**Homework 1: Array problem I**

**1.You can use dynamic array or static array in the following problems. But dynamic array prefer**

**2. Using function to operate or calculate.**

**3. Problem 1~ Problem 6 are basic problems.**

**Problem 1:**

A **binary (digital) picture** is a two-dimensional array, each of whose

elements is 0 or 1. The image is interpreted as light (1) on a dark (0)

background. For example, here is a binary picture of a football:

1. 0 0 0 0 0 0

0 0 1 1 1 0 0

0 1 1 1 1 1 0

0 0 1 1 1 0 0

0 0 0 0 0 0 0

In analyzing the picture, it is often necessary to identify the edges.

A **pixel ( p**icture **el**ement ) is an edge pixel if it is 1 and at least one of

pixels immediately above, below, left or right is 0. We can show the

edges to 1 and all other pixels to 0. After identifying the edges in the

preceding picture, we obtain

0 0 0 0 0 0 0

0 0 1 1 1 0 0

0 1 0 0 0 1 0

0 0 1 1 1 0 0

0 0 0 0 0 0 0

**Problem 2:**

**Design a matrix multiplication function. Using dynamic array only.**

**Write a main program to test this function.**

**Example:**

** ×  = **

**Problem 3:**

Write a program:

1. reads the row “r” and the column “c” of one matrix.
2. reads r x c number of integers,
3. stores these integers in one r x c array,
4. displays the sum of all its elements,
5. display the sum of each row’s elements,
6. display the sum of each column’s elements.

**Problem 4:**

In linear algebra, a matrix is called a **Toeplitz matrix** when the elements of each diagonal to the main diagonal are equal between each other. For example the following 5x5 matrix demonstrates the generic form of a 5x5

**Toeplitz matrix** :

Write a program that reads the one positive integer “r” and reads “r” integers and stores them in the first row and first column of a “ r x r ” array. Next the program should create the Toeplitz matrix and display its elements.

**Problem 5 :**

Write a program that creates an array of 150 random integers in the range from 1 to 200 and then, using the sequential search, searches the array 200 times using randomly generated targets in the same range(產生200個 random data 以作為200次的search之用) .

At the end of the program, display the following statistics:

1. The number of successful searches.
2. The percentage of successful searches.

**Problem 6**: Repeat Problem (5) using the **binary search.**

**Problem7:**

**Please define the following functions: (at least 7 functions)**

**1. Input data for one vector ( 1-d array 1×m or n×1 )**

**2. Output data for one vector ( 1-d array 1×m or n×1 )**

**3. Input data for a matrix ( 2-d array n×m )**

**4. Output data for a matrix ( 2-d array n×m )**

**5. Multiplying a matrix by a vector:**

**(a). Multiplying on the right**

**   ×  ＝ **

**(b) Multiplying on the left.**

**   ×    =   **

**6. The transpose of a matrix.**

**A =  =   **

**7. Matrix Addition**

**8. Matrix Multiplication**

**9. You can define other operations for the vector and matrix**

**10. Please write a main program (using the switch structure) and let the user to choose the operation that he wants to execute.**

**11. You can put all the functions in 1 ~ 9 to create your heading file and include this heading file in the main program to execute.**

**Problem 8 ( *The Sieve of Eratosthenes* )**

**A prime integer is any integer greater than 1 that can be divided evenly**

**only by itself and 1.** The most efficient way to find all of the small primes (say all

those less than 10,000,000) is by using a sieve such as **the Sieve of Eratosthenes**

Make a list of all the integers less than or equal to *n* (and greater than one). Strike out the multiples of all primes less than or equal to the square root of *n*, then the numbers that are left are the primes.

For example, to find all the primes less than or equal to 30, first list the numbers from 2 to 30. 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The first number 2 is prime, so keep it (we will color it green) and cross out its multiples (we will color them red), so the red numbers are not prime.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The first number left (still black) is 3, so it is the first odd prime. Keep it and cross out all of its multiples. We know that all multiples less than 9 (i.e. 6) will already have been crossed out, so we can start crossing out at 32=9.

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Now the first number left (still black) is 5, the second odd prime. So keep it also and cross out all of its multiples (all multiples less than 52=25 have already been crossed out, and in fact 25 is the only multiple not yet crossed out).

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

The next number left, 7, is larger than the square root of 30, so there are no multiples of

7 to cross off that haven't already been crossed off (14 and 28 by 2, and 21 by 3), and

therefore the sieve is complete.

Therefore all of the numbers left are primes: {2, 3, 5, 7, 11, 13, 17, 19, 23, 29}.

Notice we just found these primes without dividing.

**Problem 9:**

**Plotting a graph: Please see the Note of this problem.**

**The plotting function is**

**f(x)= (Π ) 0≦ x ≦ 10**

**NOTE: Plotting a graph:**

**1. Generate a table of number pairs :**

***t* *p(t)***

**0 30.0**

**1 27.1**

**2 22.6**

**… … (for *t*= 0,…, n-1)**

**Or**

***i* *x y***

**0 0.0 0.0**

**1 2.0 32.0**

**2 3.0 72.0**

**… … … (*x, y*for i=0,…, n-1)**

**These numbers may represent experimental data, say position *p(t)* vs. time *t* for a falling object or may be generated from a particular functional relation between *x* and *y* as *y(x*)=8*x***

**2. Determine the range of both *x*and *y*: A determination of both the minimum and maximum values of *x* and *y* in the data set :**

**(Range)=(*x - x*)**

**(Range)= (*y - y*)**

**These values are then used for scaling the *x* and *y* axes.**

**That is, adjusting the scales of the axes so that the graph fits neatly on the graph paper.**

**3. Step through the points and graph them one by one.**

X **Y**

**| \***  **|**

**| \* |**

**| \* |**

**| \* |**

**| \* |**

**| \* |**

**| \* |**

**Note:**

**1.When one line is displayed, 80 or so characters will be printed for a particular value of *x*.**

**2. All of these characters will be blanks except one.**

**3. The position corresponding to *y* will contain some symbol --- e.g.,an asterisk.**

**4. To determine the proper placement of asterisk, consider the following:**

***y* = 16.38 *y*= –7.21**

**so**

**(Range)= 16.38 – (–7.21) =23.59**

**Problem:**

**If at x= 3.0, y(3.0)=12.2,where in the horizontal line is the asterisk to be printed?**

**If the y axis is to be 81 columns wide, we could first define**

**Ratio = **

**Notice that ratio is between 0.0 and 1.0.**

**The appropriate column(position) for the asterisk is**

**IY = ceil ( 80 \* Ratio)**

**For the particular choice of numbers above , y(3) =12.2**

**We obtain:**

**Ratio =  = 0.82281**

**IY =ceil( 80 \* 0.82281) = ceil(65.8248) =66**

**Where double ceil(double x)**

**Return 大於x值的最小整數**

**Note:**

**1. When y(i)= *y* Ratio =0 then IY=80\*0=0**

**2. When y(i)= *y* Ratio =1 then IY=80\*1=80**

**3. 最小值落在第0行, 最大值落在第80行, 因此總共有81行.**