UEE1302(1049) F19

Midterm Examination

FULL SCORES:
120 %

EXAMINATION TIME:
18：30～21：30，total 180 minutes

INSTRUCTIONS:

Problems are classified into 3 different levels according to the difficulties: **Entry**, **Moderate** and **OMG**. You may pick arbitrary numbers of problems to solve with a total score of up to 120% where scores from the OMG level should be at least 30%. Otherwise, the max credit will be 90%.

You are allowed to open any notes or books, as well as browse on the internet to search for information. However, you are **not allowed to copy others' code**. Otherwise, you'll get 0% in this examination and be punished by the school regulations.

Read carefully the statements and requirements of each problem. Once you complete your program for one problem, please raise your hand and TA will come to your desk for examining. Please note that no credit will be given if your program fails to fully meet the requirements of each problem.

Good luck!

UEE1302(1049) F19: Midterm Examination

Demo Sheet

Student ID # : _____, Name :_____

FULL SCORES: 120 points
→ You may pick arbitrary numbers of problems to solve with a total score up to 120% where scores from the OMG level should be at least 30%. Otherwise, the max credit will be 90%.

Entry Level (10 points each)		Moderate Level (20 points each)		OMG Level (30 points each)	
No.	TA Signature	No.	TA Signature	No.	TA Signature
01		06		12	
02		07		13	
03		08		14	
04		09		15	
05		10		16	
		11			
		Score			

EXAMGFCVCQ2HSUZZSVVUMS&VF19EXAM4MMNEKPQQB%Q2T3AAWCRF19EXAMAFYY
CSQBJEGV^YYGHXF19EXAMDX5S8TU^JFWMBSK7CZDSF19EXAMGKSJPE&RWWVY
#D@#99#DF19EXAMRGR7HLH2YC#@%QPCMJQNF19EXAM2NTDSXFX2WMCH9Y*HAV
F19EXAM7NTGRU8T4#M56BJS#@HF19EXAM^Q94Q9Y^VHGN7F@4KQV@F19E
XAMBCKFSX^P*ZUHTS@%STZ8F19EXAMGJQDDQMAWUPPQAF9$MSDF19EXAMPDL7
#BN6^KKN3GEJZCE7F19EXAMURTKBCLK&@J9PF9MU854F19EXAM4JR6AD3JQ
WTYQH4ME548F19EXAMMJBVHRP8PNPWUMLZS43@F19EXAMNNAA7$GVVSFKNXK9
3QSWF19EXAMFVK*TSZYZ*VVEYYUYD5KF19EXAMYMX^JJWE2ZMZK5&EYXY8F19
EXAMCB5UYF*W3RMQCK79R9^GF19EXAMFJ9CQGNK276JURW2UGGAF19EXAMRN
2B9G56PAVYQ7VHLAXHF19EXAMKEXR5W&9VWB9FPCB9QZMF19EXAMM5$5*C#M
UGBLMUFLFRQWF19EXAMH&2UPCFRKGLTXRWRBC2AF19EXAM95BPVQRTURUUUZ#
FGM4SF19EXAM%DARNFHBBHD4Q^LXRKZBF19EXAMVH@#5WZC%C8NJ*#UPACJF1
9EXAMTDH^8PJXY&5NN@*7M5F19EXAMVUSJAT7A&SM^CZXXH2G2F19EXAM
CYF$%E4EBA98KZ5RVQKAF19EXAMHMB*AKUAX&XNWYWXHPR8F19EXAMWNZKV3X
N5XC@DBJCJQX9F19EXAMPUHDZEYS$NWDUUT&UE45F19EXAMYS&BGKEYRXDBWH
SZS9NVF19EXAMZRA9*9*MWWSVTZPVCDXUF19EXAMLTSWHSUE@U9LPHLUF2^RF

