Assume that all variables without declaration shown in the following program segments have already been declared properly. Integers in problem statements are 32-bit signed variables (**Pascal**: `longint`, **C**: `int`). Assume all the programs are compiled properly without using any compiler flag (except the `"-o"` option in **C**).

	Format	# **Questions**	**Total Marks**
Section A	Multiple Choice	25	25
Section B	Fill-in-the-blanks	7 (A-L)	20
Total			**45**

Section A (25 marks)

For each question, choose the most appropriate answer and mark the corresponding box (**A**, **B**, **C** or **D**) on the answer sheet. One mark for each correct answer. No marks will be deducted for wrong answers.

1. What is the output of the following program?

Pascal Version

```pascal
var
  a, b: longint;
begin
  a := 8;
  b := 2016;
  while (b >= a) do
  begin
    b := b + 4;
    a := a * 2
  end;
  write(b)
end.
```

C Version

```c
#include <stdio.h>
int a, b;
int main() {
  a = 8;
  b = 2016;
  while (b >= a) {
    b = b + 4;
    a = a * 2;
  }
  printf("%d", b);
}
```

A. 2044
B. 2048
C. 2052
D. 2056

2. What is the output of the following program?

Pascal Version

```
var
  sum: longint;
procedure abc(x, y: longint);
begin
  if x = 8 then
    sum := sum + y
  else
  begin
    abc(x + 1, y * 2);
    abc(x + 1, y * 2 + 1)
  end
end;
begin
  sum := 0;
  abc(0, 0);
  write(sum)
end.
```

C Version

```
#include <stdio.h>
int sum;
void abc(int x, int y) {
  if (x == 8)
    sum = sum + y;
  else {
    abc(x + 1, y * 2);
    abc(x + 1, y * 2 + 1);
  }
}
int main() {
  sum = 0;
  abc(0, 0);
  printf("%d", sum);
}
```

A. 16320
B. 32640
C. 65408
D. 130816

3. Alice describes a tree to Bob:

- This tree has 5 vertices.
- The parent of `E` is `B`.
- The parent of `A` is `C`.
- The parent of `B` is `D`.
- The parent of `C` is `B`.

Which of the following is the root of the tree?

A. `A`
B. `B`
C. `C`
D. `D`

4. Let T be a tree with 45 vertices.
Define the depth of a vertex as the number of edges between that vertex and the root. In particular, the depth of the root is 0.
In T, there are exactly $i + 1$ vertices with depth i.

Which of the following is/are true?

i. If the number of leaves is 9, T is a binary tree.
ii. The sum of depths of all vertices is 72.
iii. The maximum number of leaves is 36.

A. i only
B. ii only
C. i and iii only
D. ii and iii only

5. There are n people, P_1, P_2, ..., P_n, where n is an integer greater than 2. Each of them spoke a sentence simultaneously:

P_1: 1 of us is lying.
P_2: 2 of us are lying.
$\ldots$
P_i: i of us are lying.
$\ldots$
P_{n-1}: $n-1$ of us are lying.
P_n: n of us are lying.

Who is the only one that is telling the truth?

A. P_1
B. P_{n-1}
C. P_n
D. None of the above

6. How many different possible outputs are there for the following program? Assume input will always be a 32-bit signed integer.

Pascal Version

```
var
  x: longint;
begin
  read(x);
  write((x mod 5) mod 3)
end.
```

C Version

```
#include <stdio.h>
int x;
int main() {
  scanf("%d", &x);
  printf("%d", (x % 5) % 3);
  return 0;
}
```

A. 2
B. 3
C. 4
D. 5

7. Concerning only the worst case performances of the following sorting algorithms, which algorithm has the best time complexity?

