2023-DSE
ICT
PAPER 2D

HONG KONG EXAMINATIONS AND ASSESSMENT AUTHORITY

HONG KONG DIPLOMA OF SECONDARY EDUCATION EXAMINATION 2023

Please stick the barcode label here.

Candidate Number									

INFORMATION AND COMMUNICATION TECHNOLOGY

PAPER 2D

Software Development

Question-Answer Book

11:15 am – 12:45 pm (1 hour 30 minutes)
This paper must be answered in English

INSTRUCTIONS

(1) After the announcement of the start of the examination, you should first write your Candidate Number in the space provided on Page 1 and stick barcode labels in the spaces provided on Pages 1, 3, 5 and 7.

(2) Answer **THREE** out of four questions. Write your answers in the spaces provided in this Question-Answer book. Do not write in the margins. Answers written in the margins will not be marked.

(3) Supplementary answer sheets will be supplied on request. Write your candidate number, mark the question number box and stick a barcode label on each sheet, and fasten them with string **INSIDE** this book.

(4) No extra time will be given to candidates for sticking on the barcode labels or filling in the question number boxes after the 'Time is up' announcement.

A200E02D

Answer THREE questions only.

1. Eva plans to write a burger cooking game. She uses a queue A to store the ingredients. A is implemented by an array L. She designs subprograms with global variables, as shown below:

Global variable	Description
L	An array for storing the elements in A with indices from 1 to 5
C	A variable for storing the number of elements in A

Subprogram	Description
enq(A, K)	Inserts an element K in A if A is not full
deq(A)	Removes and returns an element in A if A is not empty

(a) Suppose that C = 0. After sequentially executing enq(A, Fish) and enq(A, Onion),

C = 2 and

i	1	2	3	4	5
L[i]	Fish	Onion			

enq(A, K) outputs a message 'Full!' when A is full. Complete the pseudocode for enq below.

```
enq(A, K)
    if □ then
        output 'Full!'
    else
        L[ □ ] ← K
        C ← C + 1
```

(2 marks)

(b) (i) Suppose that C = 3 and

i	1	2	3	4	5
L[i]	Onion	Tomato	Bacon		

Then, after executing deq(A),

C = 2 and

i	1	2	3	4	5
L[i]	Tomato	Bacon			

Fill in the content of L after further sequentially executing deq(A), enq(A, Cheese), enq(A, Beef) and deq(A).

i	1	2	3	4	5
L[i]					

(2 marks)

Please stick the barcode label here.

(ii) Complete the pseudocode for deq below.

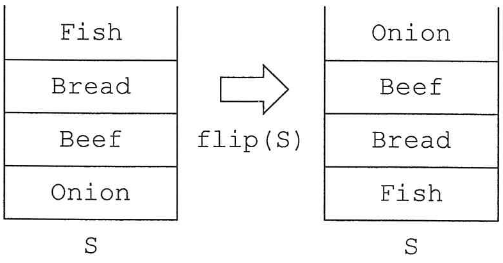
```
deq(A)
    if C = 0 then
        output 'Empty! '
    else
        j ← 1
        tmp ← L[1]
        while j < C do

            j ← j + 1


        return tmp
```

(2 marks)

In the game, Eva uses a stack S to represent a burger. She writes cook(A) using the following subprograms where queue A stores all the ingredients of the burger.

Subprogram	Description
push(S, K)	Inserts an element K in S as its top element
flip(S)	Reverses the order of all elements in S

In the following example, after executing flip(S), the order of all elements in S is reversed.



Answers written in the margins will not be marked.
Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Suppose that A stores all the ingredients of a specific burger. Eva writes cook(A) to cook the ingredients in A. Suppose that S is initially empty.

```
cook(A)
   while A is not empty do
      push(S, deq(A))
      flip(S)
```

(c) (i) Suppose that the content of L is:

i	1	2	3	4	5
L[i]	Beef	Onion	Bread	Bread	

Fill in the content of S below, after executing cook(A).

