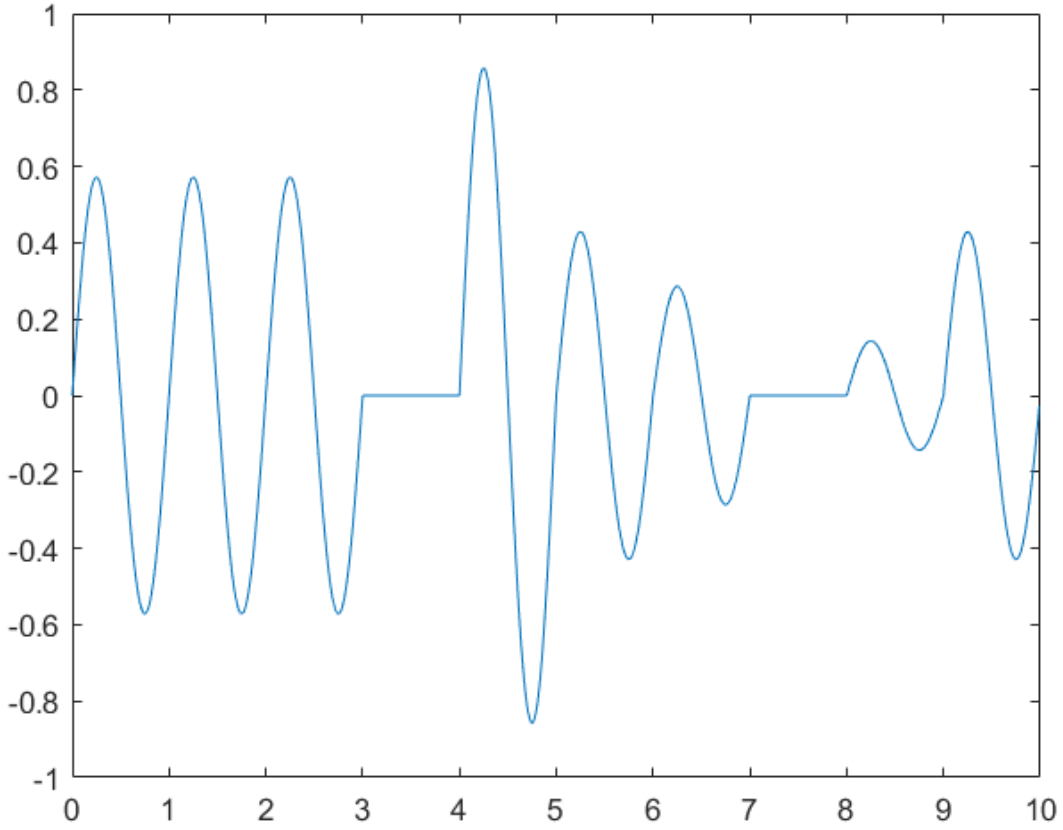
برای رشته ی ‘signal’ به ازای سرعت های مختلف داریم:



speed = 1



speed = 2



speed = 3

سوال 1_5:

ابتدا کورولیشن را حساب میکنیم. سپس یکسری مرز برای تعیین اعداد نویز دار تعیین میکنیم و برای اینکار کافیست به هر ضریب یک دوم اضافه کنیم.

سپس برای بدست آوردن ضرایب، کافیست کورولیشن هر عدد را با مرز تعیین شده مقایسه کنیم و چونکه در اعداد بصورت سورت شده حرکت میکنیم، شرط کوچکتر بودن از هر مرز برای تشخیص کافیست.

و حالا با تبدیل این ضرایب به اعداد مرتبط به کاراکترها با باینری کردن این اعداد، کد باینری مسیج را بدست اورده ایم.

حالا با استفاده از دو حلقه به راحتی میتوانیم کاراکتر مربوط به هر 5 بیت در مسیج باینری را با میپست شناسایی کرده و مسیج اصلی را بسازیم.

```
function decodedmessage = decoding_amp(coded_massg, speed, Mapset)
    all_corrs = gen_corrs(coded_massg);

    coef = 0 : 2^speed-1;
    thresh = coef + 1/2;

    massg_id = gen_massg_id(all_corrs, thresh, speed);

    decodedmessage = [];
    for i = 1 : 5 : size(massg_id, 2)
        for j = 1 : 32
            if(massg_id(i : i + 4)== Mapset{2, j})
                decodedmessage = [decodedmessage Mapset{1, j}];
            end
        end
    end
end
```