A. Merge Sort
B. Bubble Sort
C. Quick Sort
D. Selection Sort

8. Which of the following data structures support(s) binary search?

i. Stack
ii. Queue
iii. Linked list

A. iii only
B. i and ii only
C. i, ii and iii
D. None of the above

9. Consider the following program:

Pascal Version

```
var
  n, i: longint;
begin
  read(n);
  for i := 2 to n - 1 do
  begin
    if (i mod n = 0) then
    begin
      write('Composite');
      halt
    end
  end;
  write('Prime')
end.
```

C Version

```
#include <stdio.h>
int n, i;
int main() {
  scanf("%d", &n);
  for (i = 2; i <= n - 1; i++) {
    if (i % n == 0) {
      printf("Composite");
      return 0;
    }
  }
  printf("Prime");
  return 0;
}
```

The input `n` is always an integer between 2 and 100000000 (inclusive). Which of the following statements is correct?

A. Although it is not the most efficient way to do so, the program correctly determines whether `n` is a prime or a composite

B. The program may output nothing

C. There exists a valid input `n` that will cause runtime error.

D. When `n` is a square number, such as 4, 9, 16, ..., the program always output `"Prime"`

10. Cancelled due to inconsistency between English and Chinese versions

For questions 11 to 12, consider the following program:

Pascal Version

```
var
  i, j: longint;
  tri: array[0..14, 0..14] of longint;
begin
  for i := 0 to 14 do
    for j := 0 to 14 do
      tri[i][j] := 0;
  tri[0][0] := 1;
  for i := 1 to 14 do
  begin
    tri[i][0] := 1;
    tri[i][i] := 1;
    for j := 1 to i - 1 do
      tri[i][j] := tri[i - 1][j] +
         tri[i - 1][j + 1]
  end;
  write(tri[7][1], ' ', tri[13][7])
end.
```

C Version

```
#include <stdio.h>
int tri[15][15], i, j;
int main(){
  for (i = 0; i <= 14; i++)
    for (j = 0; j <= 14; j++)
      tri[i][j] = 0;
  tri[0][0] = 1;
  for (i = 1; i <= 14; i++) {
    tri[i][0] = 1;
    tri[i][i] = 1;
    for (j = 1; j <= i - 1; j++)
      tri[i][j] = tri[i - 1][j] +
         tri[i - 1][j + 1];
  }
  printf("%d %d", tri[7][1],
     tri[13][7]);
  return 0;
}
```

11. What is the first output number?
   A. 7
   B. 13
   C. 21
   D. 1716

12. What is the second output number?
   A. 7
   B. 13
   C. 21
   D. 1716

13. Consider the following program:

Pascal Version

```
var
  n, i, sum: longint;
begin
  read(n);
  sum := 0;
  for i := 0 to n do
    sum := sum + (i and (n - i));
  write(sum)
end.
```

C Version

```
#include <stdio.h>
int n, i, sum;
int main() {
  scanf("%d", &n);
  sum = 0;
  for (i = 0; i <= n; i++)
    sum = sum + (i & (n - i));
  printf("%d", sum);
  return 0;
}
```

Which of the following input n would give the smallest output?
   A. 31
   B. 32
   C. 33
   D. 34

14. What is the output of the following program?

Pascal Version

```
var
  i, sum:  longint;
begin
  sum := 0;
  for i := 0 to 100 do
    sum := sum + (i xor 7);
  write(sum)
end.
```

C Version

```
#include <stdio.h>
int i, sum;
int main(){
  sum = 0;
  for (i = 0; i <= 100; i++)
    sum = sum + (i ^ 7);
  printf("%d", sum);
  return 0;
}
```

A. 5049
B. 5050
C. 5065
D. 5066

15. Which of the following can be implemented using one or more arrays? Assume that the arrays are large enough.

i. Stack
ii. Queue
iii. Linked List

A. i only
B. ii only
C. i and ii only
D. i, ii and iii

16. Considering the following program: (Variable declarations and implementations of `P_push`, `P_pop`, `Q_push`, `Q_pop` are omitted)

Pascal Version

```
var
  i: longint;
  a: array[0..7] of longint;
begin
  for i := 0 to 7 do
    read(a[i]);
  for i := 0 to 7 do
    P_push(a[i]);
  for i := 0 to 7 do
    Q_push(P_pop());
  for i := 0 to 6 do
    write(Q_pop(), ' ');
  write(Q_pop())
end.
```

