109-1 Data Structure Quiz 1

學號:__________姓名:__________

1. Given a Define as below. (note that O and o is different.)

Def:

$f(n) = o(g(n))$ $\quad iff\ for\ any\ positive\ constant\ c,$

$there\ exists\ an\ positive\ constant\ n_0$

$s.t.\ f(n) < c * g(n)\ for\ all\ n, n \geq n_0$

$f(n) = \omega(g(n))$ $\quad iff\ for\ any\ positive\ constant\ c,$

$there\ exists\ an\ positive\ constant\ n_0$

$s.t.\ f(n) > c * g(n)\ for\ all\ n, n \geq n_0$

_____(a) Choose the correct answer(s). (5%)

(A) $2^n = O(2^n)$.

(B) $if\ f(n) = O(g(n))\ then\ g(n) = O(f(n))$.

(C) $(\log n)! = O(n^{100})$.

(D) $if\ f(n) = O(g(n))\ then\ 2^{f(n)} = O(2^{g(n)})$

(E) $if\ f(n) = O(g(n))\ then\ g(n) = \Omega(f(n))$.

_____(b) Choose the correct answer(s). (5%)

(A) $2n^2 = o(n^2)$.

(B) $2n^2 = O(n^2)$.

(C) $2n^2 = o(n^3)$.

(D) $f(n) + g(n) = \theta(Max(f(x), g(n))$.

(E) $if\ f(n) = \omega(g(n))\ then\ g(n) = o(f(n))$.

2. Sort the following options with the time complexity(O) from low to high. (5%)

(a) $(n-1)!$ (b) $\log(n!)$ (c) n^5 (d)n^n (e)$\log^2 n$ (f)$\sqrt{2}^{logn}$

Ans:______________

3. There is a function defined as follows.

$$A(m,n) = \begin{cases} n+1, & if\ m=0 \\ A(m-1,1), & if\ n=0 \\ A(m-1, A(m,n-1)), & otherwise \end{cases}$$

(a) Write a recursive function in C. (2%)

```
int ak(int m, int n){





}
```

(b) $A(3,2) = ?$ (3%)

Ans:____________

(c) How many times will it call the function ak() when m = 2, n = 4? (includes the first time) (5%)

Ans:____________

4. Sparse matrix

$$\text{Matrix A} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 3 & 1 \\ 8 & 0 & 7 \\ 0 & 5 & 0 \end{pmatrix}$$

(a) Use 3-tuple represent the matrix A. (3%)

(b) Use the answer of (4.a) to find a transpose matrix. (3%)

____(c) Does using 3-tuple save more space? (Answer True or False) (4%)

5. Given a set of n elements (n≥1), function P will print out all possible permutations of this set.

For example, the permutations of {a, b, c} are

{ (a, b, c), (a, c, b), (b, a, c), (b, c, a), (c, a, b), (c, b, a) }

```
void P(char* list, int i){
    size_t length = strlen(list); //get the size of list
    if(i == length){
        for(int i = 0;i<length;i++) printf("%c", list[j]);
        printf("\n");
    }
    else{
        for(j=i; j< (a) (2%) ; j++){
            swap( list[i] , list[j]); // The swap function will exchange the element of parameter.
            P(   (b) (3%)   );
            swap( list[i] , list[j]);
        }
    }
}
void main(){
    // Assume list has been initialized and the size of list is n.
    P(&list,0);
}
The time complexity of this algorithm is:   (c) (5%) .
```

Please finish the function.

Ans: (a)_____ (b)_______ (c) O(_______)

6. Please implement Queue by Stack in c, write down the functions of Queue. (max_size = 10)

struct stack{
    int max_size;
    int *items;
    int top;
};
typedef enum {false, true} bool;
	struct stack* create(max_stack_size);
	bool isFull(stack);
	bool isEmpty(stack);
	void push(stack, item);
	int pop(stack);

(2%)struct queue{
 	
};
	(2%)queue createQ(unsigned max_size){
	
	}

(1%)bool isFullQ(struct queue *q){
 	
}
	(1%)bool isEmptyQ(struct queue *q){
	
	}

(3%)void addQ(struct queue *q, int item){
 	
}
	(3%)int deleteQ(struct queue *q){
	
	}

7. Write the prefix form of the following expression. (5%)

(A + B) * C / (D && E) ^ F << G

Ans: ____________________________

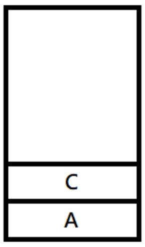
8. Write the postfix form of the following prefix expression. (5%)

*** + X Y / Z % N << T && R B**

Ans: ____________________________

9. Assume the string of "ABCDEF" are sequentially pushed into a stack where elements in the stack can be popped during the period of pushing the letters. The state of the stack is shown as below and the next char is ‘F’. What is the possible outcome of the stack permutation? (3%)

Ans:



10. Following C programs to write the code for the questions below. This code is the structure of each node.

```
struct pNode{
        float coef;
        int expon;
        struct pNode *link;
};
```

(a) Insert $3x^5$ between $2x^7$ and $9x^0$. (2%)

```
//pNode *head points to a pNode 2x^7 -> 9x^0 -> NULL
```

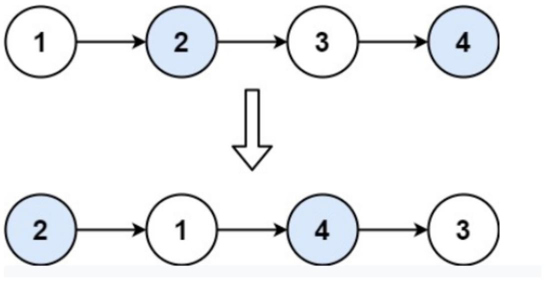
(b) Delete the tail of the linked list. (3%)

```
//pNode *head points to a pNode c_1x^a1 -> … -> c_nx^an -> NULL
```

(c) Finish a function that differentiates the polynomial function. (5%)

```
// For example, 2x^4 -> 4x^3 will become 8x^3-> 12x^2.
void differential(struct pNode *head) {









}
```

11. Given a linked list, swap every two adjacent nodes and return its head. (10%)
※You cannot modify the values in the list's nodes.



```
/* struct ListNode{
       Int val;
       struct ListNode *next; };*/
struct ListNode swapPairs(struct ListNode *head) {









}
```

12. According to the following code (in 64-bit), fill in the blanks. (10%)

```
int main(){
        int mark[4] = {9,5,2,7};
        int a = 50;
        int *b = a;
        int *c = &a;
        int *d[4];
        int (*e)[4] = mark;
        int **f = e;
        d[0] = d;
        d[1] = b;
        d[2] = c;
        printf("mark’s position: %d\n", &mark);
        printf("a1:%d, a2:%d\n", a, &a+10);
        printf("b1:%d, b2:%d\n", b, &b);
        printf("c1:%d, c2:%d\n", c, &c);
        printf("d1:%d, d2:%d, d3:%d, d4:%d\n", *d[0], &d[1], d[2], &d);
        printf("e1:%d, e2:%d\n", e[2], &e);
        printf("f1:%d, f2:%d\n", *f+2, &f);
        return 0;
}
```

Output:

mark’s position: 5000
a1:50, a2:4000
b1:__(a)__, b2:3000
c1:__(b)__, c2:2000
d1:1000, d2:1008, d3:__(b)__, d4:__(c)__
e1:__(d)__, e2:500
f1:__(e)__, f2:250

Ans: (a)______ (b)_______ (c)_______ (d)_______ (e)________