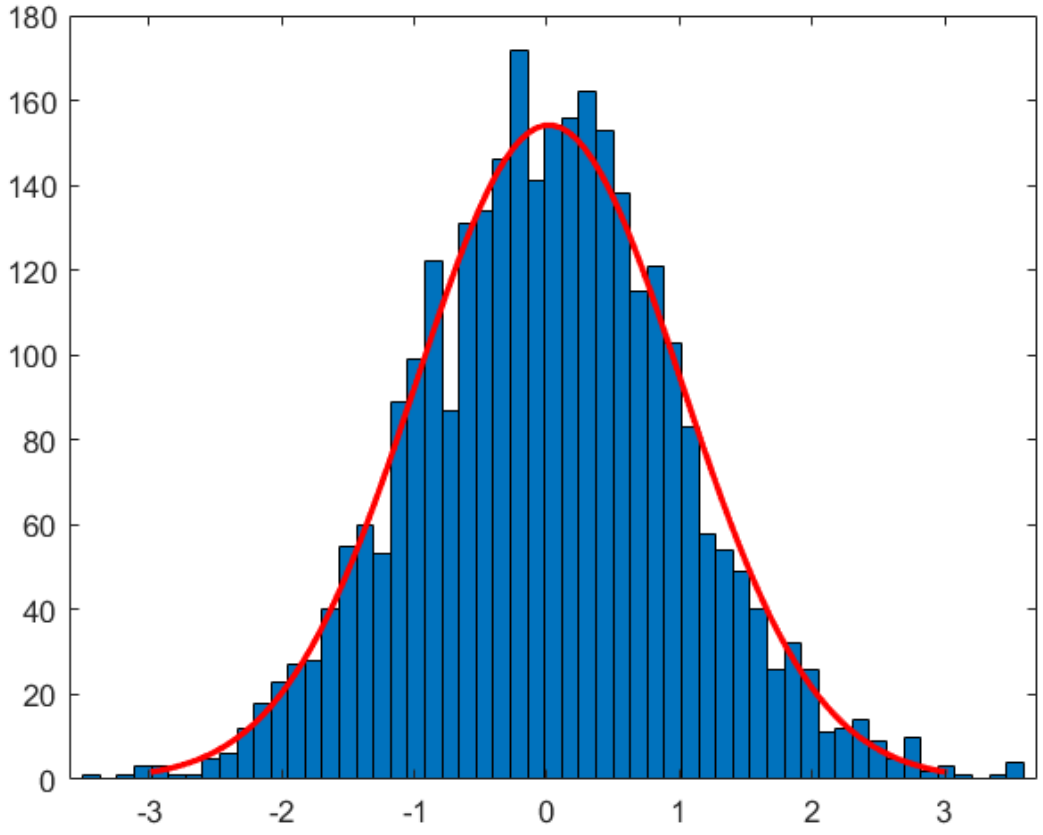
```
function all_corrs = gen_corrs(coded_massg)
    all_corrs = [];
    t = 0 : 1/100 : 0.99;
    f = sin(2 * pi * t);

    for i = 1 : 100 : size(coded_massg, 2)
        partSignal = coded_massg(i : i+99);
        corr = dot(f, partSignal);
        all_corrs = [all_corrs 2/100*corr];
    end
end

function massg_id = gen_massg_id(all_corrs, thresh, speed)
    massg_id = [];
    for i = 1 : size(all_corrs, 2)
        corr = all_corrs(i) * (2^speed - 1);
        coef = 2 ^ speed - 1;
        for j = 1 : size(thresh, 2)
            if(corr < thresh(j))
                coef = j - 1;
                break;
            end
        end
        massg_id = [massg_id dec2bin(coef, speed)];
    end
end
```

تست کد:

```
res = coding_amp('signal', 1, Mapset);
massage = decoding_amp(res, 1, Mapset);
fprintf("%s\n", massage);

res = coding_amp('signal', 2, Mapset);
massage = decoding_amp(res, 2, Mapset);
fprintf("%s\n", massage);

res = coding_amp('signal', 3, Mapset);
massage = decoding_amp(res, 3, Mapset);
fprintf("%s\n", massage);
```

```
>> p1_4
signal
signal
signal
>>
```

اول برای مشاهده ی نمودار دیتای تولید شده هیستوگرام را رسم میکنیم:



و سپس با محاسبه ی میانگین و واریانس داریم:

```
>> p1_5
y mean : 0.014180
y variance : 1.002131
```

و با توجه به اینکه میانگین گوسی برابر صفر و واریانسش برابر یک است، به خروجی مورد نظر رسیدیم.