C Version

```
#include <stdio.h>
int i, a[8];
int main() {
  for (i = 0; i <= 7; i++)
    scanf("%d", &a[i]);
  for (i = 0; i <= 7; i++)
    P_push(a[i]);
  for (i = 0; i <= 7; i++)
    Q_push(P_pop());
  for (i = 0; i <= 6; i++)
    printf("%d ", Q_pop());
  printf("%d", Q_pop());
}
```

Suppose `P`, `Q` are data structures. Suppose functions/procedures `P_push()`, `Q_push()`, `P_pop()` and `Q_pop()` are correctly implemented for the corresponding data structures. The input is an array of 8 integers separated by spaces. Which of the following combinations of `P` and `Q` will output the reversed array?

i. `P`: Stack, `Q`: Stack
ii. `P`: Stack, `Q`: Queue
iii. `P`: Queue, `Q`: Stack
iv. `P`: Queue, `Q`: Queue

A. i only
B. ii only
C. ii and iii only
D. iv only

For questions 17 to 18, consider the following four procedures.

Pascal Version

```
procedure a();
var
  i, j: longint;
begin
  for i := 0 to 10 do
    for j := i + 1 downto 0 do
      write('*')
end;

procedure b();
var
  i, j: longint;
begin
  i := 20;
  while (i >= 0) do
  begin
    j := (i - 3) div 2;
    i := j;
    while (i > 0) do
    begin
      write('*');
      dec(i)
    end;
    dec(i)
  end
end;

procedure c();
var
  i, j, k: longint;
begin
  for i := 1 to 4 do
    for j := 1 to 4 do
      for k := 1 to 4 do
        if (i * j * k <= 20) then
          write('*')
end;

procedure d();
var
  i, j, k: longint;
begin
  for i := 1 to 6 do
    for j := 1 to 5 do
      for k := j to 7 do
        if ((i + j + k) mod 3 = 0) then
          write('*')
end;
```

C Version

```
void a() {
  int i, j;
  for (i = 0; i <= 10; i++)
    for (j = i + 1; j >= 0; j--)
      printf("*");
}

void b() {
  int i, j;
  i = 20;
  while (i >= 0) {
    j = (i - 3) / 2;
    i = j;
    while (i > 0) {
      printf("*");
      i--;
    }
    i--;
  }
}

void c() {
  int i, j, k;
  for (i = 1; i <= 4; i++)
    for (j = 1; j <= 4; j++)
      for (k = 1; k <= 4; k++)
        if(i * j * k <= 20)
           printf("*");
}

void d() {
  int i, j, k;
  for (i = 1; i <= 6; i++)
    for (j = 1; j <= 5; j++)
      for (k = j; k <= 7; k++)
        if ((i + j + k) % 3 == 0)
          printf("*");
}
```

17. Which of the four procedures outputs the least number of "*"s?
   A. a()
   B. b()
   C. c()
   D. d()

18. Which of the four procedures outputs the most number of "*"s?
   A. a()
   B. b()
   C. c()
   D. d()

19. What is the output of the following program?

Pascal Version

```
var
  i, j: longint;
  x: array[0..999] of longint;
begin
  for i := 1 to 999 do
    x[i] := 0;
  for i := 2 to 999 do
    if x[i] = 0 then
    begin
      j := i;
      while (j <= 999) do
      begin
        x[j] := x[j] + 1;
        j := j + i
      end
    end;
  write(x[30] + x[37] +
     x[60] + x[999])
end.
```

C Version

```
#include <stdio.h>

int i, j, x[1000];

int main() {
  for (i = 1; i <= 999; i++)
    x[i] = 0;
  for (i = 2; i <= 999; i++)
    if (x[i] == 0) {
      j = i;
      while (j <= 999) {
        x[j] = x[j] + 1;
        j = j + i;
      }
    }
  printf("%d", x[30] + x[37] +
     x[60] + x[999]);
  return 0;
}
```