S

(2 marks)

(ii) Suppose that the content of S after executing cook(A) is:

Bread
Tomato
Lettuce
Beef
Bread

S

Fill in the initial content of L below.

i	1	2	3	4	5
L[i]					

(2 marks)

Another queue B is implemented by an array R.

(d) (i) Suppose that the initial contents of S and R are:

Onion
Mushroom
Tomato
Fish

S

i	1	2	3	4	5
R[i]					

pop(S) is a subprogram that removes and returns the top element of S.
Fill in the contents of S and R after sequentially executing enq(B, pop(S)), enq(B, pop(S)) and enq(B, pop(S)).

S

i	1	2	3	4	5
R[i]					

(2 marks)

(ii) Complete the pseudocode for flip(S) below.

```
flip(S)
    while  S is not empty  do
        enq(                              )
    while                              do
        push(                             )
```

(3 marks)

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

2. There are 30 booths in a food fair. Mary designs a floor plan with 6 rows and 5 columns. She uses an array `F[i,j]` to store the booth number of the booth in row `i`, column `j`.

Row `i`

6					
5					
4					
3					
2					
1					
	1	2	3	4	5

Column `j`

Mary writes a subprogram `assign1` that assigns booth numbers to `F`, as shown below:

```
assign1
    for i from 1 to 6 do
            for j from 1 to 5 do
                    F[i, j] ← 5*(i-1) + j
```

(a) (i) Write down the booth number stored in `F[5, 4]`. ______________________

(1 mark)

(ii) Write down the indices of the element in `F` that stores booth number `17`.

F[_______,_______]

(1 mark)

Mary modifies `assign1` to `assign2` to reassign the booth numbers in the floor plan, as shown below:

Row `i`

6	30	29	28	27	26
5	21	22	23	24	25
4	20	19	18	17	16
3	11	12	13	14	15
2	10	9	8	7	6
1	1	2	3	4	5
	1	2	3	4	5

Column `j`

Please stick the barcode label here.

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

(b) (i) Complete the pseudocode for `assign2` below.

```
assign2
    for i from 1 to 6 do
        for j from 1 to 5 do

            if                                          then

                F[i, j] ← 5*(i-1)+j
            else
                F[i, j] ←
```

(3 marks)

(ii) Mary writes a subprogram `findRow(num)` that returns the row number of the booth with the booth number `num`. For example, `findRow(30)` returns `6`. Write the pseudocode for `findRow(num)`.

```
findRow(num)
```

(2 marks)

There are 10 vegetarian booths in the food fair and their booth numbers are stored in an array `P` in ascending order, as shown below:

i	1	2	3	4	5	...	10
P[i]	4	7	13	16	18	...	28

Mary writes a subprogram `isVeg(num)` that returns `TRUE` if `num` is in `P`, `FALSE` otherwise.

(iii) Complete the pseudocode for `isVeg(num)` below.

```
isVeg(num)
      left ← 1
      right ← 10

      while                          do

            m ← integral part of (left + right)/2
            if P[m] = num then
                  return TRUE
            else if P[m] < num then


            else


      return FALSE
```

(3 marks)

Answers written in the margins will not be marked.

(c) Mary considers using a low level language and a linker to develop the subprograms.

(i) Give an advantage of using low level language for software development.

____

____

(1 mark)

(ii) Describe the major function of a linker in the execution of a program.

____

____

____

(2 marks)

(d) Mary compares phased conversion and direct cutover conversion for system conversion. Give an advantage of each kind of conversion.

Phased conversion: ________________________________

____

____

Direct cutover conversion: ____________________________

____

____

(2 marks)

Answers written in the margins will not be marked.

