Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures - Middle Exam, Oct. 30, 2023

1. What are printed by each of the following C programs? (20%)

(a)
```
int a=22, b=12;
printf("%d    %d\n", a&(~b), b&(~a)); // &: bitwise AND, ~: bitwise NOT
```

(b)
```
int a=13, b=23;
printf("%d    %d\n", a^b^a, a^b^b^a^a^b^a^a); // ^: bitwise XOR
```

(c)
```
int b[ ]={0,1,2,3,4,5,6,7,8,9,10};    int *p, *q;
p=b+1; q=p+2; *(p+2) += 12; *(p++)=b[0]+17; *(b+2) += *(q+1)+15;
printf("%d %d %d %d \n",b[0], b[1], b[2], b[3]);
```

(d)
```
int g(int n) {
    if (n <=1)    return n;
    else    return g(n-2)*2+g(n-1); }
void main( ) { printf("%d %d    \n", g(5), g(6));}
```

(e)
```
void f(int *w, int *x, int y[ ], int z[ ]) {
   *w=55;    *(w+2)=35;
   printf("%d    %d    %d    %d \n", x[0], y[2], *(y+4), z[3]);    }
int main( ) {
   int a[ ]={20,21,22,23,24,25,26,27,28,29,30};
   f(&a[0]+5, &a[2], a+3, a); }
```

2. Determine the frequency count for the statement in line iv of the following program segment: (10%)

```
 i.    for (i=1; i<=n; i++)
 ii.       for (j=1; j<=i; j++)
 iii.          for (k=1; k<=j; k++)
 iv.              y++;
```

3. The data elements are numbered as shown in the left figure, which are stored in a 2-D array, as shown in the right figure. Now, suppose element *n* is stored in (*a*,*b*). Please derive the formula for calculating the value of *n* from *a* and *b*. (10%)

0	1	5	6	14
2	4	7	13	…
3	8	12	…	…
9	11	…	…	…
10	…	…	…	…

0,0	0,1	0,2	…
1,0	1,1	1,2	…
2,0	2,1	2,2…	…
3,0	3,1	3,2	…
…	…	…	…

4. On a stack machine, two operations can be performed: PUSH and POP. When a

sequence of three letters, A, B, and C, is input to the stack machine, please present all possible outputs when all elements in the stack are popped out. (10%)

5. Please present a method for converting an infix expression to a postfix expression by using a stack. You should write down the algorithm and use the conversion of ((A-B)*C-D)/(E+F)-G as an example. (10%)

6. Explain each of the following terms. (16%)
   (a) constructor in a C++ class
   (b) circular queue
   (c) reference count in a generalized list
   (d) asymptotic O notation (represented $f(n)$ as $O(g(n))$

7. Write a recursive C/C++ function to solve the Hanoi tower problem for moving n disks from peg A to peg C with auxiliary peg B, $n \geq 1$. For each disk movement, you have to print the message such as "move disk 3 from peg B to peg A". Note that a larger disk must be put below a smaller disk at any time. (12%)

```
void tower(….)    // the recursive function, printing the movement message
{
Please write the body of tower( ).
} // end of tower( )
void main( ) {
   int n=10;
   tower(n, 'A', 'B', 'C');
} /* end of main ( )*/
```

8. Write a C++ function to concatenate two linear chains (singly linked lists), $x=(x_1, x_2, \ldots, x_{m-1}, x_m)$ and $y=(y_1, y_2, \ldots, y_{n-1}, y_n)$, into a linear chain $z=(x_1, x_2, \ldots, x_{m-1}, x_m, y_1, y_2, \ldots, y_{n-1}, y_n)$. In each chain, a "first" pointer points to the first node, and a "last" pointer points to the last node. Note that x or y may be empty. (12%)

```
class ChainNode {
  public:
    int data;
    ChainNode *link;    // Point to the next node
};
class Chain {
  public:
    ChainNode *first, *last;      // first and last pointers
}
Chain & concatenate(Chain &x, Chain &y )
    // y is concatenated to the end of x. You have to consider empty chains.
{   Chain z;          // The resulting chain
Please write the body of concatenate( ).
    return z;
} // end of concatenate( )
```

Answers:

1. (a) 18 8
   (b) 23 26
   (c) 0 17 21 15
   (d) 11 21
   (e) 22 55 35 23

Explanation:

(a) a=22=$(10110)_2$, ~a=$(01001)_2$, b=12=$(01100)_2$, ~b=$(10011)_2$
a&(~b)=$(10010)_2$=18
b&(~a)=$(01000)_2$=8

(b)
a=13=$(01101)_2$, b=23=$(10111)_2$
a^b^a=b=$(10111)_2$=23
Note: a^a=0. The result is 0 if an even number of 'a' values are performed on XOR.
a^b^b^a^a^b^a^a=a^b=$(11010)_2$=26

