Assume that all variables without declaration shown in the following program segments have already been declared properly. Integers (`integer`/`int`) and long integers (`longint`/`long`) in problem statements are 16-bit signed variables and 32-bit signed variables respectively. Assume all the programs are compiled properly without using any compiler flag (except the "-o" option in C).

Notations

For any real number *x*,

(1) $\lfloor x \rfloor$ denotes the largest integer not greater than *x*.

(2) $\lceil x \rceil$ denotes the smallest integer not less than *x*.

(3) $|x|$ denotes the absolute value of *x*.

For non-negative integer *n*, its factorial *n*! = 1 × 2 × 3 × … × (*n*-1) × *n*; and 0! = 1.

Truth tables

a	***b***	***a* XOR *b***	***a* XNOR *b***
F	F	F	T
F	T	T	F
T	F	T	F
T	T	F	T

Section A (25 marks)

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

1. You are given an NxN chessboard. Initially, all cells are white except one is blue. Now Ada wants to colour the cells with a blue pen so that no adjacent cells share the same colour. Further, it is known that the only existing blue cell does not share any edges with the top-leftmost cell. Which of the following method(s) always work(s)?

   (i) Colour top-leftmost cell, as well as every alternate cell on the right, if any. Then for each column, repeatedly colour the topmost cell X that has a white cell on top of X, if any.
   (ii) Pick a blue cell, colour all the white cells (if any) that share the same corner with the blue cell. Repeat this procedure until no more cells can be coloured.

   A. (i) only
   B. (ii) only
   C. (i) and (ii)
   D. None of the above

2. At least how many times of hand shaking are required among 100 people so that everyone can shake hands at least once?

   A. 5050
   B. 4950
   C. 100
   D. 50

3. Euclidean distance between 2 points (x1,y1) and (x2,y2) is defined as $\sqrt{(x1-x2)^2+(y1-y2)^2}$. Manhattan distance between 2 points (x1,y1) and (x2,y2) is equal to $|x1-x2|+|y1-y2|$

   Which of the following statement(s) about these two distances must be **TRUE**?

(i) For a triangle ABC, the sum of Euclidean distances between A and B, B and C is greater than that between A and C.
(ii) For a triangle ABC, the sum of Manhattan distances between A and B, B and C is greater than that between A and C.
(Assume that A, B and C do not form a straight line.)

A. (i) only
B. (ii) only
C. (i) and (ii)
D. None of the above

4. What is the output of the following program?

Pascal Version

```
var
i,j:integer;
    a:array[1..16] of boolean;
begin
    for i:=1 to 16 do a[i]:=false;
    for i:=1 to 16 do
    begin
        j:=i;
        while j<=16 do
        begin
           a[j]:=a[j] xor true;
           j:=j+i;
        end;
    end;
    writeln(a[8],' ',a[16]);
end.
```

A. TRUE TRUE
B. TRUE FALSE
C. FALSE TRUE
D. FALSE FALSE

C Version

```
#include <stdio.h>
int i,j,a[17];
int main(){
       for (i=1;i<=16;i++) a[i]=0;
       for (i=1;i<=16;i++){
          j=i;
          while (j<=16){
               a[j]^1;
               j=j+i;
          }
       }
       printf("%d %d\n",a[8],a[16]);
}
```

A. 1 1
B. 1 0
C. 0 1
D. 0 0

5. Consider the following two program segments:

(i)

Pascal Version

```
procedure f(x,y:integer);
var
      tmp:integer;
begin
      if x<y then
      begin
             tmp:=a[x];
             a[x]:=a[y];
             a[y]:=tmp;
             f(x+1,y-1);
      end;
end;
```

C Version

```
void f(int x,int y){
      int tmp;
      if (x<y){
             tmp=a[x];
             a[x]=a[y];
             a[y]=tmp;
             f(x+1,y-1);
      }
}
```

(ii)

Pascal Version

```
procedure f(x,y:integer);
var
      i,tmp:integer;
begin
      for i:=x to y do
      begin
            tmp:=a[i];
            a[i]:=a[y-i+x];
            a[y-i+x]:=tmp;
      end;
end;
```

C Version

```
void f(int x,int y){
      int i,tmp;
      for (i=x;i<=y;i++){
            tmp=a[i];
            a[i]:=a[y-i+x];
            a[y-i+x]:=tmp;
      }
}
```

Which of the above program segments can correctly reverse the order of `a[`*x*`]` to `a[`*y*`]` inclusively?
(Assume no stack overflow problem.)