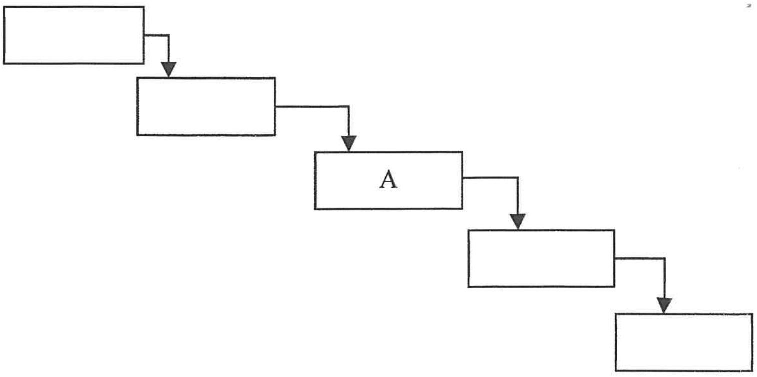
Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

3. Peter works on a project to redevelop a computer system using the Waterfall model with five phases.

(A) System implementation
(B) System conversion
(C) System analysis
(D) System maintenance
(E) System design

(a) (i) Fill in the phases in the Waterfall model below.



(2 marks)

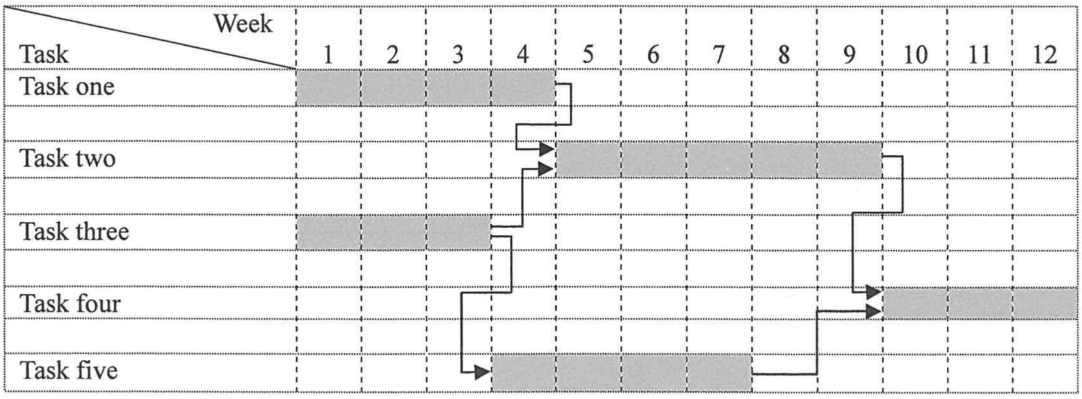
(ii) What does Peter do in the system maintenance phase?

____

____

(1 mark)

The Gantt chart of the project is shown below.



(b) (i) What is the critical path of the project? ______________________

(1 mark)

(ii) Suppose that Task three finally needs 4 weeks to complete. What is the minimum number of weeks required to complete the project?

____

(2 marks)

Peter develops a program with the following subprograms to sort integers in an array A in ascending order. All integers in A are distinct.

Subprogram	Description
findmin(s, e)	Return the index of the minimum value in A[s], A[s+1],..., A[e] where s < e.
findmax(s, e)	Return the index of the maximum value in A[s], A[s+1],..., A[e] where s < e.
swap(x, y)	Swap the values of A[x] and A[y].

Suppose that the initial content of A contains 7 integers:

i	1	2	3	4	5	6	7
A[i]	19	28	11	43	9	16	23

findmin(1, 4) returns 3 and findmax(3, 5) returns 4.

(c) (i) What is the return value of findmax(5, 7)? ______________________

(1 mark)

(ii) Complete the pseudocode for findmax(s, e) below.

```
findmax(s, e)
      tmp ← s
      for k from s+1 to e do

          if A[k] [            ] then

                tmp ← [          ]

      return tmp
```

(3 marks)

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Peter tries to write the pseudocode for a subprogram sub1 for sorting.

```
sub1
    for j from 1 to 3 do
            a ← findmin(j, 8 – j)
            b ← findmax(j, 8 – j)
            swap(j, a)
            swap(8 – j, b)
```

(d) (i) Suppose that the initial content of A is:

i	1	2	3	4	5	6	7
A[i]	19	28	11	43	9	16	23

Fill in the content of A after the first pass and second pass of the loop in sub1.

