1. Ackermann’s function $A(m, n)$ is defined as follows.

$$A(m, n) = \begin{cases} 
  n + 1 & \text{if } m = 0 \\
  A(m - 1, 1) & \text{if } n = 0 \\
  A(m - 1, A(m, n - 1)) & \text{otherwise}
\end{cases}$$

Write a recursive function for computing this function. (10%)

2. If we have the declaration $A[1..15][1..10][1..20]$ and $\alpha$ is the address of $A[1][1][1]$, using row major order, please answer the following questions.

1) Obtain an addressing formula for the element $A[i][j][k]$. (5%)

2) What is the address for $A[3][7][2]$. (5%)

3. About the infix translating to postfix expressions. Answer the following questions.

1) Write the postfix form of the expressions $A / (B – C) + D * (E – A) * C / F$ (5%)

2) Use the stack to translate infix to postfix expressions, please write out the processing sequence including input token, stack, and output. (10%)

4. A stack may be regarded as a railway switching network like the one in the following figure. Railroad cars numbered 1, 2, 3, $\ldots$, n are initially in the top right track segment (in this order, left to right). Railroad cars can be moved into vertical track segment one at a time from either of the horizontal segments and then moved from the vertical segment to any one of the horizontal segments. The vertical segment operates as a stack as new cars enter at the top and cars depart the vertical segment from the top. For instance, if n=3, we could move car 1 into the verical segment, move 2 in, move 3 in, and then take the cars out producing the new order 3, 2, 1. For $n=4$, Please answer as follows.

1) Write the all permutations. (5%)

2) What are the impossible permutations of cars that can be not obtained? (5%)

3) Write the recursive formula for n ($B_n$). (5%)

5. 請回答下列關於 Queues 的問題。

1) 利用 Linear Array 實作 Queues，可能面臨哪二種最糟的情況？請說明之。 (5%)

2) 利用 Circular Array 可以避免 Linear Array 實作 Queues 所產生的缺點，但亦會面臨一些問題，請說明之並提出解決方法。 (10%)

背面尚有試題

試題務請以黑色筆書寫，俾便製版印刷。
6. 请图示说明下列 Linked Lists.
   (1) Singly Linked Lists. (5%)
   (2) Circular Linked Lists. (5%)
   (3) Doubly Linked Lists (circular and with head node). (5%)

7. If we represent generalized lists using the following node structure:

<table>
<thead>
<tr>
<th>tag = True/False</th>
<th>data/dlink</th>
<th>link</th>
</tr>
</thead>
</table>

   Let $A$ be a list, then the data/dlink data member holds an atom if head($A$) is an atom (tag == False) and holds a pointer to the list representation of head($A$) if head($A$) is a list (tag == True). Using this node structure, please draw the representation of the following lists.

   (1) $A = (a, (b, c))$ (5%)
   (2) $B = (a, (b, c), (d, e), f)$ (5%)

8. In String Pattern Matching: The Knuth-Morris-Pratt Algorithm. We should caculus the failure function of the pattern first. The definition of failure function is

   \[
   f(j) = \begin{cases} 
   \text{largest } k < j \text{ such that } P_0P_1\ldots P_k = P_{j-k}P_{j-k-1}\ldots P_j & \text{if such a } k \geq 0 \text{ exists.} \\
   -1 & \text{otherwise}
   \end{cases}
   \]

   Please compute the failure function for each of the following the patterns.

   (1) baaaab (5%)
   (2) abccabab (5%)

9. Answer the questions according to below function prod.

   (1) How many times is line 8 executed? (5%)
   (2) What is the time complexity of the following program? (5%)

   ```
   void prod(matrix a, matrix b, matrix c, int m, int n, int p)
   {
   for (int i=0; i<m; i++)
   for(int j=0; j<p; j++)
   { c[i][j]=0
   for (int k=0; k<n; k++)
   c[i][j] += a[i][k] * b[k][j];
   }
   }
   ```

10. 有二個功能相同的程式，已知程式 A 採用的演算法時間複雜度為 $O(n\log n)$，程式 B 採用的演算法時間複雜度為 $O(n^2)$，問是否表示程式 A 一定執行的比程式 B 快，為什麼？ (10%)