A. (i) only
B. (ii) only
C. (i) and (ii)
D. None of the above

6. Ada and Bob are playing a game. There are *N* stones on the ground and they take turns to pick at least 1 stone and up to 10 stones each time. Ada picks stones first. The one picking the last stone wins.
Which of the following values of *N* would make it possible for Ada to win?
(Assume both Ada and Bob are clever and use the best strategies to strive for winning.)

(i) 30
(ii) 264
(iii) 1027

A. (i) and (ii) only
B. (i) and (iii) only
C. (ii) and (iii) only
D. All of the above

7. You are now given a data structure called "drawer" with the following operations:

1. Put(*x*): put an integer *x* into the "drawer".
2. Extract(): return the maximum and minimum numbers in the "drawer" and remove them from the drawers.

Which of the following(s) can always be done by using only the operations of "drawer"?

(i) Find the median of *N* integers.
(ii) Sort *N* integers in ascending order.

A. (i) only
B. (ii) only
C. (i) and (ii)
D. None of the above

8. How many pairs of prime numbers between 1 to 20 (inclusively) have their sum divisible by 4?

A. 10
B. 11
C. 12
D. 13

9. Let `Rand(2)` returns 0 or 1 with equal probablity.

Pascal Version

```
x := Rand(2);
y := Rand(2);
writeln(x-y);
```

C Version

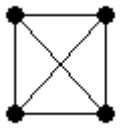
```
x = Rand(2);
y = Rand(2);
printf("%d\n",x-y);
```

Which of the following must be **TRUE**?

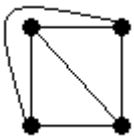
(i) The output is 0.
(ii) There are 3 possible outputs.
(iii) All possible outputs have equal possibility of appearing.

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

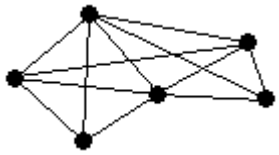
10. A planar graph is a graph which exists a drawing such that the edges do not intersect.
For example, the following graph is a planar graph.

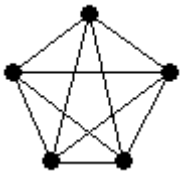


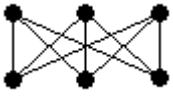
This is because it can be transformed to the graph below, without changing the connectivity of each pair of vertices:



Which of the following(s) is/are planar graph?

(i)


(ii)


(iii)


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

(Questions 11-12)

11. Alice had written the program segment below for sorting an array of 5000 integers in ascending order.

Pascal Version

```
for i:=0 to 4999 do
   for j :=0 to 4998 do
      if a[j]>a[j+1] then
         swap(a[j],a[j+1]);
```

C Version

```
for (i=0; i<5000; i++)
   for (j=0; j<4999; j++)
      if (a[j]>a[j+1])
         swap(a[j],a[j+1]);
```

Peddy argued that his program was not efficient enough. She changed the initial value of *j* in the inner for-loop.

Pascal Version

```
for i:=0 to 4999 do
   for j :=___?___to 4998 do
      if a[j]>a[j+1] then
         swap(a[j],a[j+1]);
```

C Version

```
for (i=0; i<5000; i++)
   for (j=___?___; j<4999; j++)
      if (a[j]>a[j+1])
         swap(a[j],a[j+1]);
```

Which of the following should be filled in the blanks such that the program is correct and consumes least time? You can assume the smallest element is not the last element.

A 1
B i
C i+1
D. i*2

12. Assume comparisons consume most of the running time of a program. If it takes 1 second for Billy's program to execute, what is the run time for the optimized correct program written by Peddy?