   A. 8
   B. 9
   C. 10
   D. 11

20. Consider the following program segment. `sort(x, y)` is a procedure to sort the elements from `a[x]` to `a[y]` (inclusive) in non-decreasing order. (Implementation of procedure `sort` is omitted)

Pascal Version

```
var
  a: array[0..14] of longint;
  i: longint;
begin
  for i := 0 to 14 do
    read(a[i]);
  sort(0, 3);
  sort(6, 10);
  sort(2, 9);
  sort(10, 14)
end.
```

C Version

```
#include <stdio.h>
int a[15];
int i;
int main() {
  for (i = 0; i <= 14; i++)
    scanf("%d", &a[i]);
  sort(0, 3);
  sort(6, 10);
  sort(2, 9);
  sort(10, 14);
  return 0;
}
```

At the end of the program, which of the following must be true?

i. `a[9] >= a[0]`
ii. `a[7] >= a[0]`
iii. `a[14] >= a[5]`

A. i and ii only
B. i and iii only
C. ii and iii only
D. i, ii and iii

21. Consider the following program:

Pascal Version

```
var
  n, i, x, y: longint;
  a: array[0..9] of longint;
begin
  for i := 0 to 9 do
    read(a[i]);
  x := 0;
  y := 9;
  while (x <> y) do
  begin
    if (a[x] < a[y]) then
      inc(x)
    else
      dec(y)
  end;
  write(x)
end.
```

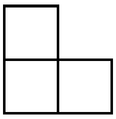
C Version

```
#include <stdio.h>
int n, i, x, y;
int a[10];
int main() {
  for (i = 0; i <= 9; i++)
    scanf("%d", &a[i]);
  x = 0;
  y = 9;
  while (x != y) {
    if (a[x] < a[y])
      x++;
    else
      y--;
  }
  printf("%d", x);
  return 0;
}
```

Which of the following input leads to the maximum output?

A. 1 0 5 7 8 0 3 4 6 2
B. 7 1 3 9 4 8 2 5 6 0
C. 5 4 9 0 3 2 7 8 9 1
D. 1 2 7 4 3 0 8 6 5 6

22. L-piece is a shape that can cover exactly three grid cells with four different rotations, here is an L-piece:



The grid is tiled completely if and only if each grid cell is covered by a L-piece.
Assume that we are tiling 16 L-pieces on 7×7 grid with a single cell removed initially, which of the following grid(s) are possible to be tiled completely? (You may treat $(1, 1)$ and $(7, 7)$ as the top-leftmost and bottom-rightmost cell respectively. (x, y) denotes the grid in row x and column y)

i. The grid with cell $(7, 7)$ removed
ii. The grid with cell $(3, 6)$ removed
iii. The grid with cell $(2, 5)$ removed

A. ii only
B. i and ii only
C. i and iii only
D. i, ii and iii

23. Consider the following system of inequalities:

$$\begin{cases} a \geq b - 1 \\ b \geq c + 2 \\ c \geq a - 3 \\ 0 \leq c \leq 3 \end{cases}$$

How many groups of integer solutions (a, b, c) are there to the above system of inequalities?

A. 8
B. 16
C. 24
D. 32

24. Skylake is the ...

A. codename for a version of CPU
B. codename for a version of operating system
C. name of a programming language
D. name of a server software

25. You have 200 units of food in the storage and 3 villagers initially. Your base can create a villager instantaneously by consuming 50 units of food from the storage. A villager produces 1 unit of food each second and put it into the storage in the end of the second.

What is the fastest time of having 500 units of food in the storage?

A. 69
B. 70
C. 72
D. 100

END OF SECTION A

Section B (20 marks)

The blanks are labeled from A to L. Please fill in the blanks on the answer sheet.

Except otherwise specified, two marks for each correct blank. No marks will be deducted for wrong answers.