EXAMINATION STARTS FROM NEXT PAGE

（Please do not turn to the next page until TAs tell you to do so.）

19EXAMWTYKTDN8JASEMZ$G&*JFF19EXAM5CSDSYUV@EXRBCKS@AU5F19EXAMS
2T8DN3U^&PZMYF^MKXAF19EXAMQ@YYFZGYRHHHV%YATWNKF19EXAMYFDWEZ67
G&3QYKESVY3XF19EXAMAZV#UWNVHM%VKB5RKL&9F19EXAMX6QJ9QMJVPZMUZ
C73TRQF19EXAMS$$EC%URRW*DCKC@WNNFF19EXAMJ3YDFJ2TD2CDCMVSSDSHF
19EXAMQDAL5XYT#PKFSHMJAEHMF19EXAMCDFS$KJQX%X%FX3%XMWEF19EXAM
PET%RJ7JGEBH#HFF$EF^F19EXAM$HR%JSBK&EPEYF4L2#TNF19EXAM^HKAUV
#R%P9JBRERN2R%F19EXAMBJWZBRA%B7N4G$B85HTEF19EXAMY4TKFFNQ9Y$4
FUNEX6^VF19EXAMA8RC5HJSKLRGWH9NFV%RF19EXAMHWBBBEF&T&YU@XALGHZ
HF19EXAMVEGUBFHT8&FJX*Y2D^CDF19EXAMP&YVSKVFXSBEFX&EBPTUF19EXA
M*T27M5V*ETAYFY8EBFZZF19EXAMUU6ZUY@K3UVUG9XQG*@WF19EXAMUHCZ9T
$2WZWV4NWCYA^XF19EXAMCHAYKQ2HWZMR7ME7&VTFF19EXAMLZQKDFRMVV#KA
PYQ86*UF19EXAMSHDHMQJXSWE&JBFDS#$&F19EXAMVJ*&W5YE%E%VW7WQCGGK
F19EXAMND^F8CMAL38KZVKRCVDZF19EXAM8HY584KU7G2PZWA6TCQ5F19EXA
M@WYJKL3HGXNXWSKE$U@9F19EXAMEC4FMFZMW4Y7WVCRNST#F19EXAMMRE78
V%Y3DQXBWNRFFFHF19EXAMSMSFJ$UAN^VPHC8NZD3SF19EXAMT4ZYESFQXFX7
8R^HNAV$F19EXAMW^YPB^KWYH8ZWAHFVE9MF19EXAMW^MHETPDN&$HUSCFKV

【ENTRY LEVEL】

PROBLEM 01 (10 points)

Input a character in range A~Z (or a~z), and you need to output the corresponding number 1~26.

```
(Ex1)   >./pg01
        A ↵
        1
        >
(Ex2)   >./pg01
        w ↵
        23
        >
(Ex3)   >./pg01
        z ↵
        26
        >
```

PROBLEM 02 (10 points)

Please input a where a > 0. Calculate the sum of every digit of a positive integer.

7543824

7+5+4+3+8+2+4 = 33

```
(Ex1)   >./pg02
        Enter one number: 7543824 ↵
        Sum of every digit: 33
        >
(Ex2)   >./pg02
        Enter one number: 3787483 ↵
        Sum of every digit: 40
        >
(Ex3)   >./pg02
        Enter one number: 87333 ↵
        Sum of every digit: 24
        >
```

PROBLEM 03 (10 points)

For a 2x2 Matrix A,

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The inverse matrix is,

$$A^{-1} = \frac{1}{det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Where determine A is,

$$det(A) = ad - bc$$

Please write a program to calculate the determine and the inverse matrix of an input matrix A.
Input a, b, c and d will be an integer.
You **don't need** to set the output precision.

```
(Ex1)   >./pg03
        Input a, b, c, d for matrix A: 1 2 3 4 ↵
        det(A) = -2
        inverse(A)=
        -2 1
        1.5 -0.5
        >

(Ex2)   >./pg03
        Input a, b, c, d for matrix A: 1 0 0 1 ↵
        det(A) = 1
        inverse(A)=
        1 0
        0 1

(Ex3)   >./pg03
        Input a, b, c, d for matrix A: 1 3 1 3 ↵
        det(A) = 0
        A is not invertible.
        >
```