A. 0.1 seconds
B. 0.5 seconds
C. 1 second
D. 2 seconds

13. Let `A[0..50]` be an array of integers. We get `B[]` by reversing the order of elements in `A[]`. Which of the following is **NOT TRUE**?

A. `A[25]` is equal to `B[25]`.
B. If `A[10]<A[20]`, then `B[30]>B[40]`.
C. The sum of `A[15]` to `A[30]` is equal to the sum of `B[20]` to `B[35]`.
D. If `A[40]` is the maximum among A, then `B[40]` is the minimum among B.

14. Crystal picks two consecutive integers (e.g. 2 and 3) from 1 to 10 inclusive. David tries to guess one of her two consecutive numbers. After each guess, Crystal reports whether his guess is too small, too large or equal to one of the two consecutive numbers. What is the minimum number of guesses David needs such that he can be sure that he knows one of the number picked?

A. 2
B. 3
C. 4
D. 6

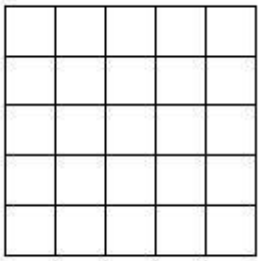
15. You have a strip of paper with some digits written on it. You want to cut this paper into several pieces, each with some digits on it and form an integer, such that the sum of the integers in all pieces is divisible by 10. How many ways can you do that? NO leading zero is allowed. You can choose not to cut at all.

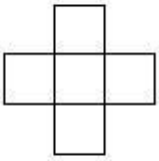
5490760

For example, 5490 + 760 = 6250 is divisible by 10

A. 5
B. 6
C. 7
D. 8

16. How many '+' piece do you need to cover every square in a 5x5 board? (The '+' pieces can be overlapped or out of boundary)





'+' piece

A. 6
B. 7
C. 8
D. 9

17. There are 10 cards labeled 0,1,2,3...,9. Now, one card is secretly taken. Which of the following is sufficient to show what the hidden card is?

(i) The unit digit of the sum of the remaining cards
(ii) The unit digit of the product of the remaining cards
(iii) The result of bitwise **OR** of the remaining cards
(iv) The result of bitwise **XOR** of the remaining cards

A. (i), (iii) and (iv) only
B. (i), (ii) and (iii) only
C. (ii) and (iii) only
D. (i) and (iv) only

18. Alice claims that for all positive integer x not less than y, she can pay \$$x$ exactly using only \$3 coins and \$7 coins. What is the minimum value of y?

A. 12
B. 15
C. 21
D. 24

19. In a football tournament, each team plays a game against another team exactly once. Each game has one winner (no ties). A team is a champion if no other team wins more game than it.

There are 5 teams participating this year and Joe's team is a champion. What is the minimum possible of winnings of Joe's team?

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

20. Suppose there are *n* islands in the Bob's Kingdom. Each bridge connects two islands. What is the minimum number of bridges required such that for all pairs of island, it is possible to travel from one to another using at most 2 bridges? You can assume all bridges are bidirectional.

A. *n*-1
B. *n*
C. *n*+1
D. 2*n*

21. How many decimal numbers in the range 1,2,...,100 have an even number of 1's when written in binary form?

A. 49
B. 50
C. 51
D. 52

22. Which of the following statement is **NOT** always **TRUE**?

A. If an integer x can be stored using a 32-bit signed integer, then x^2 can be stored using a 64-bit signed integer.
B. If a positive integer x can be stored using a 64-bit signed integer, then $\sqrt{x}$ can be stored using a 32-bit signed integer.
C. If a positive integer x can be stored using a 32-bit signed integer, then 2x can be stored using a 32-bit unsigned integer.
D. If a posiive integer x can be stored using a 32-bit signed integer, then $2x^2$ can be stored using a 64-bit unsigned integer.

23. Consider the following 4 statements about unknown integers *a*, *b* and *c*.

(i). $a > b$
(ii). $b > c$
(iii). $a > c$
(iv). $c \geq a$

Which of the following statement is **NOT** always **TRUE**?