After the first pass

i	1	2	3	4	5	6	7
A[i]							

After the second pass

i	1	2	3	4	5	6	7
A[i]							

(2 marks)

(ii) With the value in A[6] after the second pass, Peter understands that an incorrect sorting result is obtained. To correct the algorithm, he modifies sub1 to sub2. Complete the pseudocode for sub2 below.

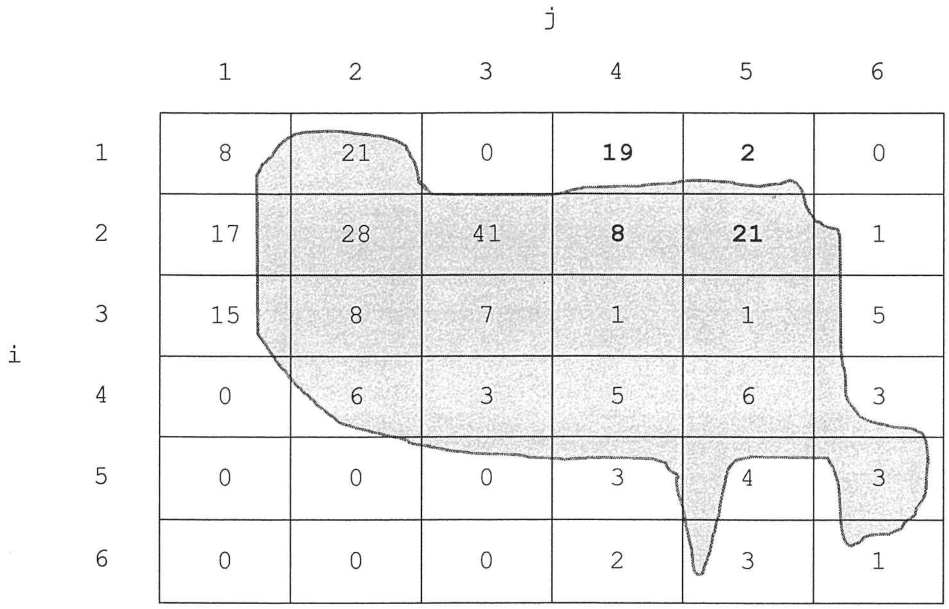
```
sub2
        for j from 1 to 3 do
                a ← findmin(j, 8 – j)
                b ← findmax(j, 8 – j)
                swap(j, a)
                if □ then
                     swap(□, □)
                else
                     swap(8 – j, b)
```

(3 marks)

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

4. A map is represented by N×N cells. John works in a food delivery company. He uses an array MAP to store the number of restaurants in each cell. The following example shows MAP with N=6.



j

i	1	2	3	4	5	6
1	8	21	0	**19**	**2**	0
2	17	28	41	**8**	**21**	1
3	15	8	7	1	1	5
4	0	6	3	5	6	3
5	0	0	0	3	4	3
6	0	0	0	2	3	1

MAP

John writes a subprogram `sum(i, j, K)` that returns the sum of the numbers in K×K cells where `MAP[i, j]` is the top left cell. For example, `sum(1, 4, 2)` returns `19+2+8+21 = 50` where `MAP[1, 4]` is the top left cell.

(a) (i) What is the return value of `sum(1, 4, 3)`?

____

(1 mark)

(ii) What are p and q so that `sum(p, q, 2)` returns the maximum value among all the 2×2 cells?

