108-1 資料結構 第一次小考 答案

1. (1) An algorithm is a finite set of instructions that accomplished a particular task (3%)

(2) input, output, definiteness, finiteness, effectiveness (3%)

(3) An ADT is a data type that is organized in such a way that the specification of the objects and the operations on the objects is separated from 1.the representation of the object. 2.the implementation of the operations (3%)

2. (6%)

	space	time
2D Array	O(rows * cols)	O(rows * cols)
Transpose a Matrix	O(elements)	O(cols * elements)
Fast Transpose Matrix	O(elements+MAX_COL)	O(col+elements)

3.

Table2 (2%)

0	1	2	3	4	5
3	1	2	1	0	1

Table3 (3%)

0	1	2	3	4	5
1	4	5	7	8	8

4. (3%) (3%)

Table_1 Table_2

	row	col	value
a[0]	6	6	11
a[1]	0	1	13
a[2]	0	4	9
a[3]	1	0	8
a[4]	1	2	-9
a[5]	2	0	5
a[6]	2	5	-7
a[7]	3	1	7
a[8]	4	2	8
a[9]	4	4	1
a[10]	5	0	2
a[11]	5	3	18

	row	col	value
b[0]	6	6	11
b[1]	0	1	8
b[2]	0	2	5
b[3]	0	5	2
b[4]	1	0	13
b[5]	1	3	7
b[6]	2	1	-9
b[7]	2	4	8
b[8]	3	5	18
b[9]	4	0	9
b[10]	4	4	1
b[11]	5	2	-7

5.(a)top—　　(4%)
(b) top++　　(4%)

6.(1)ABC*+DE/-　　(5%)
(2)-+A/-BCD/E*FG　　(3)A+(B*C)/(E-2)*F
(4)(A+B)/(C-D)*E　　(5)/*+A-BCD*F+GH

7.　　(6%)
$O(n^n) > O(n!) > O(3^n) > O(2^n) > O(n^3) > O(n^2) > O(n log^2 n) > O(n log n^2) > O(n log n) > O(2n) > O(n) > O(\sqrt{n}) > O(1)$

8.　　(5%)
(1) $O(log n)$
(2) $\frac{n(n+1)(2n+1)}{6}$
(3) $[\frac{n(n+1)}{2}]^2$
(4) $O(5^n)$
(5) $O(1.5^n)$

9.　　(6%)
n-1

10.
(1)(a)no　　(1%)
(b)no　　(1%)
(c)no　　(1%)
(2)1234, 2134, 3214, 1243, 2143, 3241, 1324, 2341, 3421, 1342, 2314, 4321, 1432, 2431　　(3%)

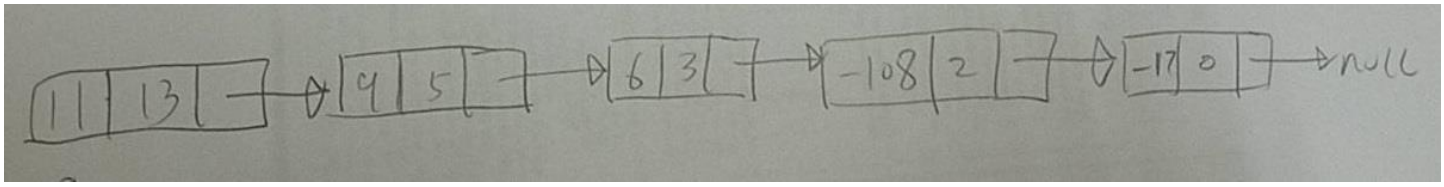
11.(1)　　(3%)

Structuretureureeeeuts

(2)O(1)　　(5%)

12.

(1)(1%)



(2)

(a)　　(3%)

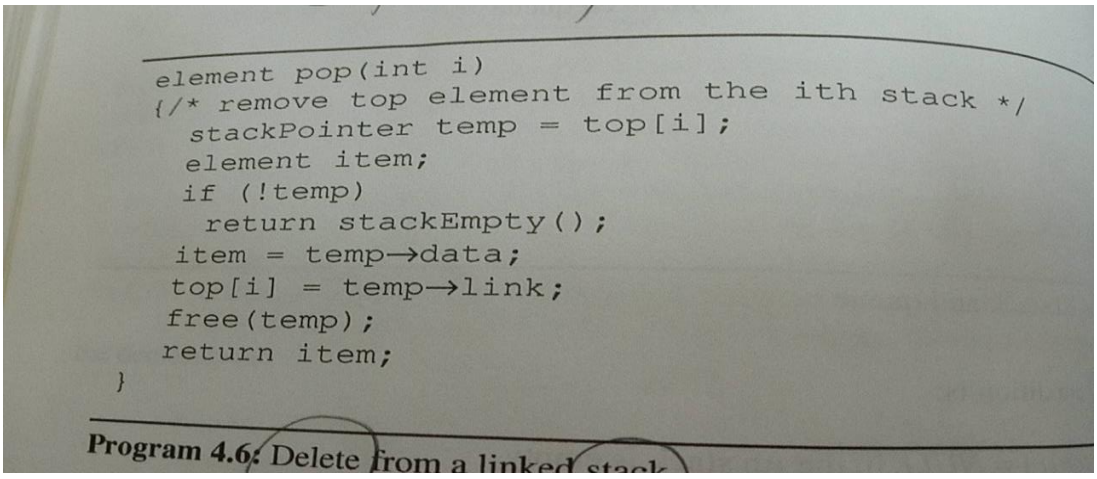
```
sum = a->coef + b->coef;
if (sum)
   attach (sum , a->expon , &rear);
   a = a->link ;
   b = b->link;
   break;
```

(b)　　(3%)

```
attach(a->coef,a->expon,&rear);
a = a->link;
break;
```

13.