A. At least one of statements (i), (ii) and (iii) is true
B. At least one of statements (i), (ii) and (iv) is true
C. At least one of statements (i), (ii) and (iv) is false
D. At least one of statements (i), (iii) and (iv) is false

24. Consider the following program segment. How many lines having exactly 5 "#" are there in the output?

Pascal Version

```
for i:=1 to 10 do
begin
    writeln('#');
    for j:=i+1 to 10 do
    begin
            writeln('#'); write('#');
            for k:=j+1 to 10 do
                write('#');
    end;
end;
```

C version

```
for (i=1;i<=10;++i){
   printf("#\n");
   for (j=i+1;j<=10;++j){
      printf("#\n#");
      for (k=j+1;k<=10;++k)
         printf("#");
   }
}
```

A. 4
B. 5
C. 6
D. 7

25. How many non-empty consecutive subsequences of the following sequence {3,10,8,9,4,6,7,5,1,2} have sum divisible by 3 ?
For example, {3,10,8,9} and {7,5}.
A. 16
B. 17
C. 18
D. 19

END OF SECTION A

Section B (20 marks)
The blanks are labeled from A to J. Please fill in the blanks on the answer sheet.
Two marks for each correct answer. 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 on each box on the answer sheet.
(4) No answer with length greater than the designated number of boxes will be accepted.

1. You are going to modify a string containing only capital letters using the HKOI text editor. Originally, the string is on the editor and a cursor is highlighting the first character of the string.
There are 4 kinds of command provided by the editor.

command	meaning
m?	Modify the character highlighted by the cursor to "?"
l	Shift the cursor to the position on the left of it. If the cursor is at the leftmost position, it will stay at the leftmost position
r	Shift the cursor to the position on the right of it. If the cursor is at the rightmost position, it will stay at the rightmost position
(command)N	Repeat the command by N times. For example "(rr)3" means shifting the cursor to the right by 6 times.

In the example and question below, '_' indicates the position of the cursor.

If the screen is initially shown as:
ABCDE

After executing the sequence of command "`mF(r)7mZ`", the screen will be shown as:

FBCDZ

Write the sequence of command for modifying the following strings respectively.

Initial string:
TCTCTCTCTCTC

Target string:
AGAGAGAGAGAG

Command: **A**
Initial string:
XXXXXXYXY

Target string:
YYYYYYXYX

Command: **B**

2. Alice has written a binary search which was going to print the index of the target in array `a[]`={0,1,2,3,4,5,6,7,8,9,10,11} (`a[0]..a[11]`). However, Alice found that the program did not print the correct index.

   Rewrite line with comment "#" in order to correct the program. Assume the initial value of target is between 1 and 10 inclusively.

Pascal version

```
left:=1;
right:=10;
while left<right do
begin
   mid:=(left+right) div 2; // #
   if a[mid] > target then
 right := mid-1
   else
left := mid;
end;
writeln(left);
```

C version

```
left = 1;
right = 10;
while (left < right){
    mid = ( left + right ) / 2; /* # */
    if (a[mid] > target)
       right = mid - 1;
    else
       left = mid;
}
printf("%d\n", left);
```

#: **C**

3. A regular ruler may have integer marks from 0 to the length of the ruler. However, sometimes a ruler has some missing marks, but you can still use it to measure any integer length from 1 up to the ruler's full length. This kind of ruler is called 'sparse ruler'. An example of a sparse ruler of full length 6 is given below:

   You can measure any integer length from 1 to 6. For example, you can get length 2 by measuring between 4 and 6, length 5 by measuring between 1 and 6 etc.

   Create a sparse ruler of length 9 with at most 5 marks.
   D

   Create a sparse ruler of length 13 with at most 6 marks
   E

4. The number of inversions of an array `A[`*n*`]` (`A[0]..A[n-1]`) is defined as the number of pairs (*i*,*j*), where 0≤ *i* < *j* <*n*, such that `A[`*i*`]>A[`*j*`]`.

   Alice has written the following program segment to count the number of inversion pairs of array `A`, where each element is an integer between 1 and 1001. The program will output the number of inversions in the array. Can you help Alice to complete the program?

