# Department of Computer Science and Engineering
# National Sun Yat-sen University
# Data Structures Quiz, Chapter 1, Sept. 23, 2024

1. What are printed by each of the following C programs? (30%)

   (a)
   ```
   char d=172;
   printf("%d \n",d);
   ```

   (b)
   ```
   char a= 'C';      // 'A'=65
   char b= 'G';
   printf("%d    %c    %d \n", a, a, b-a);
   ```

2. Ackermann's function $A(m,n)$ is defined as follows:

   $A(m,n)=n+1$, if $m=0$

   $A(m,n)=A(m-1,1)$, if $n=0$

   $A(m,n)=A(m-1,A(m,n-1))$, otherwise.

   Calculate the value of $A(2,2)$. (30%)

3. Write a recursive C/C++ function to print out all permutations of given elements. (40%)

```
void Permu(char a[ ], int k, int m)
//Generate all the permutations of a[k], ..., a[m]
{
```

| Please write the body of Permu ( ). |
|---|

```
} // end of Permu ( )
int main( )
{    char a[ ]={ 'a','b','c','d'};
     Permu(a,0,3);
};
```

Answers:

參考解答

1. (a) -84

解釋: 172 儲存於 char 時，其二進位之值為 10101100。列印時，以%d 印出，將其視為 signed char (integer) ，其 8 bits 中，最左方的 bit 作為判斷正負使用，若為 1 則為負數，若為 0 則為正數。本題最左方為 1，故為負數。10101100 之 2 補數 (2's complement)為 $01010100_{(2)} = 84_{(10)}$，故答案為 $84_{(10)}$的負數 -84。

註：10101100 之 2 補數：先將全部的 bit，0 轉換成 1，1 轉換成 0(亦即為 1's complement)，得到 01010011，最後再加 1，得到 01010100。另一種簡單算法，計算 172 的 256 補數，亦即 256-172=84。

(b) 67 C 4

解釋: 'C' = 67，以 %d 十進制印出，答案為 67。

'G' = 71，故'G' – 'C' =4，以 %d 十進制印出為 4。

2. $A(2, 2) = A(1, A(2, 1))$
   $A(2, 1) = A(1, A(2, 0))$
   $A(2, 0) = A(1, 1)$
   $A(1, 1) = A(0, A(1, 0))$
   $A(1, 0) = A(0, 1)$
   $A(0, 1) = 2$
   $A(1, 0) = 2$
   $A(1, 1) = A(0, 2) = 3$
   $A(2, 0) = 3$
   $A(2, 1) = A(1, 3)$
   $A(1, 3) = A(0, A(1, 2))$
   $A(1, 2) = A(0, A(1, 1)) = A(0, 3) = 4$
   $A(1, 3) = A(0, 4) = 5$
   $A(2, 1) = 5$
   $A(2,2) = A(1, 5)$
   $A(1, 5) = A(0, A(1, 4))$
   $A(1, 4) = A(0, A(1, 3)) = A(0, 5) = 6$
   $A(1, 5) = A(0, 6) = 7$
   $A(2, 2) = 7$

3.

```
void Permu(char a[ ], int k, int m)
//Generate all the permutations of a[k], ..., a[m]
{
    if (k == m) {    //Output permutation
        for (int i = 0; i <= m; i++) cout << a[i] << " ";
        cout << endl;
    }
    else { //a[k], ..., a[m] has more than one permutation
        for (int i = k; i <= m; i++)
        {
            swap(a[k], a[i]); // exchange
            Permu(a, k+1, m);
            swap(a[k], a[i]);
        }
    } // end of else
} // end of Permu ( )
int main( )
{   char a[ ]={ 'a','b','c','d'};
    Permu(a,0,3);
};
```