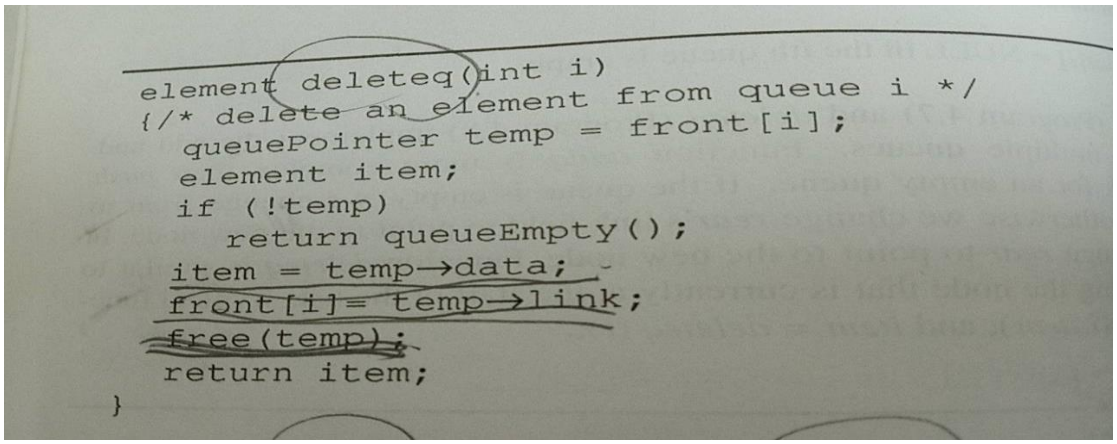
(a)　　(4%)



```
element pop(int i)
{/* remove top element from the ith stack */
  stackPointer temp = top[i];
  element item;
  if (!temp)
    return stackEmpty();
  item = temp→data;
  top[i] = temp→link;
  free(temp);
  return item;
}
```

Program 4.6: Delete from a linked stack

(b) (4%)



```
element deleteq(int i)
{/* delete an element from queue i */
  queuePointer temp = front[i];
  element item;
  if (!temp)
    return queueEmpty();
  item = temp->data;
  front[i]= temp->link;
  free(temp);
  return item;
}
```

14.

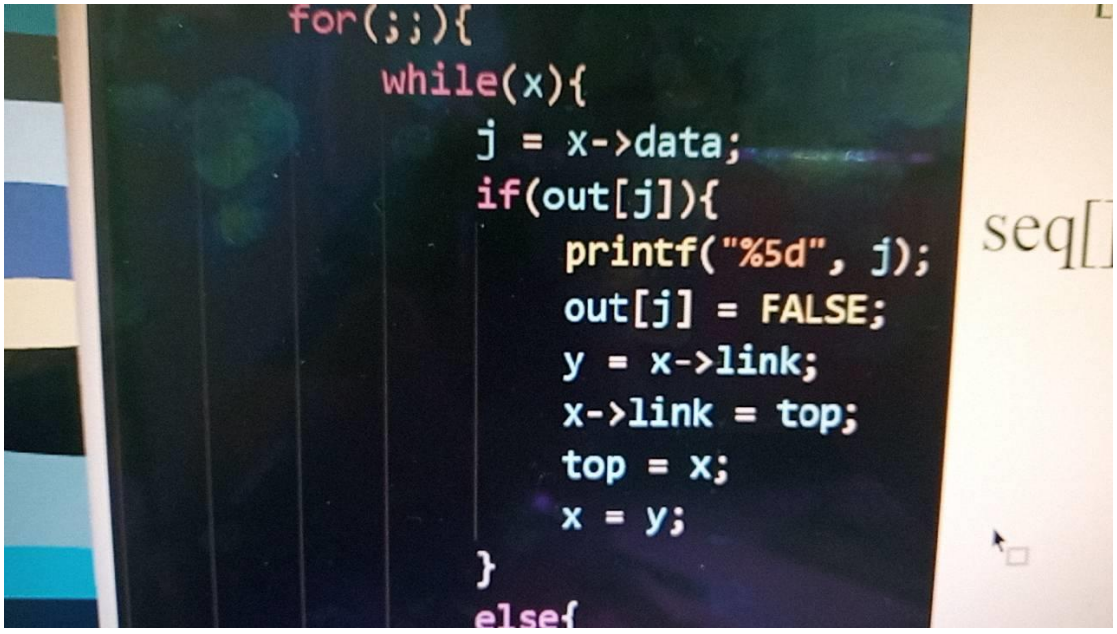
(a) current->next = previous; (2%)

(b) previous = current; (2%)

(c) current = preceding; (2%)

(d) preceding = preceding->next; (2%)

15. (7%)



```
for(;;){
    while(x){
        j = x->data;
        if(out[j]){
            printf("%5d", j);
            out[j] = FALSE;
            y = x->link;
            x->link = top;
            top = x;
            x = y;
        }
        else{
```