سوال 1_6:

برای اینکه واریانس عدد های رندوم گاوسی تولید شده برابر با 0.0001 شود باید در تمام این اعداد 0.01 را ضرب کنیم چرا که ضریب به توان دو برابر تغییرات واریانس میشود و بدین صورت واریانس از 1 به 0.0001 تغییر میکند.

از آنجایی که برای مسیج با نویز در دیکود، به ازای هر کاراکتر مرز تعریف کردیم بنابراین انتظار میرود با اضافه شدن این نویز باز هم خروجی صحیح باشد:

```
res = coding_amp('signal', 1, Mapset);
res = res + 0.01 * randn(1, length(res));
massage = decoding_amp(res, 1, Mapset);
fprintf("%s\n", massage);

res = coding_amp('signal', 2, Mapset);
res = res + 0.01 * randn(1, length(res));
massage = decoding_amp(res, 2, Mapset);
fprintf("%s\n", massage);

res = coding_amp('signal', 3, Mapset);
res = res + 0.01 * randn(1, length(res));
massage = decoding_amp(res, 3, Mapset);
fprintf("%s\n", massage);
```

```
>> p1_6
signal
signal
signal
```

تمرین 1_7:

همانطور که انتظار میرفت با زیاد کردن ضریب نویز، جواب ها کم دقت تر شدند.

اما در مورد اینکه کدام یک از سرعت ها زودتر دقت خود را از دست میدهند، با توجه به نتایج حاصل شده میتوان نتیجه گرفت که با بالا بردن سرعت، دقت کاهش میابد که با توجه به توضیحات در صورت سوال این امر منطقی است:

```
for i = 0.1 : 0.2 : 2
    res = coding_amp('signal', 1, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 1, Mapset);
    fprintf("output for coef = %d is : %s\n", i, massage);
end

for i = 0.1 : 0.2 : 2
    res = coding_amp('signal', 2, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 2, Mapset);
    fprintf("output for coef = %d is : %s\n", i, massage);
end

for i = 0.1 : 0.2 : 2
    res = coding_amp('signal', 3, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 3, Mapset);
    fprintf("output for coef = %d is : %s\n", i, massage);
end
```

```
>> p1_7
output for coef = 1.000000e-01 is : signal
output for coef = 3.000000e-01 is : signal
output for coef = 5.000000e-01 is : signal
output for coef = 7.000000e-01 is : signal
output for coef = 9.000000e-01 is : signal
output for coef = 1.100000e+00 is : signal
output for coef = 1.300000e+00 is : signal
output for coef = 1.500000e+00 is : signal
output for coef = 1.700000e+00 is : tignal
output for coef = 1.900000e+00 is : signal
output for coef = 1.000000e-01 is : signal
output for coef = 3.000000e-01 is : signal
output for coef = 5.000000e-01 is : kignal
output for coef = 7.000000e-01 is : signal
output for coef = 9.000000e-01 is : signal
output for coef = 1.100000e+00 is : uigma;
output for coef = 1.300000e+00 is : signal
output for coef = 1.500000e+00 is : kim"c;
output for coef = 1.700000e+00 is : qigmil
output for coef = 1.900000e+00 is : ,mgnix
output for coef = 1.000000e-01 is : signal
output for coef = 3.000000e-01 is : signql
output for coef = 5.000000e-01 is : sygnsl
output for coef = 7.000000e-01 is : rygnae
output for coef = 9.000000e-01 is : oihiqt
output for coef = 1.100000e+00 is : sieisd
output for coef = 1.300000e+00 is : owwoee
output for coef = 1.500000e+00 is : nggism
output for coef = 1.700000e+00 is : kmffq,
output for coef = 1.900000e+00 is : ryhrfd
```

تمرین 8_1:

با قدم های 0.1 ، به مقدار ضریب انقدر اضافه میکنیم تا دیگر جواب درست ندهد، اینکار را برای بقیه ی سرعت ها نیز میکنیم:

```
 i = 0.1;
while(true)
    res = coding_amp('signal', 1, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 1, Mapset);
    if ~strcmp(massage ,'signal')
        fprintf("first error for speed = 1 in : %f\n", i);
        break;
    end
    i = i + 0.1;
end

i = 0.1;
while(true)
    res = coding_amp('signal', 2, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 2, Mapset);
    value = 'signal';
    if ~strcmp(massage ,'signal')
        fprintf("first error for speed = 2 in : %f\n", i);
        break;
    end
    i = i + 0.1;
end

i = 0.1;
while(true)
    res = coding_amp('signal', 3, Mapset);
    res = res + i * randn(1, length(res));
    massage = decoding_amp(res, 3, Mapset);
    value = 'signal';
    if ~strcmp(massage ,'signal')
        fprintf("first error for speed = 3 in : %f\n", i);
        break;
    end
    i = i + 0.1;
end
```

```
>> p1_8
first error for speed = 1 in : 1.600000
first error for speed = 2 in : 0.700000
first error for speed = 3 in : 0.400000
>>
```

