

## C Programming Language Exam 2015/9/15(120 minutes)

1. Which statement is true ?  
(A) The array " a[i] " equals to the array " i[a] ".  
(B) The " scanf " function's return value is a boooling value, which means you successfully or failed input a value.  
(C) The latter "☆" represent "74" in ASCII value.  
(D) The value of c in the following statement , " i=1, c=(double)((((++i\*i\*=i/=-i)&!i)%=i++)) ", is 26.
2. What are the differences between a linked list and a stack?
3. What are the differences between a stack and a queue?
4. Greatest common divisor(gcd) can be defined by the following equations:  $\text{gcd}(a,0)=a$  and  $\text{gcd}(a,b)=\text{gcd}(b,a\%b)$   
Please write a recursive function and another non-recursive function to compute gcd of two integers.
5. Big number : write a program to do the large number operation  
a)  $275757506275757565303 + 653036520342634264$   
b)  $314159265358979323846 - 28825252$
6. You get a letter from a spy. The letter is encrypted, please write a program to decrypted the letter. The encryption's way is that every alphabet excluding punctuations is move backward 5 times, for instance, a -> f , Y->D, and the content is preposterous(前後顛倒的). Notice the capital and the lower case of the alphabet! You can assume that there are no more than 100 chars in the letter.

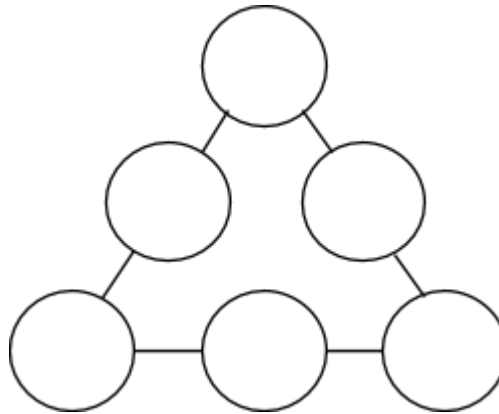
Following is an example:

!yxjg wztd dwY !lsntl ujjP

7. Please write a program to fill the blanks with number 1 to 6. And show “all the possibilities”.

Rules :

1. Each number can only be used one time.
2. The sum of each side of the triangle should be the same.



8. Please write a program to match numbers of the receipt. You need to output the prize and the award money if you win the prize. Notice that one receipt can only win one prize, if you win many prizes with one receipt, you can only choose the biggest one. Following table is the rule.

Prize	Rules	Award money
特別獎	Have the exact same 8 digits to one 8-digit number	10,000,000
特獎	Have the exact same 8 digits to another 8-digit number	2,000,000
頭獎	Have the exact same 8 digits to another three 8-digit numbers	200,000
二獎	Have same last 7 digits to the three numbers above	40,000
三獎	Have same last 6 digits to the three numbers above	10,000
四獎	Have same last 5 digits to the three numbers above	4000
五獎	Have same last 4 digits to the three numbers above	1000
六獎	Have same last 3 digits to the three numbers above	200
增開六獎	Have same last 3 digits to the other three 3-digit numbers	200

9. write a statement or set statements to accomplish each of the following. Assume that all the manipulations occur in `main` (therefore, no addresses of pointer variables are needed), and assume the following definitions:

```
struct gradeNode {
    char lastname[ 20 ];
    double grade;
    struct gradeNode *nextPtr;
};
typedef struct gradeNode GradeNode;
typedef GradeNode GradeNodePtr;
```

- a) Create a pointer to the start of the list called `startPtr`. The list is empty.
- b) Create a new node of type `GradeNode` that is pointed to by pointer `newPtr` of type `GradeNodePtr`. Assign the string "Jones" to member `lastname` and the value 91.5 to member `grade` (use `strcpy`). Provide any necessary declarations and statements.
- c) Assume that the list pointed to by `startPtr` currently consist of 2 nodes—one containing "Jones" and one containing "Smith". The nodes are in alphabetical order. Provide the statements necessary to insert in order nodes containing the following data for last name and grade:

"Adams"	85.0
"Thompson"	73.5
"Pritchard"	66.5

Use pointers `previousPtr`, `currentPtr` and `newPtr` to perform the insertions. State what `previousPtr` and `currentPtr` point to before each insertion. Assume that `newPtr` always points to new node, and that the new node has already been assigned the data.

