

Department of Computer Science and Engineering
National Sun Yat-sen University
Data Structures - Middle Exam, Nov. 17, 2014 參考解答

1. Explain each of the following terms in C++ language. (12%)

(a) Protected

Ans: 用來保護class內部的成員(變數與函式)，只有class內部成員與其繼承者可使用。

(b) Constructor

Ans: class的建構子，與class名稱相同的函式，當創建class物件時，可使用constructor來初始化物件內容。

(c) operator overloading

Ans: 運算子多載，由使用者重新定義相同class的物件之間運算子的運算模式。

2. A *upper triangular* array a is an n -by- n array in which $a[i][j]=0$, if $i>j$. Suppose that array a is stored in one-dimensional array b sequentially with $a[0][0]$ being stored in $b[0]$. In other words, the sequence for storing in array b is $a[0][0]$, $a[0][1]$, ..., $a[0][n-1]$, $a[1][1]$, $a[1][2]$, ..., $a[1][n-1]$, $a[2][2]$, $a[2][3]$, ..., $a[2][n-1]$, Calculate the addressing formula for the element $a[i][j]$ stored in $b[k]$ in the upper triangular part. (10%)

Ans: $k = n+(n-1)+\dots+(n-i+1)+(j-i) = (n+n-i+1)i/2+(j-i) = in+j-i(i+1)/2$

3. Please give a method, with the help of a stack, to check whether an arithmetic expression containing multiple types of parentheses is valid or not. For example, $\{x + (y - [a+b]) * c - [(d+e)]\} / (h-j)$ is valid, but $[(d+e)]$ is invalid. (10%)

Ans:

(1)遇到左括號，則 push 放入到 stack 中。

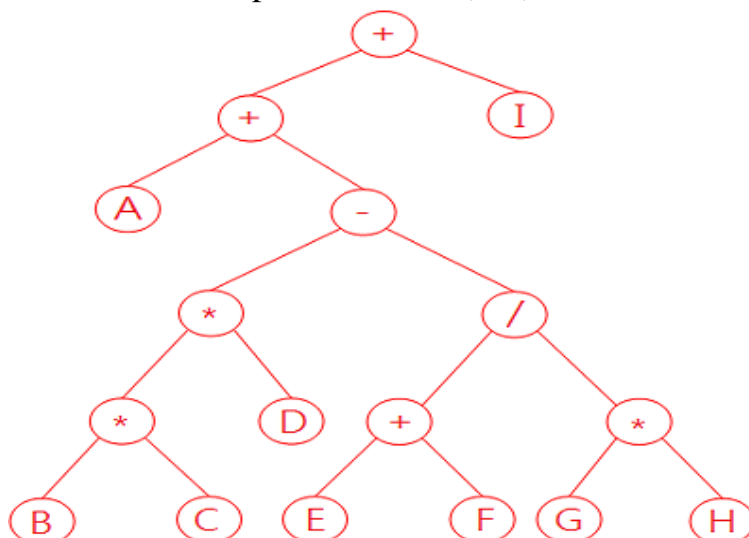
(2)遇到右括號，如果 stack 的 top 是與其匹配的左括號，便將左括號 pop。

(3)如果遇到右括號，但是 stack 是空的，或是 top 跟右括號並不匹配，則此敘述不合法。

(4)如果整個敘述已檢視完，但是 stack 不是空的，則此敘述不合法。

4. Transform the *prefix* expression $++A-**BCD/+EF*GHI$ to *infix* and *postfix* expressions. Draw its expression tree. (9%)

Ans:



Infix: $A+[B*C*D-(E+F)/(G*H)]+I$

Postfix: $ABC*D*EF+GH*/-+I+$

5. What are printed by each of the following C programs? (16%)

(a) `char c=13; printf("%d\n",~((c >> 3) << 2));`

Ans: -5

$c = (00001101)_2 \rightarrow (c \gg 3) = (00000001)_2 \rightarrow ((c \gg 3) \ll 2) = (00000100)_2 \rightarrow$
 $\sim ((c \gg 3) \ll 2) = (11111011)_2 = -5$