PROBLEM 04 (10 points)

Please print the same figure as follow.

```
(Ex1)    >./pg04
              *
             ***
            *********
            *axwvuts*
            *b*****r*
           **c*****q**
          ***d*****p***
           **e*****o**
            *f*****n*
            *ghijklm*
            *********
               ***
                *
```

PROBLEM 05 (10 points)

Please print out the same figure as follow.

```
(Ex1)  >./pg05

       * * * * * * * * * * * * *
       * I                     *
       *   LOVE                *
       *         C++           *
       *              IN       *
       *                 YMCTU.*
       * * * * * * * * * * * * *
```

【MODERATE LEVEL】

PROBLEM 06 (20 points)

If someone buys something with price M and gives N dollars, and you need to make change with fewest coins. But if money is not enough you should say "The money is not enough."

Input: Two integers M, N.

Output: Number of coins.

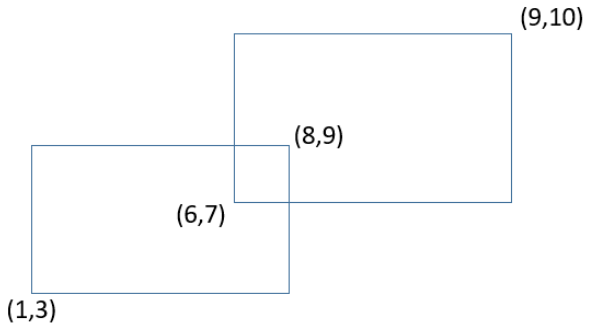
Note: Coins contain 50, 10, 5, 1.

```
(Ex1)    >./pg06
         100 1000 ↵
         50: 18  10: 0   5: 0    1: 0
         >
(Ex2)    >./pg06
         125 50 ↵
         The money is not enough.
         >
(Ex3)    >./pg06
         199 500 ↵
         50: 6   10: 0   5: 0    1: 1
         >
```

PROBLEM 07 (20 points)

Give you two points of the two rectangles, please determine two rectangle overlapping area.

Input is eight numbers. First four number is the x, y of two point of the same rectangle. Last four number is the x, y of two point of the other rectangle. The sides of rectangle are parallel to the x-axis or y-axis.



```
(Ex1)    >./pg07
         Enter four point: 1 3 8 9 6 7 9 10 ↵
         Overlapping area: 4
         >
(Ex2)    >./pg07
         Enter four point: 7 9 -1 3 5 1 10 8 ↵
         Overlapping area: 10
         >
(Ex3)    >./pg07
         Enter four point: 3 -1 7 9 5 11 -1 2 ↵
         Overlapping area: 14
         >
```

PROBLEM 08 (20 points)

ASCII, American Standard Code for Information Interchange, is a character encoding standard for electronic devices.

The figure below is the ascii table.

ASCII (1977/1986)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_ 0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
1_ 16	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
2_ 32	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3_ 48	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4_ 64	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5_ 80	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6_ 96	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7_ 112	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F

Please write a program to print out the ascii code table with specified range and columns for each row. Please use **'\t'** to separate 2 characters.

```
(Ex)    >./pg08
        From: 40 ↵
        To: 50 ↵
        Columns: 4 ↵
        40:( 41:)   42:*   43:+
        44:, 45:-   46:.   47:/
        48:0 49:1   50:2
        From: 70 ↵
        To: 92 ↵
        Columns: 4 ↵
        70:F 71:G   72:H   73:I   74:J   75:K
        76:L 77:M   78:N   79:O   80:P   81:Q
        82:R 83:S   84:T   85:U   86:V   87:W
        88:X 89:Y   90:Z   91:[   92:\
        From:-1 ↵
        >
```

