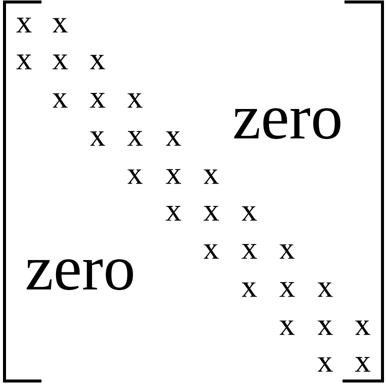
Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 22, 2021

1. What are printed by each of the following C programs? (20%)
   (a) char x, y; x='A'+'B'-100; y='A'+'B'+'C';
   printf("%d %d \n", x, y);
   (b) int a=80; printf("%d \n", (a&(-a)) >>3);
   (c) int a=36, b=13, c=25;
   printf("%d %d \n",b^a^b^b^b, (a^b^a^c^b^a^c^b)+b); //^:XOR
   (d) int c[ ]={10,14,18,22,26}; int *p;
   p=c+1; *(c+3) += 3; *(p++)=c[0]+7; *(c+2) = *(p+1)+5;
   printf("%d %d %d %d \n",c[0], c[1],c[2],c[3]);
   (e) union { char m; int n; }u;
   u.n=168; printf("%d \n",u.m);

2. In the following sparse matrix *a*[ ][ ], all elements other than those on the three diagonals are zero. Suppose the elements in the band formed by these three diagonals are represented by rows in a linear array *b*, with *a*[0][0] (upper left corner) being stored in *b*[0]. Suppose that *b*[*k*] stores the value of *a*[*i*][*j*], 0≤ *i*, *j* ≤*n*-1. Please calculate the addressing formula for *k* with *i* and *j*. (10%)



3. The *Fibonacci Plus sequence* is defined recursively as follows:
   $f(n) = n$, if $n = 0, 1$
   $f(n) = f(n-1) + f(n-2)+1$, if $n \geq 2$.
   (a) What is the value of *f*(5)? (4%)
   (b) How many additions are required for computing *f*(*n*) with an iterative implementation? (4%)
   (c) Suppose the program is written recursively for computing *f*(*n*). How many additions are required? Please derive a general pattern. (6%)

4. Given an infix expression ((A-B)*C-D)/(E+F)-G, please draw its expression tree, and then give the prefix and postfix forms. (10%)

5. Please present the algorithm for converting an infix expression to a postfix expression with a stack (not with a tree). (10%)

6. Explain each of the following terms. (12%)
   (a) constructor in C++ language
   (b) operator overloading in C++ language
   (c) row-major ordering for a 2-D array

7. Write a recursive C/C++ function to print out all permutations of given elements. (12%)

```
void Permutation(char a[ ], int k, int m)
//Generate all the permutations of a[k], ..., a[m]
{
    Please write the body of Permu ( ).
} // end of Permutation ( )
int main( )
{   char a[ ]={ 'a', 'b', 'c', 'd'};
    Permutation(a,0,3);
};
```

8. Let x=(x_1, x_2, …, x_{m-1}, x_m) and y=(y_1, y_2, …, y_{n-1}, y_n) be two linear chains (singly linked lists), where there is a "first" pointer points to the first node, and a "last" pointer points to the last node in each chain. Write a C++ function to concatenate the two chains into a linear chain z=(x_1, x_2, …, x_{m-1}, x_m, y_1, y_2, …, y_{n-1}, y_n). 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) 31 -58 (b) 2 (c) 36 54 (d) 10 17 30 25 (e) -88

Explanation:

(a) x='A'+'B'-100=65+66-100=31; so, the first printed is 31
y='A'+'B'+'C'=65+66+67=198. The maximal positive number for 8-bit is 127.
Thus, 198 should be printed with 2's complement for a negative number, which is -58.

(b) a=80=01010000; -80=101010000
01010000 & 101010000=00010000
00010000 >>3 = 00000010=2

In fact, a&(-a) is to find the rightmost position of 1 in the binary representation of a.

(c) b^a^b^b^b=a=36; //^:XOR
(a^b^a^c^b^a^c^b)+b =(a^b)+b=(36^13)+13=(00100100^00001101)+13
=00101001+13=41+13=54

(d) int c[ ]={10,14,18,22,26}

```
p=c+1; *(c+3) += 3;    *(p++)=c[0]+7;    *(c+2) = *(p+1)+5;
printf("%d %d %d %d \n",c[0], c[1],c[2],c[3]);
```

c[0]=10, which is not changed.

p=c+1, address of p is the same as c[1]

*(c+3) += 3, c[3] is increased by 3, then c[3]=25

*(p++)=c[0]+7, *p is first assigned by c[0]+7=17, that is c[1]=17. Then, p is changed to the address of c[2] by p++.

*(c+2) = *(p+1)+5, p+1 means c[3], that is, *(p+1)+5 means 25+5=30. So *(c+2), which is c[2], is changed to 30.

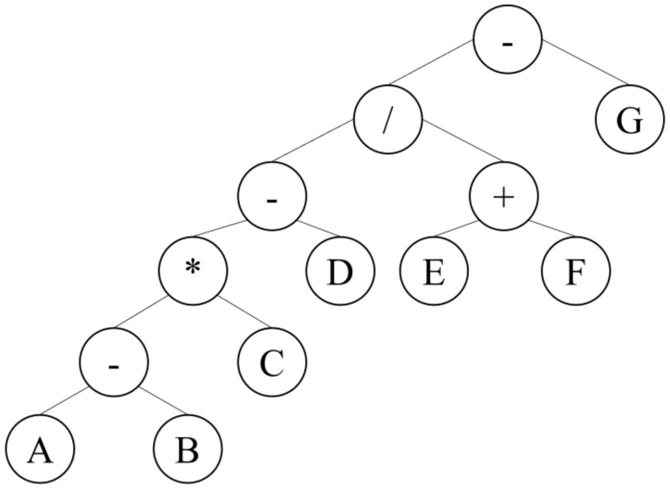
(e) u.m and u.n have the same memory location and the same content.
The maximal positive number for 8-bit is 127.
Thus, u.m=u.n=168 should be printed with 2's complement for a negative number, which is -88.

2. Answer (1) k=3i+(j-i)=2i+j
Answer (2):
k=3i+1 if i < j
k=3i if i = j
k=3i-1 if i > j

3. (a) *f*(5)=12
(b) g(0)=0, g(1)=0, g(2)=2, g(3)=4, g(n)=2(n-1), n>=2
(c) g(0)=g(1)=0
g(n)=g(n-1)+g(n-2)+2, n>=2

4. prefix: -/-*-ABCD+EFG

postfix: AB-C*D-EF+/G-



7.

```
void Permutations (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
               Permutations(a, k+1, m);
               swap(a[k], a[i]);
            }
     } // end of else
} // end of permutation( )
```

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;
}
```