بنابراین این اتفاق تقریبا برای سرعت 1 ، در 1.6 و برای سرعت 2 در 0.7 و برای سرعت 3 در 0.4 افتاد.

تمرین 1_10:

میتوان تا هرچقدر افزایش داد اما بدلیل اینکه هر عدد اعشاری را تنها در 32 بیت میتوان ذخیره کرد، با افزایش بیت ریت از جایی به بعد تمایز دوسیگنال متفاوت، غیر ممکن خواهد یود چراکه دوسیکنال بر یکدیگر منطبق میشوند.

تمرین 1_11:

هیچ تاثیری ندارد چراکه اگرچه دامنه ی سیگنال را 5 برابر کرده ایم، اما نویز هم به همران آن 5 برابر شده است.

تمرین 1_12:

سرعت اینترنت معمولی بین 8 تا 16 مگابیت بر ثانیه است. که یعنی

2^23

تا

2^24

بیت میباشد در صورتی که در این پروژه، ما تا 3 بیت بر ثانیه جلو رفتیم.

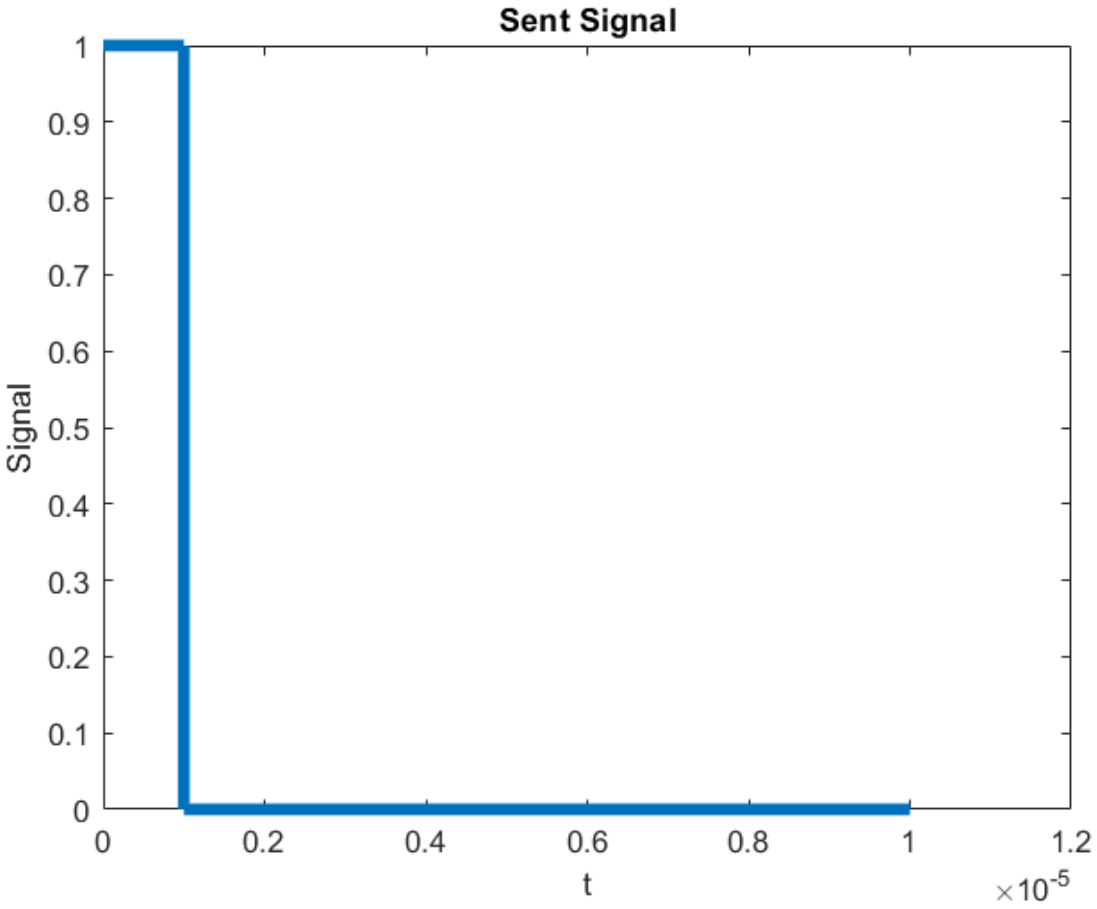
قسمت دوم:

تمرین 2_1:

همانند تمرین اول سیگنال فرستاده شده خواسته شده را میسازیم و رسم میکنیم:

```
ts = 1e-9;
the_end = 1e-5;
tau = 1e-6;

t = 0: ts: the_end;

signal = zeros(1,length(t));
signal(t <= tau) = 1;

plot(t,signal,'LineWidth',4)

title('Sent Signal');
xlabel('t')
ylabel('Signal')
```

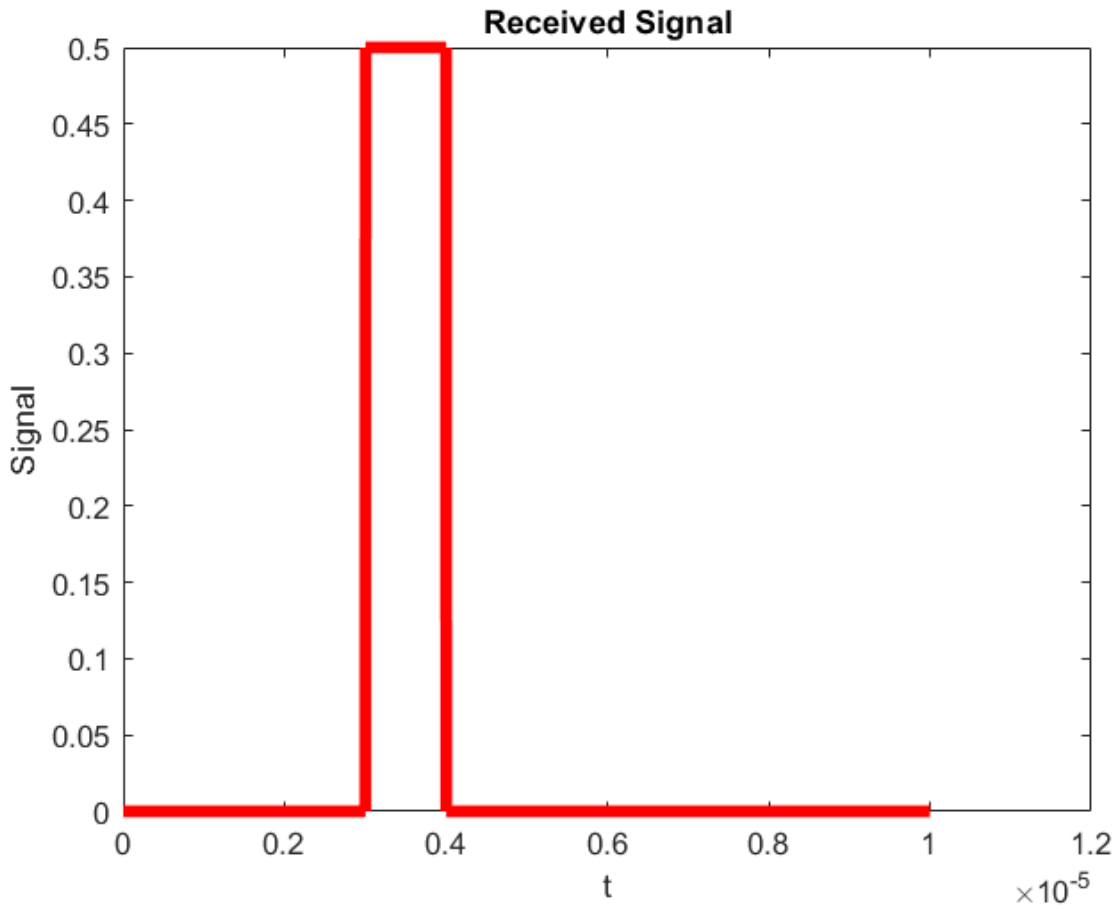
نمودار سیگنال:



کد قسمت تولید سیگنال دریافتی:

```
ts = 1e-9;
the_end = 1e-5;
tau = 1e-6;

t = 0: ts: the_end;

td = 2*450/physconst('LightSpeed');
alpha = 0.5;

signal = zeros(1, length(t));
signal(td <= t & td + tau >= t) = alpha;

plot(t,signal,'r', 'LineWidth',4)

title('Received Signal');
xlabel('t')
ylabel('Signal')
```

نمودار سیگنال:



قسمت سوم:

ابتدا ارور های احتمالی را بررسی میکنیم.

