Autumn 2023, ISTM, Purdue-FCU 2+2 ECE Program
ISTM116 Programming Applications for Engineers, Final Exam

Use file name fexam_DXXXXXXX_1.c for Question 1, file name fexam_DXXXXXXX_2.c, and file name fexam_DXXXXXXX_3.c for Question 3 of your solutions, where DXXXXXXX is your student ID. When you finish a question, submit the above files to the instructor's computer.

1. (30 points) You may start with program skeleton **fexam_skeleton_1.c** and change the file name to **fexam_DXXXXXXX_1.c**. Let A be an m×n banded matrix with lower bandwidth r and upper bandwidth s and B be the transposed matrix of A. Note that matrix B is an n×m banded matrix with lower bandwidth s and upper bandwidth r. Write a C program to perform the following steps:
   a. Enter four positive integers m, n, r, and s that, respectively, specify the matrix size, the lower bandwidth, and the upper bandwidth of matrix A, where 1≤m,n≤20.
   b. Use dynamic memory allocation to create exact memory space for the non-zero banded elements of matrices A and B and then randomly generate values of the non-zero elements of matrix A such that the values generated are between 0 and 99 (including).
   c. Perform matrix transposition to set matrix B to be the transposed matrix of A. You may write the matrix transposition code in the main program directly.
   d. Output matrix A and B, but fill in the lower off-band elements using space characters.
   e. Release memory space of matrix elements of A and B.

   Program execution example:

```
>>>>> Enter matrix size of matrix A, m and n (between 1 and 20, including): 15 12

>>>>> Enter the lower and the upper bandwidth of matrix A, r and s: 8 6

Matrix A, 15X12 with lower bandwidth 8 and upper bandwidth 6:
     92  14  44  51  22  35  75
     42  64  98  81  55  48  11   5
     50  29  84  58  76  35  15  25  51
     58  95   0  40  41  93  75  14  15   2
     90  93  14  64  28  29   2  24  82  28  96
     65  15  65  19  10  10  98  79  80  29  73   5
      0  93  30  90  20   9  76  51  64  64  37  39
     35  49  35  63  85  69  81   3  19  87  21  14
     57  26  15  88  52  53  45  86  96  11  44  46
         27  50  20  83  27  40  48  95  52  46  16
             31  86  85   2  91   2  34  50  86  66
                 10  33  72  38  33  93  58  52  46
                     95  94  95  76  24  41  76  92
                         40  64  29  79  64  30  86
                             38  99  34  13  42  76

Matrix B, 12X15 with lower bandwidth 6 and upper bandwidth 8:
     92  42  50  58  90  65   0  35  57
     14  64  29  95  93  15  93  49  26  27
     44  98  84   0  14  65  30  35  15  50  31
     51  81  58  40  64  19  90  63  88  20  86  10
     22  55  76  41  28  10  20  85  52  83  85  33  95
     35  48  35  93  29  10   9  69  53  27   2  72  94  40
     75  11  15  75   2  98  76  81  45  40  91  38  95  64  38
          5  25  14  24  79  51   3  86  48   2  33  76  29  99
             51  15  82  80  64  19  96  95  34  93  24  79  34
                  2  28  29  64  87  11  52  50  58  41  64  13
                     96  73  37  21  44  46  86  52  76  30  42
                          5  39  14  46  16  66  46  92  86  76
```

(to be continued)

2. (35 points) You may start with program skeleton **fexam_skeleton_2.c** and change the file name to **fexam_DXXXXXXX_2.c**. The following table is the digit-value mapping for base-62 numerals. For a string of digits and English letters str, it is converted an integer num with the ***smallest possible base***. For example, “1234321” is converted to a decimal integer 24336 as a base-5 numeral; “abcd” is converted to a decimal integer 2364759 as a base-40 numeral; “45yesAD” is converted to a decimal integer 211144510206 as a base-61 numeral.

digit	value	digit	value	digit	value	digit	value	digit	value
0	0	E	14	S	28	g	42	u	56
1	1	F	15	T	29	h	43	v	57
2	2	G	16	U	30	i	44	w	58
3	3	H	17	V	31	j	45	x	59
4	4	I	18	W	32	k	46	y	60
5	5	J	19	X	33	l	47	z	61
6	6	K	20	Y	34	m	48		
7	7	L	21	Z	35	n	49		
8	8	M	22	a	36	o	50		
9	9	N	23	b	37	p	51		
A	10	O	24	c	38	q	52		
B	11	P	25	d	39	r	53		
C	12	Q	26	e	40	s	54		
D	13	R	27	f	41	t	55		

