

About mid1

General rules

1. Open book & internet, computer-based exam.
2. No communications with others are allowed.

Contact with others will be regarded as Honor Code violation, and you'll get sanction by SJTU, including probation or even dismissal.

General tips

1. Don't give redundant input/output!(hw1 is a warning)

example:

write a function with prototype "function result=add(a,b)", which accept two numbers and return their numbers

correct answer:

```
function result=add(a,b)
result=a+b;
```

wrong answer 1:

```
function result=add(a,b)
result=a+b %no semicolon and will get redundant output.
```

wrong answer 2:

```
function result=add(a,b)
a=input("a="); %redundant input.
b=input("b=");
result=a+b;
disp(result);%redundant output
```

2. Easier questions may be graded only by the testcases you passed, while we'll look at your idea if the questions are hard. So, you should be considerate of easy questions. An omission for a easy question may be a lot of points off your grade!

example questions

question 1: count words(FA2021 mid1)

Write a function with prototype

```
function nums = countwords(A)
```

in which A is an array with a sentence, and nums gives the total words in the sentence. A "word" means a substring that does not contain space.

For example, countWords('I love VG101') is 3, and countWords(' A bc def g hi jkl') is 6

Cautions:

1. Don't simply calculate the numbers of space because there may be more than one space between each words.
2. There may(or may not) be space before the first word and after last word.
3. The input may be empty array ''

Idea: For a word, The character before the first letter is ' ', except that it's the first letter in the array. So, we can add ' ' (two space) to the beginning of the array, which won't change the words, but make things easier.

```
function nums = countWords(A)
add_space='  ';
A=[add_space,A];
nums=0;
size=length(A);
for ii=2:size
    if A(ii)~=' ' && A(ii-1)==' '
        nums=nums+1;
    end
end
end
```

question 2: auto2dec

Please write a script with name "auto2dec" with the following instruction:

Input an array of number from keyboard.

If the array is a normal number with no leading zero(for example, 0, 382, 921, ...), regard it as a decimal number, and please directly output it in the form of number.

If the array contains leading zero(for example, 0367, 0743, 05), please regard it as an octal number, and output its decimal representing. In this case, there'll not be '8', '9' in the input.

If the array start with '0x'(for example, 0x73, 0xa6cc), please regard it as an Hexadecimal number, and output its decimal representing. In this case, we use 'a'-'f' to represent 10-15.

Suppose there'll not be wrong input.

Your input from keyboard will not contain ''.

```
>> auto2dec
please input your number:732
732

>> auto2dec
please input your number:0261
177

>> auto2dec
please input your number:0xa1
161
```

sample answer:

```
in=input('please input your number:','s');
sz=length(in);
mode=0;%dec-1, oct-2, hex-3
if sz <= 2 % not 0x, and octal is the same as decimal
    disp(str2double(in))
else
    if in([1,2])==['0','x']
        mode=3;
    elseif in(1)=='0'
        mode=2;
    else
        mode=1;
    end
switch mode
    case 1
        disp(str2double(in));
    case 2
        result=0;
        temp=in(2:end);
        for ii=1:sz-1
            result=result+(temp(ii)-'0')*8^(sz-ii-1);
        end
        disp(result);

    case 3
        result=0;
        temp=in(3:end);
        for ii=1:sz-2
            result=result+tonum(temp(ii))*16^(sz-ii-2);
        end
        disp(result);

end
end

function result_num=tonum(char)%'0'-'9'->0-9; 'a'-'f'->10-15
if char<='9' && char>='0'
    result_num = char-'0';
```

```

else
    result_num=char-'a'+10;
end
end

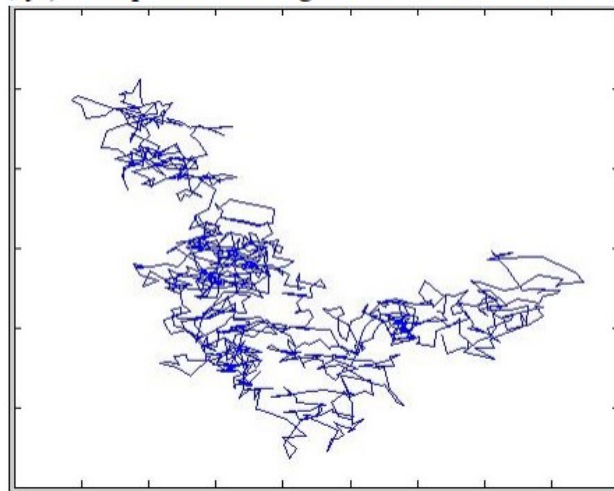
```

question 3: Two-dimensional Brownian motion simulation

5. (25 points) Two-dimensional Brownian motion simulation

Consider the two-dimensional Brownian motion of a particle starting from the origin (0, 0). At time $t = 0$, the particle is at the origin (0, 0). Suppose at time $t = n$ ($n = 0, 1, 2, \dots$), the particle is at (x_n, y_n) , then at time $t = n+1$, the location of the particle can be simulated as $(x_{n+1}, y_{n+1}) = (x_n + \Delta x, y_n + \Delta y)$, where Δx and Δy are random numbers with normal distribution (mean = 0 and standard deviation = 1). In this way, we can simulate the location of the particle at time $t = n$, given that the particle is starting from the origin (0, 0) at $t = 0$.

Your task is to simulation the Brownian motion of 1000 particles, all starting from the origin at $t = 0$. In the command window, **accept the user input of a time n and a distance r** (r is defined as $\sqrt{x^2+y^2}$) with the *input* function, find out the number of particles (M) whose distance from the origin is greater than r . **Display M in the command window.** Feel free to set the input and output messages by yourself, as long as they are clear. At the same time, plot the Brownian motion of the last particle in a figure by connecting all the coordinates (x_n, y_n) in sequence. The figure should be similar as the following:



Hint: use the built-in function *randn* to get the random step.

GOOD LUCK!