سپس مپست را لود میکنیم. سپس صدای "شماره ی" را از مپست پخش میکنیم.

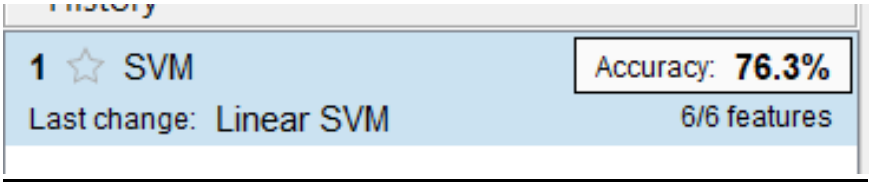
حالا یکان و دهگان شماره را محاسبه کرده و طبق اینکه یکان و دهگان غیر صفر هستند یا خیر صدای مورد نیاز را پخش میکنیم:

```
function customer_calling(number, baje)
    if((number > 99 || number < 1) ||...
            (baje > 9 || baje < 1))
        fprintf('invalid number');
        return;
    end

    sound = [];

    Mapset = set_mapset();

    call_shom(Mapset);

    first = mod(number, 10);
    secnond = floor(number / 10) * 10;

    if(secnond ~= 0)
        handle_second(Mapset)
    end

    for i = 1 : size(Mapset, 2)
        if(strcmp(Mapset{2 , i}, string(first)))
            sound = [sound Mapset{1 , i}'];
        end
    end
    for i = 1 : size(Mapset , 2)
        if(strcmp(Mapset{2 , i} , 'Baje'))
            sound = [sound Mapset{1 , i}'];
        end
    end


     for i=1:size(Mapset, 2)
         if(strcmp(Mapset{2,i}, string(baje)))
             sound = [sound Mapset{1,i}'];
         end
     end

     sound(sound , fs);
end
```

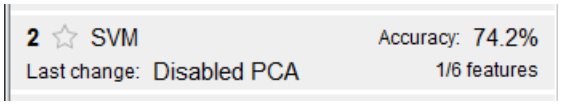
```
function Mapset = set_mapset()
    file = dir('Mapset');
    Mapset=cell(2, size(file, 1) - 2);
    fs = 0;
    for i= 3 : size(file)
        filename = fullfile('Mapset', file(i).name);
        [Mapset{1, i - 2}, fs] = audioread(filename);
        name=file(i).name;
        for j = 1 : length(name)
            if(name(j)=='.')
                break;
            end
        end
        Mapset{2, i - 2}=name(1 : j - 1);
    end
end

function handle_second(Mapset)
    sound = [];
        for i=1:size(Mapset, 2)
            if(strcmp(Mapset{2,i}, string(secnond)))
                sound = [sound Mapset{1,i}'];
            end
        end
        for i=1:size(Mapset, 2)
            if(strcmp(Mapset{2,i}, 'O'))
                sound = [sound Mapset{1,i}'];
            end
        end
end

function call_shom(Mapset)
    sound = [];
    for i = 1 : size(Mapset, 2)
        if(strcmp(Mapset{2,i}, 'Shom'))
            sound = [sound Mapset{1,i}'];
        end
    end
end
```

قسمت چهارم:

تمرین 4_1:

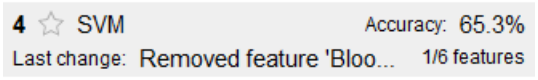


گلوکز:

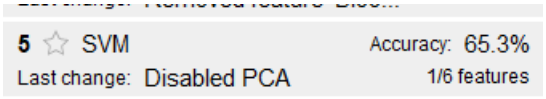


فشار خون:

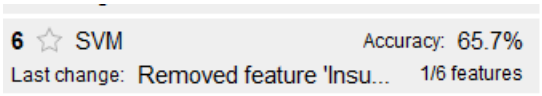
Skinthickness:



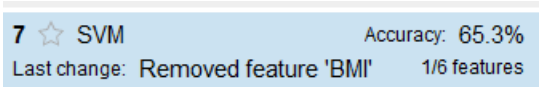
انسولین:



Bmi:



Age:



بنابراین گلوکز مهم ترین فاکتور است.