Write a C program to perform the following steps:

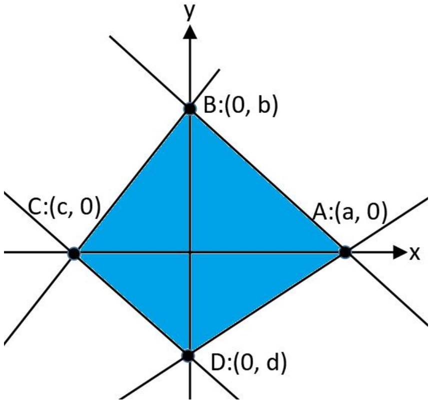
a. Enter a string of digits and English letters str.
b. Find the smallest possible base base and convert str to its equivalent decimal numeral num. Report an error message, if str contains a non-alphanumerical character.
c. Output the values of base and num.
d. Output num as a 64-bit binary numeral with leading zeros and print a space after every eight bits.
e. Output num as a 16-digit hexadecimal numeral with leading zeros and print a space after every four digits.

Repeat the above steps until the input numeral str is a string of 0’s. Program execution example:

```
Enter a numeral (a string of digits and English letters): 1234.567
The input string is an invalid numeral.

Enter a numeral (a string of digits and English letters): 1234567
Base: 8, Decimal value: 342391
Binary numeral: 00000000 00000000 00000000 00000000 00000000 00000101 00111001 01110111
Hexadecimal numeral: 0000 0000 0005 3977

Enter a numeral (a string of digits and English letters): ABC582abc
Base: 39, Decimal value: 55072328406809
Binary numeral: 00000000 00000000 00110010 00010110 10000110 10010010 01010111 00011001
Hexadecimal numeral: 0000 3216 8692 5719

Enter a numeral (a string of digits and English letters): MyNumeral999
Base: 61, Decimal value: 4262436098452512148
Binary numeral: 00111011 00100111 00110111 00001100 01010110 00010000 01011101 10010100
Hexadecimal numeral: 3B27 370C 5610 5D94

Enter a numeral (a string of digits and English letters): 0000
```

(to be continued)

3. (35 points) You may start with the project skeleton in directory **fexam_skeleton_3** and change the directory name to **fexam_DXXXXXXX_3**. Giving four points A(a, 0), B(0, b), C(c,0), and D(0, d), where c<0<a and d<0<b on the XY-plane as the following figure, the area of quadrilateral (四邊形) (blue shaded region) is 1/2|(a-c)(b-d)|. The equations of lines $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, and $\overline{DA}$ are bx+ay=ab, bx+cy=bc, dx+cy=dc, and dx+ay=ad, respectively.



Write a C program to perform the following steps:

a. Define and implement four functions **double** f1(**double** x), **double** f2(**double** x), **double** f3(**double** x), and **double** f4(**double** x) for lines $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, and $\overline{DA}$.
b. Define and implement function **double** right_Riemann_sum(**double** r, **double** s, **double** (*f)(**double**), **double** (*g)(**double**)) to compute the area covered by two functional parameters f and g between interval (r, s) along the X axis.
c. Enter four real numbers (**double** type) a, b, c and d, where c<0<a and d<0<b, to represent points A(a, 0), B(0, b), C(c,0), and D(0, d).
d. Compute the area of quadrilateral ABCD using ***right Riemann sum*** approach.
e. Print the area of triangle BCD, the area of triangle BAD, and the area of quadrilateral ABCD.
f. Verify the result with the area formula 1/2|(a-c)(b-d)| with the error less than 10^{-6}.

(Hint: The equations of line segments $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, and $\overline{DA}$ can be rewritten as functions y=f1(x)=-b/a x+b and y=f2(x)=-c/b x+c, y=f3(x)=-d/c x+d, and y=f4(x)=-d/a x+d, respectively.)