(c)
```
p=b+1   //p points to b[1]=1
q=p+2   //q points to b[3]=3
*(p+2) += 12      //p+2, q and b[3] have the same address. b[3]+=12, b[3]=*q=15
*(p++)=b[0]+17 //先做 b[0]+17，p 尚未更改，p 指向 b[1]=b[0]+17=17
                  //p++指向 b[2], *p=2
*(b+2) += *(q+1)+15      //q+1 指向 b[4], b[4]+15=19
                         //b+2 指向 b[2], b[2]=b[2]+19=21
b[ ]={0, 17, 21, 15, 4, 5, 6, 7, 8, 9, 10}
```

(d)
```
g(0)=0
g(1)=1
g(2)=g(0)*2+1=1
g(3)=g(1)*2+g(2)=3
g(4)=g(2)*2+g(3)=5
g(5)=g(3)*2+g(4)=3*2+5=11
g(6)=g(4)*2+g(5)=5*2+11=21
```

(e)
```
f(&a[0] + 5, &a[2], a + 3, a);
  //&a[0] + 5 為 a[5]=25, &a[2] 為 a[2]=22, a+3 為 a[3], a 為 a[0]
  w 與 a[5]同位址，x 與 a[2] 同位址，y 與 a[3] 同位址，z 與 a[0] 同位址
  *w = 55;   *(w + 2) = 35;
  //w=55, a[5]=55, a[7]=35
  x[0]=a[2]=22
  y[2]=a[3+2]=a[5]=55
  *(y + 4)=a[3+4]=a[7]=35
  z[3]=a[0+3]=a[3]=23
```

2.

$1 + (1+2) + (1+2+3) + (1+2+3+4) + \ldots + (1+2+3+\ldots+n)$

$$= \sum_{i=1}^{n} (1 + 2 + \cdots + i)$$

$$= \sum_{i=1}^{n} \frac{i(i+1)}{2}$$

$$= \frac{1}{2}\left(\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2}\right)$$

$$= \frac{n^3 + 3n^2 + 2n}{6} = \frac{n(n+1)(n+2)}{6}$$

3.

(a,b)之前的 diagonal(斜線)元素總個數為

$1+2+3+\ldots+(a+b) = \frac{(a+b)(a+b+1)}{2}$

若 $a+b$ 為奇數 (odd)，由上而下遞增；

若 $a+b$ 為偶數 (even)，由下而上遞增；

因此，得到以下答案：

$n = \frac{(a+b)(a+b+1)}{2} + a$, if $a+b$ is odd;

$n = \frac{(a+b)(a+b+1)}{2} + b$, if $a+b$ is even.

4.

可能的順序：ABC, ACB, BAC, BCA, CBA

PUSH 簡寫為 U，POP 簡寫為 O。合法的順序如下：

UOUOUO ABC
UOUUOO ACB
UUOOUO BAC
UUOUOO BCA
UUUOOO CAB

5.

(1) 遇 operand, 直接 output

(2) 遇 operator

(a) 若此 operator 之 precedence 比 top of stack 高 ==> 此 operator 放入 stack.

(b) 否則, 將所有比此 operator 之 precedence 還高之 operator 全 pop 並 output, 再將 operator 放入 stack.

(3) '(': 直接放入 stack

(4) ')': 所有在'('之上的 operator 全部 pop 並 output，但'('及')'不必 output

(5) 最後將 stack 內所有 operator 取出，並 output

symb	postfix	stack
(		(
(		((
A	A	((
-	A	((-
B	AB	((-
)	AB-	(
*****	AB-	(*
C	AB-C	(*
-	AB-C*	(-
D	AB-C*D	(-
)	AB-C*D-	
/	AB-C*D-	/
(	AB-C*D-	/(
E	AB-C*D-E	/(
+	AB-C*D-E	/(+
F	AB-C*D-EF	/(+
)	AB-C*D-EF+	/
-	AB-C*D-EF+/	-
G	AB-C*D-EF+/G	-
	AB-C*D-EF+/G-	

6.

(a) It is a special method that is automatically called when an object of a class is created. Its main purpose to set up the initial values of the object.

(b) In a circular queue, the next position of the last position is the first position. Only n-1 locations are used when there are n available locations

(c) The reference count of an object stores the number of pointers (references) pointing to the object. When the reference count is 0, the object is not used and it can be removed.

(d) $f(n) = O(g(n)$ if there exist positive integers a and b such that

$f(n) \leqq a \cdot g(n)$ for all $n \geqq b$.

7.

```
void tower(int n, char A, char B, char C)
{
      if(n == 1){
            cout <<"move disk " << 1 << "from peg " << A << " to peg " << C << endl;
       }
       else{
            tower (n-1, A, C, B);
            cout <<"move disk " << n << "from peg " << A << " to peg " << C << endl;
            tower (n-1, B, A, C);
       }
       retrun;
} // end of tower( )
```

8.

```
Chain & concatenate(Chain &x, Chain &y )
{
   Chain z;
   if(y.first == 0){    // y.first=NULL, empty y
        z.first=x.first;
        z.last=x.last;
   }
   else if(x.first == 0){    // x.first=NULL, nonempty y and empty x
        z.first=y.first;
        z.last=y.last;
   }
   else{        // nonempty y and nonempty x
     z.first=x.first;
     x.last->link=y.first;
     z.last=y.last;
   }
  return z;
}
```