214

UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

UCSC

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Second Year Examination – Semester II – 2020/2021

SCS2211 – Laboratory II - (Part A)

TWO (2) HOURS (For Parts A and B)

To be completed by the candidate

Examination Index No: .....

Important Instructions to candidates:

1. The medium of instruction and question is **English**.
2. **Write your answers** in **English.**
3. If a page or a part of this question paper is not printed, please inform the supervisor immediately.
4. Note that questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
5. Write your index number on each and every page of the Question paper.
6. This part has **02** questions on **09** pages.
7. Answer **ALL** questions. All questions carry equal (25) marks.
8. **This paper consists of two parts, Part A (Question No 1 and Question No 2) and Part B (Question No 3 and Question No 4) which need to be submitted separately.**
9. Any electronic device capable of storing and retrieving text including electronic dictionaries and mobile phones are **not allowed**.
10. **Non-Programmable** calculators are **allowed**.

For Examiner's use only	
Question No	**Marks**
1	
2	
Total	

IndexNo:.....

Part A

Question 1

a) Use the provided spaces to answer each sub part of question a).

i) Write the output of the following code fragment.

```
myNewVariable = 37+11*i; size(myNewVariable)
     1 1
```

[1 mark]

ii) Use an Octave built-in **command** to generate the vector, in the comment, from the vector Vc.

```
Vc = [ -1-i; -1; 0; 1; 1+i]; %result:[-1 - 1i, -1, 0, 1, 1 + 1i];

     mat2str(vc)
```

[1 mark]

iii) To convert the text in variable **quote** to the commented one, fill the two blanks.

```
quote = "the quick brown fox jumped over the lazy dog";

% the quick br8wn f8x jumped 8ver the lazy d8g

quote(quote  ==   "o")   =    '8'
```

[2 marks]

iv) Write down the expected output of executing the following Octave codes.

```
a = [1 -2 3];b = [-1;2;-3;]; a.*b                          %4 marks

      -1 2 -3
      2 -4 6
      3 -6 9

a = [1 -2 3];b = [-1;2;-3;]; [a; b'](5)    3               %1 mark
```

[5 marks]

IndexNo:.....

b) A two dimensional object on an Octave plot (see Figure 1) can be transformed to a scaled up (or down) object by a linear transformation. To scale an object by a factor of *k* the following matrix *Tz* can be used. $Tz = \begin{matrix} k & 0 \\ 0 & k \end{matrix}$. Complete the code in the Code Listing 1 to dilate the given triangle by a factor of 2.

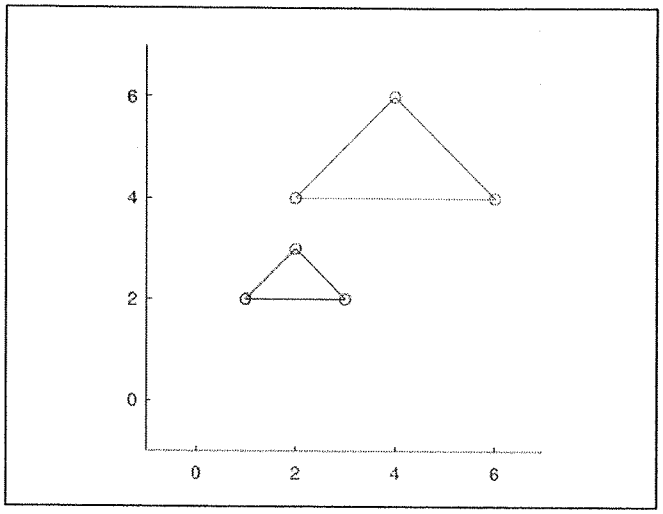

Figure 1: Dilation of triangle

Code Listing 1: Scale operation

```
triang1=[1,3,2,1;2,2,3,2];                    % Points in first triangle
Tz=[ 2 ,0;0, 2 ];                    % Transformation matrix - 1 mark
triang2=  trang1*tz ; % Transformation of the triangle - 3 marks
x1=triang2( 1 ,:); x=triang1( 1, : );     %Find the vectors - 1 mark
y1=triang2( 2, : ); y=triang1( 2 ,:);     %Find the vectors - 1 mark
 hold on
                                          % Hold the plotting - 0.5 marks
plot(x, y ,'ob-');plot( x1 ,y1,'or-')  %Plot triangles 1 & 2 - 1 mark
axis ([-1 7 -1 7] , 'equal' );

                                          % Release the hold - 0.5 marks
```

[8 marks]

c) Following recursive Octave function (Code Listing 2) takes a positive number vector as the input and finds the numbers occurring at 50%, 25% and 75%, length positions respectively. If the number of elements are even the average is taken from two numbers closest to the mid-point. Fill the indicated blanks in the code. Hint: The function **mod(A,B)** returns the remainder when A is divided by B, for positive integers A and B.

