LAB 12/24

1. **Homework Problem I**

part1.

Design a program to simulate 100 rolls of a six-sided die:
Count the number of times each side of die appears and for each count use "*" to print out.

part2.

We have the following score data, the last one represents the end of mark, not the data.

72,23,38,86,62,29,99,96,67,78,87,76,65,58,86,69,93,34,45,55,52,7,74,48,83,39 ,90,18,82,26,65,53,39,96,66,-1,

(a). Calculate the average score of the data.
(b). Count the number of data in each interval. The difference between each interval is 10 point. (統計各區間人數, 區間差距為 10 分) That is
    0~9, 10~19, 20~29, 30~39, 40~49,50~59, 60~69,70~79, 80~89, 90~100
(c). Count the number of data above the average and below the average.

```
----------part 1----------
1.:**************
2.:*********
3.:********************
4.:*********************
5.:**********************
6.:********************
----------part 2----------
72,23,38,86,62,29,99,96,67,78,87,76,65,58,86,69,93,34,45,55,52,7,74,48,83,39,90,18,82,26,65,53,39,96,66,-1,
*****SCORE REPORT*****

MEAN = 61.600
ABOVE MEAN = 15
BELOW MEAN = 20

 0~  9:*
10~ 19:*
20~ 29:***
30~ 39:****
40~ 49:**
50~ 59:****
60~ 69:******
70~ 79:****
80~ 89:*****
90~100:*****
```

2. Homework Problem II

part 1.

An interesting method of encoding data is to load a message to be encoded into a two-dimensional array and then interchange rows and interchange columns a number of times. The resulting sequence of characters is the encoded message.
In order to decode the message, the sequence of steps used in the encoding process is followed in reverse order. For example, consider the message
I HAVE BUT ONE LIFE TO GIVE FOR MY COUNTRY.
Let us load the message into a 6x7 array:

I		H	A	V	E	
B	U	T		O	N	E
	L	I	F	E		T
O		G	I	V	E	
F	O	R		M	Y	
C	O	U	N	T	R	Y

Now consider the following encoding process:

(1). Interchange rows 1 and 3   (2) Interchange columns 2 and 5

	L	I	F	E		T
B	U	T		O	N	E
I		H	A	V	E	
O		G	I	V	E	
F	O	R		M	Y	
C	O	U	N	T	R	Y

	E	I	F	L		T
B	O	T		U	N	E
I	V	H	A		E	
O	V	G	I		E	
F	M	R		O	Y	
C	T	U	N	O	R	Y

(3). Interchange rows 4 and 6   **(4) Interchange columns 1 and 5**

	E	I	F	L		T
B	O	T			U	N	E
I	V	H	A		E	
C	T	U	N	O	R	Y
F	M	R			O	Y	
O	V	G	I			E	

	L	E	I	F			T
U	O	T			B	N	E
	V	H	A	I	E	
O	T	U	N	C	R	Y
O	M	R			F	Y	
	V	G	I	O	E	

The resulting string is LEIF TUOT BNE VHAIE OTUNCRYOMR FY VGIOE. In order to decode the message, the encoding process is reversed; i.e., the encoded message would be loaded into a 6×7 array

And then the following interchanges would be performed:

> **Columns 1 and 5**
> **Rows 4 and 6**
> **Columns 2 and 5**
> **Rows 1 and 3**

Write a main program to encode and decode messages, using subroutines to perform the column interchange and row interchange operations.

part2.