Pascal Version

```
sum:=0;
for i:=0 to n-1 do
  vis[i]:=0;
for i:=1001 downto 1 do
begin
    for j:=n-1 downto 0 do
    begin
       if (    F1    ) then
       begin
          for k:=0 to j-1 do
             sum:=sum+vis[k];
          F2 ;
       end;
    end;
end;
writeln(sum);
```

C Version

```
sum=0;
for (i=0;i<n;i++)
   vis[i]=0;
for (i=1001;i>=1;i--){
   for (j=n-1;j>=0;j--){
      if (    F1    ){
         for (k=0;k<j;k++)
            sum=sum+vis[k];
         F2 ;
      }
   }
}
printf("%d\n",sum);
```

5. Alice has written the following program segment below to count the number of prime factors for a given integer *n*, where *n* is an integer between 2 and 1001 inclusively. For example, $12=2^2x3$, hence 12 has 2 prime factors. However, Bob claims that her program is incorrect. Can you provide a valid input such that his program gives an incorrect output?

Input: *n*
Output: Number of prime factors of *n*

Pascal Version

```
ans:=1;
readln(n);
i:=2;
while i*i<=n do
begin
   if n mod i=0 then
   begin
      ans:=ans+1;
      while (n mod i=0) do
          n:=n div i;
   end;
   i:=i+1;
end;
writeln(ans);
```

C Version

```
ans=1;
scanf("%d",&n);
for (int i=2;i*i<=n;i++){
    if (n%i==0){
        ans++;
        while (n%i==0) n/=i;
    }
}
printf("%d\n",ans);
```

n = **G**

6. Suppose there is an integer array `A[]`.Alice wrote a program to find the largest and the second largest elements (`A[0]` and `A[1]` after sorting `A` in non-ascending order) of `A[]` without sorting the array `A[]`. However, Bob claims that her program is incorrect. Can you state an array of `A[]` containing 5 elements such that his program will give incorrect output?

Output: The largest and second largest element
You may assume that all variables have been declared properly.

Pascal Version

```
max1:=-1;
max2:=-1;
for i := 0 to 4 do
begin
   if (A[i]>max2) and (A[i]<max1)
   then max2:=A[i]
   else if A[i]>max1 then begin
      max2:=max1;
      max1:=A[i];
   end;
end;
writeln(max1,' ',max2);
```

C Version

```
max1=-1;
max2=-1;
for (i=0;i<5;i++){
    if (A[i]>max2&&A[i]<max1)
max2=A[i];
    else if (A[i]>max1){
        max2=max1;
        max1=A[i];
    }
}
printf("%d %d\n",max1,max2);
```

The array required: _ _ _ _ _ (**H**)

7. Consider the following program segment. Assume the variable `a` is declared as a 32 bit integer, and the user's input for `a` is always positive.

Pascal Version

```
readln(a);
writeln(a);
if (a>=100) then
begin
    if (   I1   ) then
        write(a div 100)
    else
        write(   I2   );
end;
writeln(a mod 100);
```

C Version

```
scanf("%d",&a);
printf("%d\n",a);
if (a>=100){
   if(   I1   )
      printf("%d", a / 100);
   else
      printf("%d",   I2   );
}
printf("%d\n", a % 100);
```

Complete the program segment above such that the two lines of output are the same.

8. You are given 3 integer arrays `a[5]`, `b[5]`, `c[5]` (`a[0..4]`, `b[0..4]`, `c[0..4]` in Pascal):

`a[5]` = {1,0,4,2,3}
`b[5]` = {0,4,2,3,1}
`c[5]` = {4,2,3,1,0}

What is the output of the following program segment?

Pascal Version

```
ans:=0;
for i:=0 to 4 do
   for j:=0 to 4 do
       for k:=0 to 4 do

ans:=ans+a[b[i]]+b[c[j]]+c[a[k]];
writeln(ans);
```

C Version

```
ans=0;
for (i=0; i<5; i++)
    for (j=0; j<5; j++)
        for (k=0; k<5; k++)

ans+=a[b[i]]+b[c[j]]+c[a[k]];
printf("%d\n",ans)
```

J: _____

END OF PAPER