Note:

(1) You must not use the `?:` operator in C.
(2) You must not use any library function unless the appropriate library(s) is/are included.
(3) You can write only one character in each box on the answer sheet.
(4) No answer with length greater than the designated number of boxes will be accepted.

1. Complete the following program so that it outputs `"place"`.

Pascal Version

```
var
  s: string;
  i: longint;
begin
  s := 'pace';
  for _____A_____ do
    s[i] := _____B_____;
  s[i] := 'l';
  setlength(s, 5);
  write(s)
end.
```

C Version

```
#include <stdio.h>
char s[256] = "pace";
int i;
int main() {
  for (_____A_____)
    s[i] = _____B_____;
  s[i] = 'l';
  s[5] = 0;
  printf("%s", s);
  return 0;
}
```

Answer: _______A_______ (1 mark)
Answer: _______B_______ (1 mark)

2. Consider the following program segment:

Pascal Version

```
function sq_cmp(a, b: longint): longint;
begin
  if (a * a > b * b) then
    sq_cmp := 1
  else
    sq_cmp := 0
end;
function equals(a, b: longint): longint;
begin
  equals := sq_cmp(_____C_____)
end;
function greater(a, b: longint): longint;
begin
  greater := sq_cmp(_____D_____)
end;
```

C Version

```
int sq_cmp(int a, int b) {
  if (a * a > b * b)
    return 1;
  else
    return 0;
}
int equals(int a, int b) {
  return sq_cmp(_____C_____);
}
int greater(int a, int b) {
  return sq_cmp(_____D_____);
}
```

Complete the above program segment, such that for any integers `a`, `b` between -100 and 100 (inclusive), `equals(a, b)` returns `1` if `a == b` (**Pascal**: `a = b`) and returns `0` otherwise, and `greater(a, b)` returns `1` if `a > b` and returns `0` otherwise.

You can only use `a`, `b`, numerical constants, comma `","`, spaces, brackets `"()"` and the following operators: `+`, `-`, `*`, `/` (Pascal: `div`), `%` (Pascal: `mod`).

Answer: _______C_______ (2 marks)
Answer: _______D_______ (2 marks)

3. Consider the following program:

Pascal Version

```
var
  n, ans: longint;
procedure f(a, b, n: longint);
begin
  if (n = 0) then
    ans := b - a
  else if (n mod 2 = 0) then
    f(a + 1, b, n div 2)
  else
    f(a, b + 1, n div 2)
end;
begin
  read(n);
  ans := 0;
  if ((n >= 0) and (n <= 2016)) then
    f(0, 0, n);
  write(ans)
end.
```

C Version

```
#include <stdio.h>
int n, ans;
void f(int a, int b, int n) {
  if (n == 0)
    ans = b - a;
  else if (n % 2 == 0)
    f(a + 1, b, n / 2);
  else
    f(a, b + 1, n / 2);
}

int main() {
  scanf("%d", &n);
  ans = 0;
  if ((n >= 0) && (n <= 2016))
    f(0, 0, n);
  printf("%d", ans);
  return 0;
}
```

If the output of this program is 3. What is the minimum possible input and maximum possible input? The minimum possible input is ____E____ and the maximum possible input is ____F____.

Answer: ______E______ (1 mark)
Answer: ______F______ (2 marks)

4. Suppose array `a` (**Pascal**: `a[1..n] of longint`, **C**: `int a[n]`) contains distinct integers sorted in ascending order, integer `n` stores the size of array `a` and integer `x` stores the integer to find. The program below tries to implement binary search to check whether integer `x` exists in array `a`. However, the program contains an error that can be fixed by changing exactly one line. Find the line and correct it. You may assume variables `l`, `r`, `mid` are declared as `int` (**Pascal**: `longint`).