(b) `void f(int a[], int b[], int *c, int *d)`

`{ printf("%d %d %d %d\n", a[2],b[2],*(c+2),d[2]); }`

`void main()`

`{ int e[]={10,11,12,13,14,15,16,17,18,19,20};`

`f(e,e+2,&e[3],&e[3]+2); }`

Ans: 12 14 15 17

$a[2]=*(e+2)=12$

$b[2]=*((e+2)+2)=14$

$*(c+2)=*((e+3)+2)=15$

$d[2]=*(((e+3)+2)+2)=17$

(c) `int c[]={10,11,12,13,14}; int *r;`

`r=c+1; *(r++)=c[0]+5;`

`printf("%d %d %d %d\n",c[1],c[2],*r,*r+1);`

Ans: 15 12 12 13

$*r=c[1]$

$c[1]=c[0]+5=15$

$r++ \rightarrow *r=c[2]=12$

$*(r+1)=c[3]=13$

(d) `union {`

`char m;`

`unsigned char n;`

`}u;`

`u.n=197;`

`printf("%d\n",u.m);`

Ans: -59

$u.n=197=(11000101)_2$

因為 $u.m$ 為 `unsigned`， $u.m=(11000101)_2=-59$

6. How do you implement the *set* operations (including *intersection*, *union*, *difference* and *containment*) in a programming language? Give examples to illustrate your implementation. (10%)

Ans:

集合可用 `bit string` 表示，每一個 `bit` 代表相對應的元素，如果 `bit` 之值為 1，則相對的元素在集合內，0 則表示不在集合內。

假設set是1~10之整數的集合

set A={1,3,5,6,8}，其bit string為 1010110100

set B={2,3,6,7,9}，其bit string為 0110011010

set C={1,3,5}，其bit string為 1010100000

Intersection 交集:

$$A \cap B = A \text{ AND } B = 0010010000 = \{3,6\}$$

Union 聯集:

$$A \cup B = A \text{ OR } B = 1110111110 = \{1,2,3,5,6,7,8,9\}$$

Difference 差集:

$$A - B = A \text{ AND } (\text{NOT } B) = 1000100100 = \{1,5,8\}$$

Contain 包含:

$$A \supseteq B \Leftrightarrow A \cap B = B$$

$$A \cap C = A \text{ AND } C = 1010100000 = C \Leftrightarrow A \supseteq C$$

7. The *Fibonacci* sequence is defined recursively as follows:

$$f(n) = n, \quad \text{if } n = 0, 1$$

$$f(n) = f(n-1) + f(n-2), \quad \text{if } n \geq 2.$$

Assume that $f(0)$ and $f(1)$ are given.

(a) Suppose we use an iterative method to compute $f(n)$. How many additions are required? (3%)

$$\text{Ans: } g(0)=g(1)=0$$

$$g(n)=n-1, n \geq 2$$

推算過程如下：

$$f(0) \rightarrow 0 \text{次}$$

$$f(1) \rightarrow 0 \text{次}$$

$$f(2)=f(0)+f(1) \rightarrow 0+1=1 \text{次}$$

$$f(3)=f(2)+f(1) \rightarrow 1+1=2 \text{次}$$

$$f(4)=f(3)+f(2) \rightarrow 2+1=3 \text{次}$$

.....

$$\text{可推得, } g(n)=g(n-1)+1=n-2+1=n-1 \text{次, } n \geq 2$$

(b) Suppose our program is written recursively for computing $f(n)$. How many additions are required? Please derive a general pattern. (6%)

$$\text{Ans: } g(0)=g(1)=0$$

$$g(n)=g(n-1)+g(n-2)+1, n \geq 2$$

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

```
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);  
};
```

Ans:

```
void Permutations (char *a, const int k, const 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 perm
```

9. Let $x=(x_1, x_2, \dots, x_n)$ and $y=(y_1, y_2, \dots, y_m)$ be two circular chains. Write a C++ function to merge the two circular chains together to obtain the circular chain $z=(x_1, y_1, x_2, y_2, \dots, x_n, y_n, y_{n+1}, \dots, y_m)$, where $n \leq m$. (12%)

```
class ChainNode {  
  int data;  
  ChainNode *link;  
};  
class Chain {  
  ChainNode *first *last; // circular chain  
  Chain merge(Chain &y )  
  // Merge two circular chains *this (x) and y into a  
  // single circular chain.  
  {  
    Chain z; // The resulting chain
```

```
Please write the body of merge ( ).
```

```
return z;  
} // end of merge ( )
```

```
};
```

Ans:

```
Chain merge(Chain &y)
```

```
{
```

```
    Chain z;
```

```
    ChainNode *ptr_x = first;
```

```
    ChainNode *ptr_y = y.first;
```

```
    ChainNode *temp;
```

```
    while(ptr_x->link != first)
```

```
    {
```

```
        temp = ptr_x;
```

```
        ptr_x = ptr_x->link;
```

```
        temp->link = ptr_y;
```

```
        temp = ptr_y;
```

```
        ptr_y = ptr_y->link;
```

```
        temp->link = ptr_x;
```

```
    }
```

```
    ptr_x->link = ptr_y;
```

```
    y.last->link = first;
```

```
    z.first = first;
```

```
    z.last = y.last;
```

```
    return z;
```

```
}
```