Code Listing 2: Calculating the 3 points

```
function [mid,q1,q3] = stat(vector,call=0)

  sv = sort(vector);    % Sort the vector

  call = call+1;

  l = ____length____(sv);                    %Length of the vector - 1 mark

  if(____mod(l,2)____ == 0)  %If the length of the vector is even - 2 marks

    m1 = ____(l/2)+1____;          %Position before the mid-point - 0.5 marks

    m2 = ____(l/2)-1____;           %Position after the mid-point - 0.5 marks

    mid = (sv(1,____m1____)+sv(1,____m2____))/2; %Average of the numbers - 1 mark

  else                  % Length is odd

    m1=m2 = ____sv____(l/2);%Smallest integer not less than l/2 - 1 mark

    mid = sv(1,m1);

  end

  if(call < 2)

    q1 = stat(sv(1,____1____:____m1____),call);     % Call recursively,  - 1 mark

    q3 = stat(sv(1,____1____:____m2____),call);     % on partial vectors - 1 mark

  else

    return;

  end

end % of the function
```

[8 marks]

IndexNo:.....

Question 2

a) Write answers in the given spaces. For calculations in parts (ii) & (iii) show your steps to obtain full marks.

i) Complete the mid-point based numerical integration code shown in the following listing. **k** is the lower limit, **l** is the upper limit and **n** is the number of sub-intervals.

```
function result = f_midpoint(func,k,l,n)

  result = 0;                          % Value of the definite integral

  dx = (__l__ - ____ __k__)/__n__; % Width of sub interval - 2 marks

  for t=0:n-1                          % Calculate for n rectangles
    % m1 = k + dx/2;
    % m2 = k + dx/2 + dx;

    % m3 = k + dx/2 + dx + dx;

    midp = func(__k+t*dx  +dx/2__); %Find midpoint - 2 marks

    result = result + midp;

  end % of loop

  result = result *dx;                 % Find the total area

end % of function
```

[4 marks]

ii) Figure 2 shows a graphical view of a pulsating electric current. When the pulses are smoothened the resulting current is represented by a series of + symbols which is a rough horizontal line to the time axis. Using 5 equal intervals evaluate the area under **a pulse** in the interval **[0, π]**. (Some formulae to calculate the result are already given)

```
dx = 0.628
m1 = dx/2

m2 = m1 + dx

m3 = __m2+dx__

m4 = __m3+dx__

m5 = m1 + __4__*dx                          %2.5 marks

sum = sin(m1)+...+sin(m5)                   %1 mark
```

IndexNo:.....

```
result = sum*dx                                          %0.5 marks
```

[4 marks]