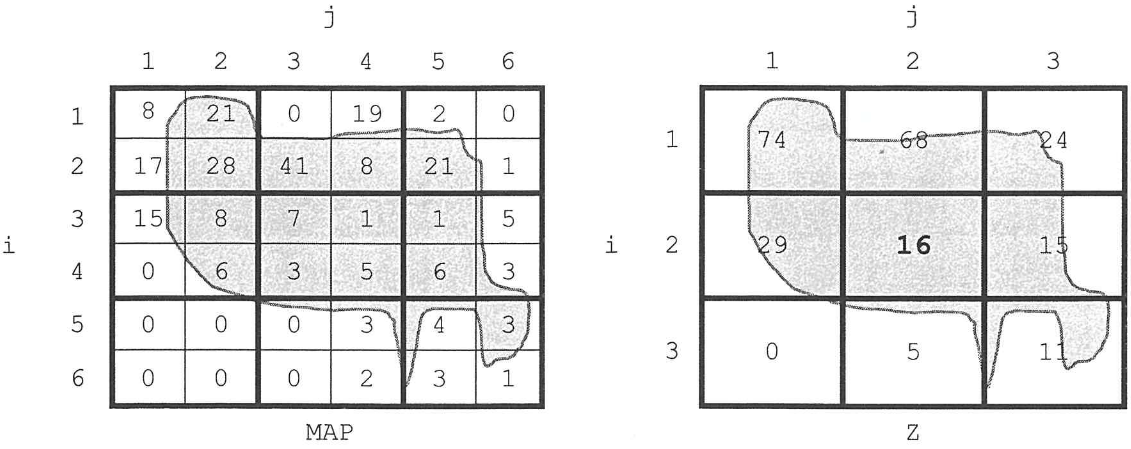
p = ____________ q = ____________

(1 mark)

Answers written in the margins will not be marked.

(b) John writes a subprogram `Zoomout(K)` that merges K×K cells into 1 cell by calculating the sum of the numbers in the K×K cells for `MAP` and storing the calculated sums in an array `Z`.

(i) For `N = 6` and `K = 2`,



MAP

i \ j	1	2	3	4	5	6
1	8	21	0	19	2	0
2	17	28	41	8	21	1
3	15	8	7	1	1	5
4	0	6	3	5	6	3
5	0	0	0	3	4	3
6	0	0	0	2	3	1

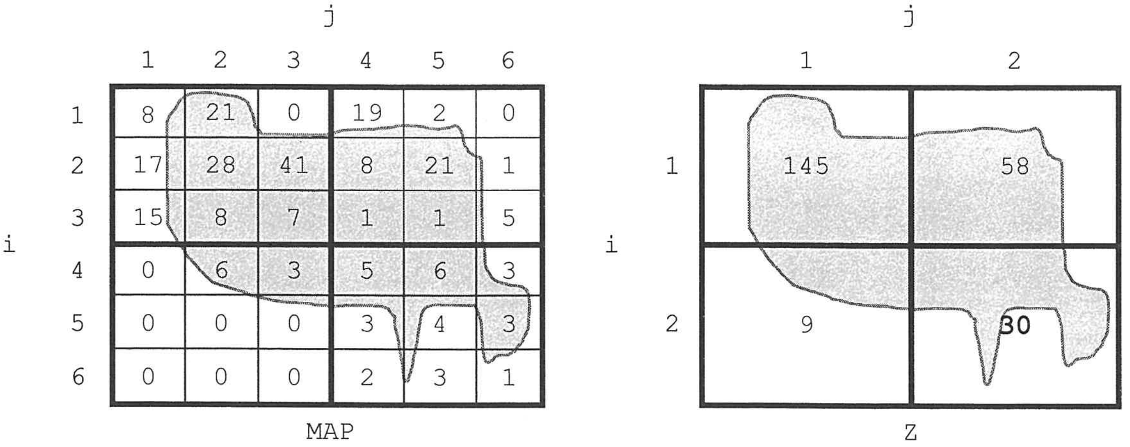
Z

i \ j	1	2	3
1	74	68	24
2	29	**16**	15
3	0	5	11

The value of `Z[2, 2]` is the return value of `sum(r, 3, 2)`. What is `r`? ______________

(1 mark)

