Week 4: C Programming Practical

1. Write a C program that accepts a positive integer N, and calculate and print (1) the sum of the first N natural numbers and (2) the product of the first N natural numbers.

2. Display Z down to A and z down to a on screen.

3. Using switch case statement to display month name (January, February, March, April, …, December according to given month number 1-12).

4. Read an integer number, check and display whether it is an EVEN or ODD number using the switch statement.

5. Input two vectors and check whether they are the same. Then input two matrices and check whether they are the same.

6. Input two vectors and output the dot product of the vectors. Print an error message if the vectors do not have the same length. Given two vectors A = [a_1, a_2, …, a_n] and B = [b_1, b_2, …, b_n], their dot product is $\sum_{i=1}^{n} a_i b_i$.

7. Input two matrices and output the multiplication of the matrices. Print an error message if the number of columns in the first matrix is not equal to the number of rows in the second matrix. Given two matrices $A = \left(a_{i,j}\right)_{l\times m}$ and $B = \left(b_{j,k}\right)_{m\times n}$, $C = A \times B = \left(c_{i,k}\right)_{l\times n}$ with $c_{i,k} = \sum_{j=1}^{m} a_{i,j} b_{j,k}$.

8. Write a C program to print all numbers between 1 to 200 which divided by 5 and the remainder will be 2, also show how many these numbers are there.

Appendix: vector and matrix

1. A vector can be thought of as a **list numbers**, for instance, a row vector can be written as :

$$[1 \quad 2 \quad 3 \quad 4 \quad 5]$$

   and a column vector can be written as:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

   Suppose we have two 1*6 vectors: one is $[a_1\ a_2\ a_3\ a_4\ a_5\ a_6]$, and the other one is $[b_1\ b_2\ b_3\ b_4\ b_5\ b_6]$. The dot product of the two vectors is: $[a_1*b_1\ \ a_2*a_2\ \ a_3*b_3\ \ a_4*b_4\ \ a_5*b_5\ \ a_6*b_6]$ which is also an 1*6 vector. Therefore, we should make sure the length of the vectors are the same when we calculate dot product.

   In C language, we can implement a vector with an 1d array.

2. A matrix is an m * n array of numbers, where m is the number of rows and n is the number of columns. For example, this is a 3*4 matrix:

$$\begin{bmatrix} 2 & 3 & 6 & 143 \\ 15 & 1 & 8 & 11 \\ 12 & 14 & 16 & 13 \end{bmatrix}$$

   How to multiply two given matrices? To multiply one matrix with another, we need to check first, if the number of columns of the first matrix is equal to the number of rows of the second matrix. For instance, we want to multiply m_1*n_1 matrix A with m_2*n_2 matrix B:

$$C_{(m1*n2)} = A_{(m1*n1)} * B_{(m2*n2)}$$

   Then n_1 must be equal to m_2. The result of the multiplication is a m_1*n_2 matrix C, which has the number of rows of the first matrix and the number of columns of the second matrix.

To give you an idea, a 3*4 matrix A can be multiplied with a 4*5 matrix B, and the resulting matrix C is a 3*5 matrix. The 3*4 matrix A **cannot** be multiplied with a 3*5 matrix B, because the number of columns of A is 4, and the number of rows of the other matrix must also be 4.

$\mathbf{C_{(3*5)} = A_{(3*4)} * B_{(4*5)}}$ **This is correct!**

$\mathbf{C = A_{(3*4)} * B_{(3*5)}}$ **This is wrong!**

How to do Matrix multiplication? Suppose A is an m*n matrix and B is an n*p matrix,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix}$$

then C=A*B is an m*p matrix:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{bmatrix}$$

where $c_{ij} = a_{i1}*b_{1j} + a_{i2}*b_{2j} + \ldots + a_{in}*b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$
which is the dot product of the i^{th} row of A and the j^{th} column of B.

For instance, A is 2*3 matrix, and B is a 3*2 matrix:

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 6 & 2 \\ 5 & 7 \\ 4 & 1 \end{bmatrix}$$

Then C = A*B is a 2*2 matrix:

C=

$$\begin{bmatrix} \textit{the dot product of 1st row of A and 1st column of B} & \textit{the dot product of 1st row of A and 2nd column of B} \\ \textit{the dot product of 2nd row of A and 1st column of B} & \textit{the dot product of 2nd row of A and 2nd column of B} \end{bmatrix}$$

$$= \begin{bmatrix} 1*6+4*5+7*4 & 1*2+4*7+7*1 \\ 2*6+5*5+3*4 & 2*2+5*7+3*1 \end{bmatrix}$$

$$= \begin{bmatrix} 54 & 37 \\ 49 & 42 \end{bmatrix}$$

In C language, we can implement a matrix with a 2d array.
For example, we declare a 2*3 array A and a 3*2 array B:

```
float A[2][3] = {{1,4,7},{2,5,3}};
float B[3][2] = {{6,2},{5,7},{4,1}};
```

Similar to 1d array, the indexes of 2d array also start with 0.
A[0][0] is 1, and A[1][2] is 3. B[0][0] is 6, and B[2][1] is 1.

We can initialize an M*N array with Nested For Loop:

```
printf("Please input the elements of %d * %d matrix:\n", M,N);
for(i=0;i<M;i++)
   for(j=0;j<N;j++)
      scanf("%f",&C[i][j]);

printf("The elements of the matrix are:\n");
for(i=0;i<M;i++)
   for(j=0;j<N;j++)
      {printf("%-8.1f  ",C[i][j]);
       if(j==N-1)
       printf("\n");
      }
```

Submission Requirement

(1) This coursework has 8 problems. Please prepare one source file (.c file) per problem.

(2) Make sure that each source file can be compiled without error, and the corresponding executable file can produce the expected result.

(3) You can either submit 8 individual source files, or compress the source files in a zip file (or a rar file) and submit the compressed file.

(4) The submission link is **Report T1 Week 8**: https://canvas.sussex.ac.uk/courses/25884/assignments/100783

(5) The due time is 20 Nov at 17:00 (UK time).