PROBLEM 09 (20 points)

List all divisors of a number with the range from 1 to 2147483647. And don't show divisors which have the same digit in itself. Also show the number of remain divisors. For example, if the user inputs "31232", all divisors are 1 2 4 8 16 32 61 64 122 128 244 256 488 512 976 1952 3904 7808 15616 31232. But `Ex1` doesn't show 122 because there two "2" in "122". Then, 244, 488, 7808, 15616, 31232 don't show, too.

```
(Ex1)   >./pg09
        31232 ↵
        1 2 4 8 16 32 61 64 128 256 512 976 1952 3904
        Total = 14
        >
(Ex2)   >./pg09
        21093171 ↵
        1 3 967 1983 2901 10637 639187
        Total = 7
        >
(Ex3)   >./pg09
        2147483422 ↵
        1 2 7 14 275389
        Total = 5
        >
```

PROBLEM 10 (20 points)

Please write a program to simulate the game of "Snake".

1. Let user enter two integer numbers for the width (row) and length (column) of the map.
2. The program random generate a location of target ($) and the initial location of player (*) within the map.
3. Print the map to the console.
4. Let user enter the next move they want to take (w(up), a(left), s(down), d(right)).
5. The game will be terminated if the player gets the target, else the program will print out the map again and let user to take his/her next move.
6. If the next move of play is out of the map, the move will not be executed. The program prints out the map without the move and let play choose again.
7. Note: The initial locations of target and player is not the same.

```
(Ex1)    >./pg10
         3 4 ↵
         - - $ -
         - - - -
         - - - *
         Next move: d ↵
         - - $ -
         - - - -
         - - - *
         Next move: w ↵
         - - $ -
         - - - *
         - - - -
         Next move: a ↵
         - - $ -
         - - * -
         - - - -
         Next move: w ↵
         Game Over!
```

PROBLEM 11 (20 points)

Input two numbers by user and get the multiplication of them. The range of input number is from -2147483648 to 2147483647. Show every step of calculation as follows. You should divide the second number into different unit digit, tens digit, hundreds digit, etc. If the certain digit is "zero" in second number, don't show that step of calculation. Please notice that the output of multiplication possibly exceeds 2147483647 or below -2147483648.

```
(Ex1)   >./pg11
        321 4248 ↵
        321*4000+
        321*200+
        321*40+
        321*8=
        1363608
        >

(Ex2)   >./pg11
        -230 -3021 ↵
        (-230)*(-3000)+
        (-230)*(-20)+
        (-230)*(-1)=
        694830
        >

(Ex3)   >./pg11
        -234324 43204023 ↵
        (-234324)*40000000+
        (-234324)*3000000+
        (-234324)*200000+
        (-234324)*4000+
        (-234324)*20+
        (-234324)*3=
        -10123739485452
        >
```

【OMG LEVEL】

PROBLEM 12 (30 points)

Given a series of vertex (r, θ) in polar coordination system, which create a polygon (you only need to consider Convex polygon), calculate the area of the polygon, the input will be terminate when r is negative.

Note: the order of the vertex will be given with increasing theta. For example: (1, 45), (1, 135), (1, 225), (1 315), (-1, 0). And #define PI 3.14159.

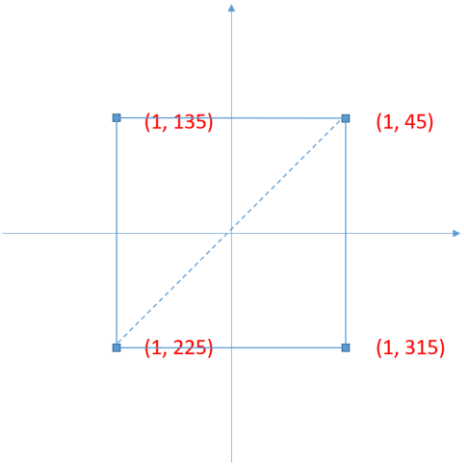
(r, θ) can be floating point number.