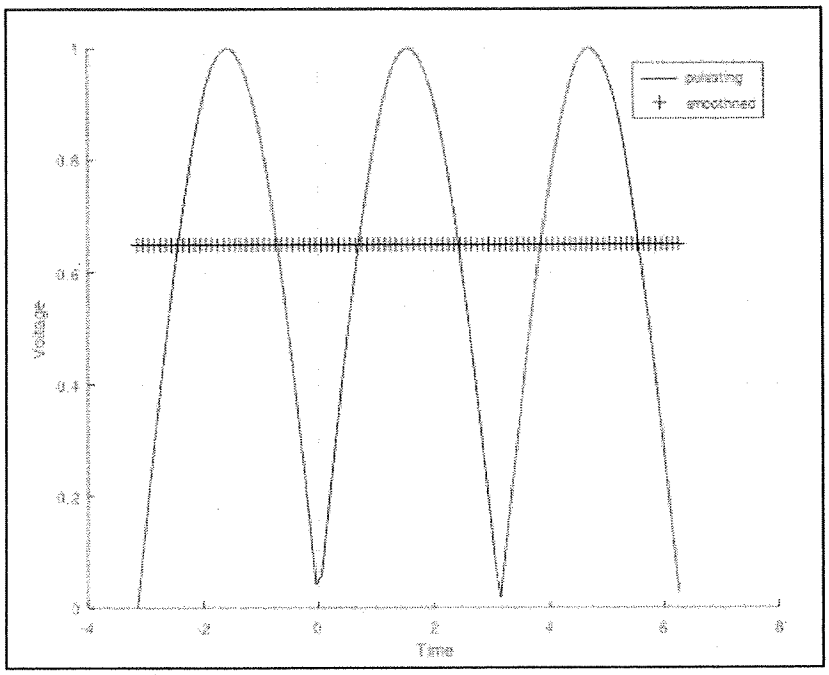

Figure 2: Sine like pulse and smooth current

iii) The relationship between the **sine like** pluses and the **smoothened** current is that the area under each pulse shall be same as the area under the line, of a width of one pulse. For a pulsating current represented by $V_l = V_p * \sin(t)$, the average voltage of a pulse is given by $V_a = V_p * C_a$, where V_l is the pulse voltage, V_p is the peak voltage and C_a is a constant which is approximately 0.63. Using **your result** of the part (ii) show how the value for C_a can be estimated. (Octave syntax is not required).

[2 marks]

b) In the following Code Listing 3, random numbers in a matrix are processed using two different methods to obtain the same result, first using a "for loop" and next via vector operations. Fill the remaining code sections to complete the non-looped version.

Code Listing 3: Looped and non-looped code

```
vR = rand(10,10); count1 = 0;
for r=1:100                                          %Start looped version
  t = vR(r);
  if(t >= 0.9 || t <= 0.1)
    count1 = count1 + 1;
  end
end                                                  %End of looped version
% Start of non-looped version
f1 = _______(_______>= 0.9);                                  % 1.5 marks
f2 = _______(_______<=0.1);                                   % 1.5 marks
count2 = ___________(f1) + __________(f2);                      % 1 mark
% End of non-looped version
```

[4 marks]

IndexNo:.....

c) Code Listing 4 Use the Figure 3 as an aid to complete the Code Listing 4. Assume the quadratic function plot is "dragged" in the Z-axis to generate a 3D surface resembling a "valley".

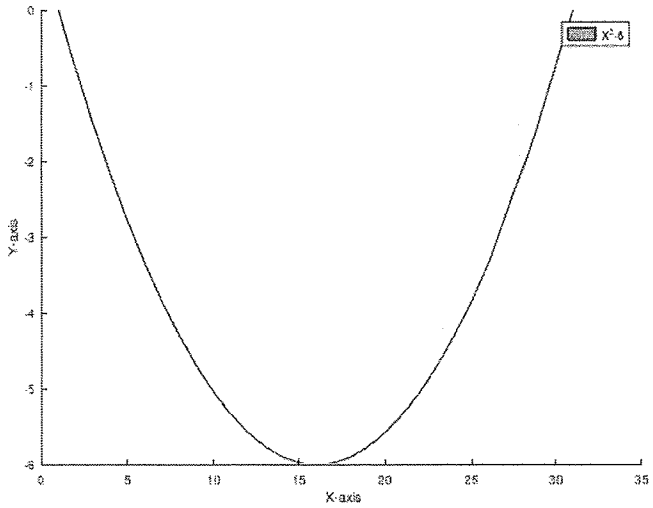

Figure 3: 2D view of the 3D object

Code Listing 4: 3D surface generator

```
function threedobject(shift = 6,points = 25)

r = sqrt(shift);          %Find r such that surface just cuts y = 0

x = linspace(-r, r, shift + points); %Make vector x - 1 mark

y = x.^2 - shift;

yz = meshgrid(y);          %Replicate y in z axis - 1 mark

colormap winter       %Coloring with winter theme - 1 mark

surf(yz);                          %Draw the surface - 1 mark

strShift = int2str(shift); %Convert shift to a string - 1 mark

lg = cstrcat('X^2-', strShift);          %Concatenate - 1 mark

legend(lg);                %Add label to the graphic - 1 mark

end % end of the function
```

[7 marks]

IndexNo:.....

d) Use the Octave **struct** definition syntax to answer the following question. Create two struct type variables called **customer1** and **customer2**. **customer1** has **name** as "Saman" and **dob** as "89-07-23"(string) while **customer2** has **name** as "Kumara" and **id** as 9007240(number).

```
customer1 = struct;
custmer2 = struct;

customer2.name = "kumara";customer2.id=9007240
customer1.name = "saman";customer1.dob="1985"
```

[2 marks]

e) Write the expected output of executing the following code.

```
cel{1,1} = "Name";
cel{1,2} = "Kamal";
cel{2,1} = "RegDate";
cel{2,2} = clock()(1:3);
cel{3,1} = "ID";
cel{3,2} = 98273;

size(cel)    3 2                                    % 1 mark

cel{4,3} = [1 2 3];

size(cel)    4 3                                    % 1 mark
```

[2 marks]

****