Pascal Version

```
11 l := 1;
12 r := n;
13 while (l <= r) do
14 begin
15   mid := (l + r) div 2;
16   if (a[mid] < x) then
17     l := mid + 1
18   else
19     r := mid - 1
20 end;
21 if (a[mid] = x) then
22   write('Found')
23 else
24   write('Not found');
```

C Version

```
51 l = 0;
52 r = n - 1;
53 while (l <= r)
54 {
55   mid = (l + r) / 2;
56   if (a[mid] < x)
57     l = mid + 1;
58   else
59     r = mid - 1;
60 }
61 if (a[mid] == x)
62   printf("Found");
63 else
64   printf("Not found");
```

Line number: ____G____ (1 mark)
Correction: __________H__________ (2 marks)

5. The following program attempts to output all factors of positive integer input x.

Pascal Version

```
var
  i, k, t, x: longint;
begin
  read(x);
  k := x;
  writeln(1);
  if (x > 1) then
  begin
    for i := 2 to x do
    begin
      t := 1;
      while (k mod i = 0) do
      begin
        k := k div i;
        t := t * i;
        writeln(t)
      end
    end
  end
end.
```

C Version

```
#include <stdio.h>
int i, k, t, x;
int main() {
  scanf("%d", &x);
  k = x;
  printf("1\n");
  if (x > 1) {
    for (i = 2; i <= x; i++) {
      t = 1;
      while (k % i == 0) {
        k = k / i;
        t = t * i;
        printf("%d\n", t);
      }
    }
  }
  return 0;
}
```

I1. Write down a positive integer input x that is greater than 30, so that the program outputs all the factors of x correctly.

Answer: I1

I2. Write down a positive integer input x that is greater than 30, so that the program outputs all the factors of x incorrectly.

Answer: I2

2 marks are given only if both correct.

6. A "cross shape" is shown below.

```
 *
***
 *
```

Consider the following grid. There are 2 cross shapes center at $(2, 2)$ and $(4, 4)$ respectively.
("." denotes an empty grid cell, (x, y) denotes the grid in row x and column y)

```
.*...
***..
.*.*.
..***
...*.
```

We call 2 cross shapes touch each other if they share the same corner but no edge.
We call 2 cross shapes connected if they share the same edge.

Therefore, the above cross shapes touch each other.

Consider the following program segment, (a, b) stored the coordinate of the center of a cross shape, (x, y) stored the coordinate of center of another cross shape.

Pascal Version

```
read(a, b, x, y);
if (____J____) then
  write('They are connected');
if (____K____) then
  write('They touch each other');
```

C Version

```
scanf("%d%d%d%d", &a, &b, &x, &y);
if (____J____)
  printf("They are connected");
if (____K____)
  printf("They touch each other");
```

Complete the following blanks so that the program can correctly determine whether the cross shapes touch, connect or not.

The program should output `"They touch each other"` if and only if they touch each other.

The program should output `"They are connected"` if and only if they are connected.

It is guaranteed that two crosses do not overlap.

Note: You are allowed to use the `abs` function.

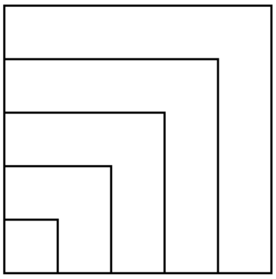
Answer: _______J_______ (2 marks)

Answer: _______K_______ (2 marks)

7. The "Logo" programming language can be used to draw shapes. Initially there is a "turtle" on the screen pointing upwards. Then you can issue commands to move it around so its path would be left on the screen. For example, consider the following program:

```
to sq :size
  if :size = 0 [ stop ]
  repeat 4 [ fd :size rt 90 ]
  sq :size - 50
end
sq 250
```

The result image would be:



Now draw, on the answer sheet, the result image of the following program. Scale (size) does not matter and minor inaccuracies are acceptable.

Hint: `repeat` means repeat, `fd` means forward, `lt` means left turn and `rt` means right turn.

```
to shape :size
  if :size = 30 [ stop ]
  repeat 3 [ fd :size / 3 lt 60 shape :size / 3 rt 60 fd :size * 2 / 3 rt 120 ]
end
shape 270
```

Answer: _______L_______ (2 marks)

END OF PAPER