Hint: Heron's formula.

a, b, c are side lengths of a triangle. $s = \frac{a+b+c}{2}$, $Area = \sqrt{s(s-a)(s-b)(s-c)}$.

A rectangle can be divided into two triangles.

Then, you can use Heron's formula.




```
(Ex1)    >./pg12
         1 45 ↵
         1 135 ↵
         1 225 ↵
         1 315 ↵
         -1 0 ↵
         2
         >
```

```
(Ex2)    >./pg12
         0 0 ↵
         3 90 ↵
         5 143 ↵
         -1 0 ↵
         5.98976
         >

(Ex3)    >./pg12
         2 0 ↵
         2 60 ↵
         2 120 ↵
         2 180 ↵
         2 240 ↵
         2 300 ↵
         -1 0 ↵
         10.3923
         >
```

PROBLEM 13 (30 points)

According to continued fraction, we can approximately exp by of order n. The number of recursions, that is, the value of n is determined by **X**.

$$e^{\frac{x}{y}} = 1 + \cfrac{2x}{2y - x + \cfrac{x^2}{6y + \cfrac{x^2}{10y + \cfrac{x^2}{14y + \ddots}}}}$$

Ex:

if x = 1,y = 1, $e^{x/y} = 1 + \cfrac{2x}{2y - x} = 3.0000000$

if x = 2,y = 1, $e^{x/y} = 1 + \cfrac{2x}{2y - x + \cfrac{x^2}{6y}} = 7.0000000$

if x = 2,y = 5, $e^{x/y} = 1 + \cfrac{2x}{2y - x + \cfrac{x^2}{6y}} = 1.4918033$

if x = 3,y = 7, $e^{x/y} = 1 + \cfrac{2x}{2y - x + \cfrac{x^2}{6y + \cfrac{x^2}{10y}}} = 1.5350630$

Please write a program to ask the user to enter **X, Y** and then output the approximate value of e with 7 decimal place. You must write a recursive function to compute the fractional part.

```
(Ex1)    >./pg13
         Enter x, y: 2 5 ↵
         e(2,5) = 1.4918033
         >
(Ex2)    >./pg13
         Enter x, y: 3 7 ↵
         e(3,7) = 1.5350630
         >
(Ex3)    >./pg13
         Enter x, y: 5 9 ↵
         e(5,9) = 1.7429090
         >
```

PROBLEM 14 (30 points)

In digital world, integers are presented in binary form, and negative integers are represented by 2's complement.

How to calculate 2's complement? Take 5 as an example in a 7-bit system.

1. 5 = 0000101
2. Inverse all the bits -> 1111010, also known as 1's complement
3. Add 1 -> 1111011
4. You will get the 2's complement of 5, in decimal: -5.

For example, in a 7-bit system. 10 is 0001010, and the 2's complement of 0001010 is 1110110. The highest bit is called the sign bit. And the range will be $-2^6 \sim 2^6-1$.
You can check your 2's complement conversion in the following website.
Remember to change the number of bits to 7.
https://ppt.cc/fhjl6x

What is overflow?
When adding or subtracting 2 integers, the result might exceed the range. For example, when adding 0111111(63) and 0000010(2). The result is 1000001(-63) but the correct result should be 65. An overflow has occurred.