Design a program to print out the result of matrix multiplication.(Input is variable)

$$\begin{bmatrix} 2 & 1 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 3 & 1 & 1 & 3 \end{bmatrix} \times \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 1 \\ 1 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 27 & 15 \\ 7 & 11 & 7 \\ 9 & 20 & 12 \end{bmatrix}$$

```
----------part 1----------
I HAVE BUT ONE LIFE TO GIVE FOR MY COUNTRY
Encoded massage:LEIF  TUOT BNE VHAIE OTUNCRYOMR FY  UGIOE
Decoded massage:I HAVE BUT ONE LIFE TO GIVE FOR MY COUNTRY
----------part 2----------
Matrix1 size:
3 4
Matrix1 content:
2 1 3 4
1 2 1 1
3 1 1 3
Matrix2 size:
4 3
Matrix2 content:
1 3 2
2 1 1
1 4 2
1 2 1
Matrix1 x Matrix2 =
 11 27 15
  7 11  7
  9 20 12
```

3. LU decomposition of a matrix.

LU decomposition is a method to decompose a n*n matrix A into the multiplication of **an upper triangular matrix U** and **a lower triangular matrix L**. Both U and L have the same dimension as A. Taking a 3*3 matrix as example:

$$
A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \overset{\text{L}}{\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}} \overset{\text{U}}{\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}}
$$

In this question, we simply set the values on the diagonal of L as 1. That is, **$l_{11} = l_{22} = l_{33} = 1$**.

Following the guidance below, please write a function **void LUdecom(...)** to calculate L, U and show the result.

For each i = 0,1,2...n-1

for k = i, i+1...n-1 produces the kth row of U:

$$
U_{i,k} = A_{i,k} - \sum_{j=0}^{i} (L_{i,j} * U_{j,k})
$$

for k = i+1, i+2 ... n-1 and $L_{i,i} = 1$ produces the kth column of L:

$$
L_{k,i} = \left(A_{k,i} - \sum_{j=0}^{i} (L_{k,j} * U_{j,i}) \right) / U_{i,i}
$$

```c
#define N 5
int M[N][N] = {1,2,3,4,5,1,3,4,9,6,10,7,3,3,1,1,4,5,7,8,12,11,8,9,9};
void LUdecom(...);
void show(float L[N][N],float U[N][N]){
    printf("L:\n");
    for(int i=0;i<N;i++){
        for(int j=0;j<=i;j++)
            printf("%3.2f\t", L[i][j]);
        printf("\n");
    }
    printf("--------------\n");
    printf("U:\n");
    for(int i=0;i<N;i++){
        for(int j=0;j<i;j++)
            printf("            ");
        for(int j=i;j<N;j++)
            printf("%3.2f\t", U[i][j]);
        printf("\n");
    }
}
int main(){
    float L[N][N]={0}, U[N][N]={0};
    LUdecom(...)
    show(L, U);
}
```

Please find out L and U, the result should be like:

4. **Wave animation**

   In homework 11, we learn how to draw a function on the screen using array. In this problem we would like to continuously implement the same method to generate the animation of sine or cosine wave.

   Let user input frequency f, and choose a wave. Please show the wave animation on the screen with width=100, height=25.

   The sine and cosine function is shown below:

   $$\sin(x) = \ sin\left(2 * \pi * f * \frac{x}{100}\right)$$

   $$\cos(x) = cos\left(2 * \pi * f * \frac{x}{100}\right)$$

   Please design a function void shift(char **), which will shift the wave to left (along x axis) by 1.

   You can use following code for help

```c
#define M 100
#define N 25

void shift(char **a);
void show(char **a){
    for(int y=N-1;y>=0;y--){
        printf("|");
        for(int x=0;x<M;x++)
            printf("%c", a[x][y]);
        printf("|");
        printf("\n");
    }
}
int main(){
    char **w1, **w2;
    double f, ratio;
    int menu, in, ymax=1, ymin=-1;

    printf("please input frequency f:");
    scanf("%lf", &f);

    //initialize array w1 and w2
    //something missed here

    for(int i=0;i<M;i++)
        //allocate space and * to your array
        //something missed here
    int time=0;
    printf("1:sin\n2:cos\n:");
    scanf("%d", &menu);
    while(1){
        switch(menu){
            case 1:
                shift(w1);
                //something missed here
                show(w1);
                break;
```

```
            case 2:
                shift(w2);
                //something missed here
                show(w2);
                break;
        }
        _sleep(200);
        system("cls");
    }
}
```

The result should be like: