1	2	3	4	5	6	7	8	9	10

Data Structure Midterm Examination

3:30pm-5:20pm (110 minutes), April. 23, 2018

#: ________ Student ID: __________________ Name: ________________________

✧ 看清題目再作答。

✧ 中英文、鉛筆原子筆做答都可以。

✧ 如有多個正確

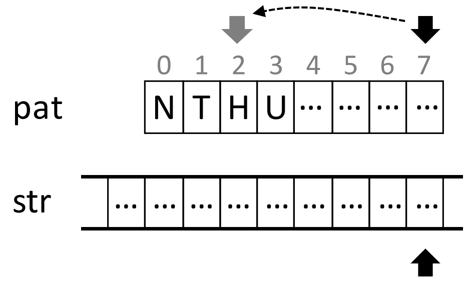
✧ 答案但題目指要求回答一個，回答任一個均可。

✧ There are 10 questions. You can obtain up to 114 points.

---

1. **(8pt) KMP Algorithm**

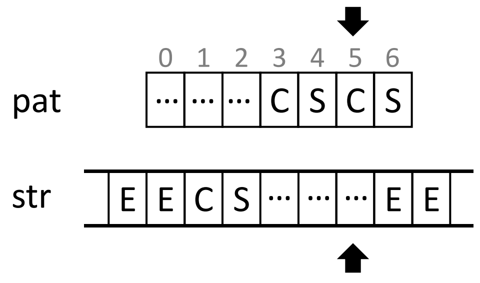
   (1) Please show a pattern (NTHU…) that lets KMP Algorithm try to match pat[2] upon a mismatch at pat[7] as shown in the following figure.



Ans: **pat** =

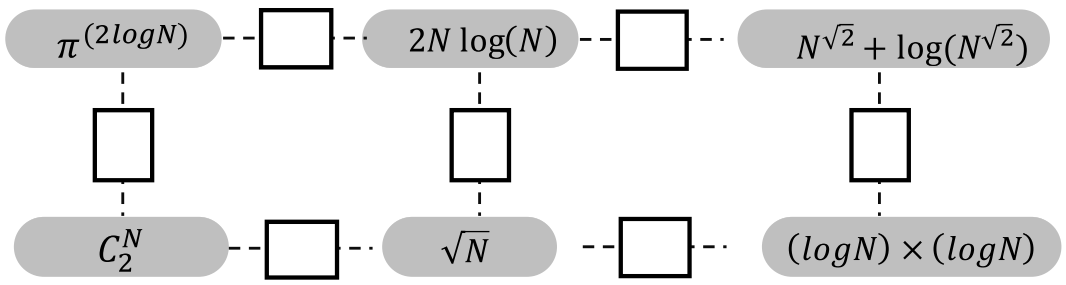
0	1	2	3	4	5	6	7
N	T	H	U				

   (2) What is the next index (0 ~ 6) KMP Algorithm would try to match upon a mismatch at pat[5] as shown in the following figure?



Ans: **next index** = _____________

2. **(14pt)** Please compare the asymptotic complexity hierarchy of the following functions using ‘=’, ‘>’, or ‘<’.



3. **(10pt)** Please complete the function that inserts a singly linked list (non-circular, with a header) into a **doubly linked list** (non-circular, with a header) right after the node pointed by the pointer P. Please assume the handling pointer of the doubly linked list is “first”.

```
void DoublyList::InsetSinglyList(Node* List, Node* P)
{

}
```

4. **(10pt)** Please analyze the **time complexity** of the following functions as tight as possible:

```
void f(int X, int Y, int a[])
{
   while (Y > 0) {
      if(X <= 0) {
         X = Y = Y - 1;
      }
      X = X - 1;
      a[Y]++;
   }
}
```

```
void g(int N, int a[], int b[])
{
   if (a[N] % N == 0)
      return;
   for (int i=0; i<N; i=i*2)
      b[N]++;
   g(N-1, a, b);
}
```

f(X, Y, …) ∈ Ω(　　　　　　　　)　　**g(N, …) ∈ Ω(　　　　　　　　)**

f(X, Y, …) ∈ O(　　　　　　　　)　　**g(N, …) ∈ O(　　　　　　　　)**

5. **(20pt)** Select the best answer (-1 points for each wrong selection)

_____ Which can cause operating systems (e.g., Linux and Windows) to report a runtime error?

(A) Reference to an out-of-bound array entry.

(B) Miss a semicolon (分號) in a C program.

(C) Reference to a variable name that is undefined.

(D) Duplicate functions or variables in a program.

(E) None of the above.

_____ Which corresponds to the slowest algorithm when the problem size is large enough?

(A) **O**(n^4)

(B) **O**($0.1(4^n)$)

(C) **O**($10\log(4^n)$)

(D) **O**($(4n)^n$)

(E) None of the above is the best answer because it depends on the implementation of the algorithm.

_____ Which programming language can solve the largest set of problems.

(A) Structure language (e.g. , C)

(B) Object-oriented language (e.g., C++)

(C) Assembly language

(D) The above can solve the same set of problems.

(E) The answer depends on compilers and operating systems.

_____ To store a polynomial **f(x, y, z)** = $2x^3y^4 + 3x^2z^3 - 7xy^2z^2 + ...$ using an array.

(A) array size = 3 * terms

(B) array size = 4 * terms

(C) array size = terms^2

(D) array size = terms^3

(E) None of the above

_____ int A[10][10][10] is an row-major array. What is the offset of A[3][3][3]?

(A) 3 * (1000 + 100) * sizeof(int)

(B) 3 * (1000 + 100 + 10 + 1) * sizeof(int)

(C) 3 * (100 + 10 + 1) * sizeof(int)

(D) 3 * (1000 + 100 + 10) * sizeof(int)

(E) None of the above

6. **(12pt)** Please convert **B * C + (A * B / C) – A / B + C** into a postfix expression**.** Only the boxes with thick borders will be graded (只有粗黑框格子計分). Please note **the priority listed in the table is not what we commonly use.**

← bottom

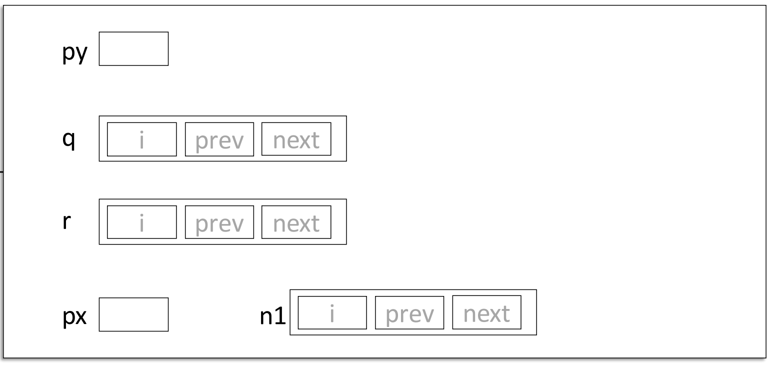
Token	Stack	Output So Far
B		
*		
C		
+		
(		
A		
*		
B		
/		
C		
)		
-		
A		
/		
B		
+		
C		
	Final output	

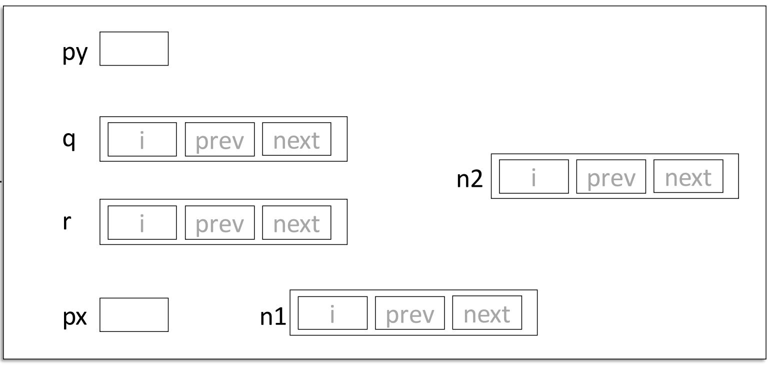
Priority	Operator
High	+
	-
Low	* /

7. (10pt) Linked list operations

```
struct Node{
  int i;  Node* prev;  Node* next;
  Node(int i, Node* prev, Node* next):i(i), prev(prev), next(next) { }
};
Node * px, *py;
Node q(3, 0, 0), r(4, 0, 0);
px = &r;
px = new Node(10, px, &q);  // assume n1 is allocated

px->next->next = new Node(20, px, px); // assume n2 is allocated
q.prev = &r;
r.prev = px;
q.prev->next = q.prev;
py = px->next->prev->next;
```





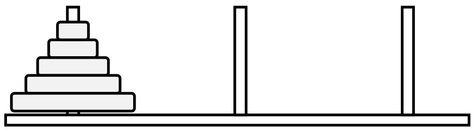
8. (10pt) Asymptotic notations

$$P(n) = \boldsymbol{a_0} + \boldsymbol{a_1 n} + \boldsymbol{a_2 n^2} + \cdots + \boldsymbol{a_k n^k} \;\; \Rightarrow \;\; \log\big(P(n)\big) = \boldsymbol{O}(n)$$

(1) **(4pt)** Is the above statement **true or false**?

(2) **(6pt)** Please prove or disprove the statement **according to the definition of Big-O**.

9. **(11pt) Hanoi Tower**



(1) **(5pt)** Suppose the smallest disk weighs 1 gram, the 2nd smallest one weighs 2 grams, …, the Nth one weighs 2^N grams, how many grams are lifted in total to move the N disks to another stick?

(2) **(6pt)** Please write a **recursive program** that simulates Hanoi Tower to calculate the above value.

10. **(9pt)**

(1) **(3pt)** Please describe two differences between **call-by-value** and **call-by-pointer**?

(2) **(3pt)** Please explain two purposes of **templates** in C++.

(3) **(3pt)** A multiplication operation is usually slower than an addition operation, but why can we still consider them the same when performing step count analysis?