You can check more information from Wikipedia.
https://ppt.cc/fRHdLx
Please write a program to compute addition or subtraction for two 7-bit binary integers. You should also show an "Overflow" warning when it occurs.
Hint: The front zeros will be truncated. You can use int to save the value.

```
(Ex1)   >./pg14
        Operator(+/-): + ↵
        Input a in 7-bit binary form: 11011 ↵
        Input b in 7-bit binary form: 1000011 ↵
        11011(27) + 1000011(-61) = 1011110(-34)
        >

(Ex2)   >./pg14
        Operator(+/-): - ↵
        Input a in 7-bit binary form: 10111 ↵
        Input b in 7-bit binary form: 1011 ↵
        10111(23) - 1011(11) = 1100(12)
        >

(Ex3)   >./pg14
        Operator(+/-): + ↵
        Input a in 7-bit binary form: 110000 ↵
        Input b in 7-bit binary form: 11111 ↵
        110000(48) + 11111(31) = 1001111(-49)
        Overflow
        >
```

PROBLEM 15 (30 points)
Please write a program to perform the computation of transfer matrix with the initial state and transfer time given by the user. Every number should be handled as a fraction number (numerator and denominator). The result should be presented as simplest fraction as well (4/6 (X), 2/3(O)).
Hint: You should handle the arithmetic of fraction number. If it is a negative fraction, the negative sign should be followed with denominator.

Transfer Matrix: $\begin{bmatrix} \frac{0}{1} & \frac{2}{3} & \frac{2}{3} \\ \frac{1}{2} & \frac{0}{1} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{0}{1} \end{bmatrix}$ (Fix condition, just encode the number in your program)

Initial State: 1 1 1 0 1 0 (stands for $\begin{bmatrix}\frac{1}{1}\\ \frac{0}{1}\\ \frac{0}{1}\end{bmatrix}$, every number is paired with numerator(first) and denominator(second))

Transfer Time: 4 (means doing the transfer process for 4 times)

Output: $\begin{bmatrix}\frac{14}{27}\\ \frac{13}{54}\\ \frac{13}{54}\end{bmatrix}$ (the state after transferring for 4 times)

```
(Ex1)   >./pg15
        1 1 1 0 1 0 ↵
        4 ↵
        [14/27, 13/54, 13/54]
        >

(Ex2)   >./pg15
        1 0 1 0 1 1 ↵
        3 ↵
        [14/27, 7/27, 2/9]
        >

(Ex3)   >./pg15
        2 1 3 1 2 1 ↵
        5 ↵
        [392/729, 43/108, 1159/2916]
        >
```

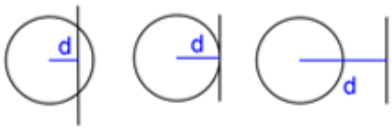
PROBLEM 16 (30 points)

Please write a program to compute the relation between a circle (**C**: $ax^2 + by^2 + cx + dy + e = 0$) and a line (**L** :$ax + by + c = 0$). The program first judge if the given equation **C** is a circle.

If **C** is a circle, print out the standard formulation of it ($(x-h)^2 + (y-k)^2 = r^2$). And print out the relationship between the given circle **C** and line **L** (**0 intersection**, **1 intersection**, **2 intersections**). The figure below shows the three relationships.

If **C** is a point, print out the coordinate of the point and the distance (to round off the 2nd decimal place) between the point and the line **L**.

If **C** is meaningless, print "meaningless".



```
(Ex1)     >./pg16
          Circle: 1 1 -2 -2 -23 ↵
          Line: 1 7 -30 ↵
          It's a circle: (x-1)^2 + (y-1)^2 = 25
          2 intersections.
          >
(Ex2)     >./pg16
          Circle: 1 1 -2 -2 -23 ↵
          Line: 4 3 -32 ↵
          It's a circle: (x-1)^2 + (y-1)^2 = 25
          1 intersection.
          >
(Ex3)     >./pg16
          Circle: 1 1 -2 -2 -23 ↵
          Line: 1 7 -100 ↵
          It's a circle: (x-1)^2 + (y-1)^2 = 25
          0 intersection.
          >
(Ex4)     >./pg16
          Circle: 1 1 -2 -4 -5 ↵
          Line: 1 1 1 ↵
          It's a point: (1,2)
          d = 2.83
          >
(Ex5)     >./pg16
          Circle: 1 1 8 -4 21 ↵
          Line: 1 1 1 ↵
          It's meaningless.
          >
```