Program execution example: (in the next page)

```
Enter real number a for point A(a, 0), a>0: 5.3
Enter real number b for point B(0, b), b>0: 4.2
Enter real number c for point C(c, 0), c<0: -6.7
Enter real number c for point D(0, d), d<0: -3.8

Points: A=(5.3000, 0), B=(0, 4.2000), C=(-6.7000, 0), D=(0, -3.8000)

**** Compute the area of triangle BCD.
Number of intervals: 1, interval size: 6.700000, area: 53.600000
Number of intervals: 2, interval size: 3.350000, area: 40.200000
Number of intervals: 4, interval size: 1.675000, area: 33.500000
Number of intervals: 8, interval size: 0.837500, area: 30.150000
Number of intervals: 16, interval size: 0.418750, area: 28.475000
Number of intervals: 32, interval size: 0.209375, area: 27.637500
Number of intervals: 64, interval size: 0.104688, area: 27.218750
Number of intervals: 128, interval size: 0.052344, area: 27.009375
Number of intervals: 256, interval size: 0.026172, area: 26.904688
Number of intervals: 512, interval size: 0.013086, area: 26.852344
Number of intervals: 1024, interval size: 0.006543, area: 26.826172
Number of intervals: 2048, interval size: 0.003271, area: 26.813086
Number of intervals: 4096, interval size: 0.001636, area: 26.806543
Number of intervals: 8192, interval size: 0.000818, area: 26.803271
Number of intervals: 16384, interval size: 0.000409, area: 26.801636
Number of intervals: 32768, interval size: 0.000204, area: 26.800818
Number of intervals: 65536, interval size: 0.000102, area: 26.800409
Number of intervals: 131072, interval size: 0.000051, area: 26.800204
Number of intervals: 262144, interval size: 0.000026, area: 26.800102
Number of intervals: 524288, interval size: 0.000013, area: 26.800051
Number of intervals: 1048576, interval size: 0.000006, area: 26.800026
Number of intervals: 2097152, interval size: 0.000003, area: 26.800013
Number of intervals: 4194304, interval size: 0.000002, area: 26.800006
Number of intervals: 8388608, interval size: 0.000001, area: 26.800003
Number of intervals: 16777216, interval size: 0.000000, area: 26.800002
Number of intervals: 33554432, interval size: 0.000000, area: 26.800001
The number of intervals: 33554432

**** Compute the area of triangle BAD.
Number of intervals: 1, interval size: 5.300000, area: 0.000000
Number of intervals: 2, interval size: 2.650000, area: 10.600000
Number of intervals: 4, interval size: 1.325000, area: 15.900000
Number of intervals: 8, interval size: 0.662500, area: 18.550000
Number of intervals: 16, interval size: 0.331250, area: 19.875000
Number of intervals: 32, interval size: 0.165625, area: 20.537500
Number of intervals: 64, interval size: 0.082812, area: 20.868750
Number of intervals: 128, interval size: 0.041406, area: 21.034375
Number of intervals: 256, interval size: 0.020703, area: 21.117188
Number of intervals: 512, interval size: 0.010352, area: 21.158594
Number of intervals: 1024, interval size: 0.005176, area: 21.179297
Number of intervals: 2048, interval size: 0.002588, area: 21.189648
Number of intervals: 4096, interval size: 0.001294, area: 21.194824
Number of intervals: 8192, interval size: 0.000647, area: 21.197412
Number of intervals: 16384, interval size: 0.000323, area: 21.198706
Number of intervals: 32768, interval size: 0.000162, area: 21.199353
Number of intervals: 65536, interval size: 0.000081, area: 21.199677
Number of intervals: 131072, interval size: 0.000040, area: 21.199838
Number of intervals: 262144, interval size: 0.000020, area: 21.199919
Number of intervals: 524288, interval size: 0.000010, area: 21.199960
Number of intervals: 1048576, interval size: 0.000005, area: 21.199980
Number of intervals: 2097152, interval size: 0.000003, area: 21.199990
Number of intervals: 4194304, interval size: 0.000001, area: 21.199995
Number of intervals: 8388608, interval size: 0.000001, area: 21.199997
Number of intervals: 16777216, interval size: 0.000000, area: 21.199999
Number of intervals: 33554432, interval size: 0.000000, area: 21.199999
The number of intervals: 33554432

>>>> Area of triangle BCD: 26.800001
>>>> Area of triangle BAD: 21.199999
>>>> Area of the quadrilateral ABCD: 48.000000
>>>> The result of 1/2|(a-c)(b-d)| is: 48.000000
>>>> The error is: 0.000000
```