- d) Write a `while` loop that prints the data in each node of the list. Use pointer `currentPtr` to move along the list.
- e) Write a `while` loop that deletes all the nodes in the list and frees the memory associated with each node. Use pointer `currentPtr` and pointer `tempPtr` to walk along the list and free memory, respectively.

10. Please describe what is memory leak and how to prevent.

11. Given a sequence of integers  $S = \{S_1; S_2; \dots; S_n\}$ , you should determine what is the value of the maximum positive product involving consecutive terms of  $S$ . If you cannot find a positive sequence, you should consider 0 as the value of the maximum product.

**Input:** The test case starts with  $1 \leq N \leq 18$ , the number of elements in a sequence. Each element  $S_i$  is an interger such that  $-10 \leq S_i \leq 10$ . Next line will have  $N$  integers, representing the value of each element in the sequence.

**Output:** print the maximum product.

**Sample Input:**

(1)  
6  
3 5 -1 2 -6 -1  
(2)  
3  
-9 -7 -8

**Sample Output:**

180  
  
63

12. Let  $A$  be an  $N \times N$  ( $1 \leq N \leq 25$ ) matrix of zeros and ones. A submatrix  $S$  of  $A$  is any group of *contiguous* entries that forms a square or a rectangle.

Write a program that determines the number of elements of the largest submatrix of ones in  $A$ . *Largest* here is measured by area.

**Sample input:**

10111000 01011110  
00010100 00011110  
00111000  
00111010  
00111111  
01011110

**Sample output:**

16

Dec	Hx	Oct	Dec	Hx	Oct	Dec	Hx	Oct	Dec	Hx	Oct
0	0	000	32	20	040	64	40	100	96	60	140
1	1	001	33	21	041	65	41	101	97	61	141
2	2	002	34	22	042	66	42	102	98	62	142
3	3	003	35	23	043	67	43	103	99	63	143
4	4	004	36	24	044	68	44	104	100	64	144
5	5	005	37	25	045	69	45	105	101	65	145
6	6	006	38	26	046	70	46	106	102	66	146
7	7	007	39	27	047	71	47	107	103	67	147
8	8	010	40	28	050	72	48	110	104	68	150
9	9	011	41	29	051	73	49	111	105	69	151
10	A	012	42	2A	052	74	4A	112	106	6A	152
11	B	013	43	2B	053	75	4B	113	107	6B	153
12	C	014	44	2C	054	76	4C	114	108	6C	154
13	D	015	45	2D	055	77	4D	115	109	6D	155
14	E	016	46	2E	056	78	4E	116	110	6E	156
15	F	017	47	2F	057	79	4F	117	111	6F	157
16	10	020	48	30	060	80	50	120	112	70	160
17	11	021	49	31	061	81	51	121	113	71	161
18	12	022	50	32	062	82	52	122	114	72	162
19	13	023	51	33	063	83	53	123	115	73	163
20	14	024	52	34	064	84	54	124	116	74	164
21	15	025	53	35	065	85	55	125	117	75	165
22	16	026	54	36	066	86	56	126	118	76	166
23	17	027	55	37	067	87	57	127	119	77	167
24	18	030	56	38	070	88	58	130	120	78	170
25	19	031	57	39	071	89	59	131	121	79	171
26	1A	032	58	3A	072	90	5A	132	122	7A	172
27	1B	033	59	3B	073	91	5B	133	123	7B	173
28	1C	034	60	3C	074	92	5C	134	124	7C	174
29	1D	035	61	3D	075	93	5D	135	125	7D	175
30	1E	036	62	3E	076	94	5E	136	126	7E	176
31	1F	037	63	3F	077	95	5F	137	127	7F	177