(ii) For `N = 6` and `K = 3`,



MAP

i \ j	1	2	3	4	5	6
1	8	21	0	19	2	0
2	17	28	41	8	21	1
3	15	8	7	1	1	5
4	0	6	3	5	6	3
5	0	0	0	3	4	3
6	0	0	0	2	3	1

Z

i \ j	1	2
1	145	58
2	9	**30**

The value of `Z[2, 2]` is the return value of `sum(4, s, 3)`. What is `s`? ______________

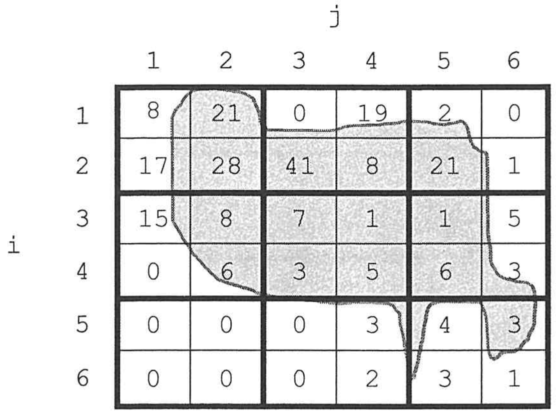
(1 mark)

(iii) Suppose that `N` is divisible by `K`. Complete the pseudocode for `Zoomout(K)`.

```
Zoomout(K)
  for i from 1 to N/K do
      for j from 1 to N/K do
         Z[i, j] ← sum(          ,          , K)
```

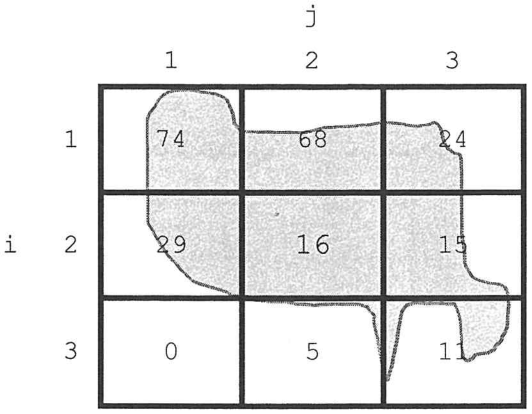
(2 marks)

(c) Below are two subprograms and MAP with N = 6 and K = 2.



i \ j	1	2	3	4	5	6
1	8	21	0	19	2	0
2	17	28	41	8	21	1
3	15	8	7	1	1	5
4	0	6	3	5	6	3
5	0	0	0	3	4	3
6	0	0	0	2	3	1

MAP



i \ j	1	2	3
1	74	68	24
2	29	16	15
3	0	5	11

Z

```
func1(i, j)
  tmp ← 0
  for p from i to i+1 do
           for q from j to j+1 do
                   if MAP[p, q] > tmp then
                           tmp ← MAP[p, q]
  return tmp


func2
  tmp ← 0
  for i from 1 to 3 do
      for j from 1 to 3 do
        if Z[i, j] >= tmp then
             if func1(i*2-1, j*2-1) > tmp then
                  tmp ← func1(i*2-1, j*2-1)
  return tmp
```

(i) What is the return value of func1(3, 1)? ____________________

(1 mark)

(ii) What is the return value of func2? ____________________

(1 mark)

(iii) What is the purpose of func2?

____

____

(1 mark)

(iv) In func2, what will happen if the fourth statement 'if Z[i, j] >= tmp then' is removed?

____

____

(1 mark)

(d) John insists on using a compiler instead of an interpreter to write programs. Give a reason to support his decision.

(1 mark)

(e) Some programming languages provide libraries to develop programs. Give **two** advantages of using libraries.

(2 marks)

(f) John uses Java, which is an object-oriented language, to develop programs. Give **two** advantages of object-oriented languages over procedural languages.

(2 marks)

END OF PAPER

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Answers written in the margins will not be marked.

Do not write on this page.

Answers written on this page will not be marked.