

Family Name: _____

Other Names: _____

Student ID: _____

Question 2.

Answer the following questions in the space provided.

(20 x 2 marks each)

- (1) Draw a set of gates representing the same logical structure as the following C Boolean condition: $((x \ || \ !y) \ \&\& \ z)$

- (2) Fill in the following table to show how the condition given in the above question is evaluated when x equals 1.

x	y	z	$((x \ \ !y) \ \&\& \ z)$
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Sample Exam Questions from Past Papers: CSE1301 Computer Programming

- (3) Briefly describe the difference between main memory and secondary storage.

- (4) Is the process of baking a cake an algorithm? Briefly justify your answer.

- (5) Briefly describe the effect of putting the `const` qualifier at the start of a declaration of a variable.

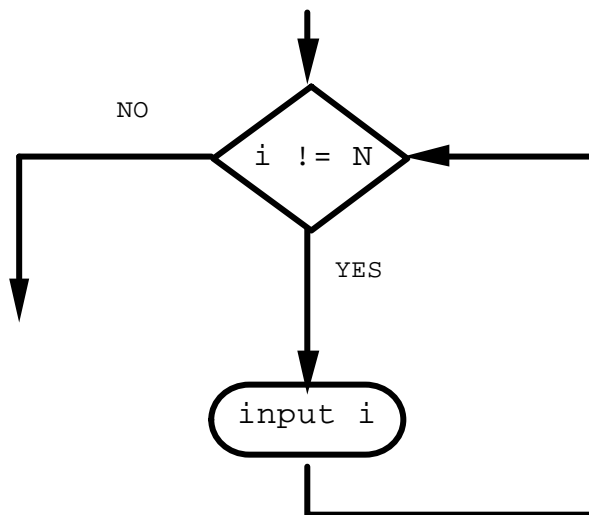
- (6) Write a statement which prints the least significant bit of the binary representation of the value of the integer variable `dilbert`.

- (7) Suppose `n` is a positive integer. Write a statement which sets the integer `m` to be the number of integers which are odd, positive, and less than or equal to `n`.

- (8) What is the order of evaluation of the operations `+`, `==`, `!`, `*` in the expression `red + green == ! blue * yellow`

- (9) Suppose the string called `familyName` contains a person's family name. Write a C expression which is true precisely when the person is listed in the first volume of the Melbourne phone book (i.e., the first letter of their family name is in the range A—K).
-

- (10) Write down the C code corresponding to the algorithm given by this flow diagram. (Assume that `i` and `N` have been declared as integers.)



Sample Exam Questions from Past Papers: CSE1301 Computer Programming

Consider the following C code fragment [for (11) and (12)]

```
int x;  
int y;  
  
scanf("%d%d", &x, &y);  
if (y > 2)  
{  
    printf("(%d,%d)", x, y);  
    x--;  
}  
printf(":%d", x+y);
```

(11) Draw a flow diagram for this code fragment

(12) Write down two sets of valid data (values for x and y) for testing this code, and the output for each.

x value	y value	Output

- (14) Describe the purpose of each of the parameters of the `fopen()` function.

- (15) Given the declarations:

```
char* str1 = "0.7";
```

```
char* str2 = "0.70";
```

what will be the type and value of the C expression:

```
( str1[0]!=0 && strcmp(str1,str2) )
```

Type: _____

Value: _____

- (16) Write a C prototype for a function `sumOfCharacters()` which, when given a string `name`, returns the sum of the ASCII values of the characters in `name`.

- (17) Define a struct to contain the number of students in a class, together with the average mark of the class.

(18) Represent the number -53 in 8-bit two's complement form.

(19) Suppose that floating point numbers are represented in base 10 with precision of 3 digits and exponents in the range $-20 \leq \text{exponent} \leq 21$. Numbers of greater precision are rounded to fit.

Consider the real numbers

$$x = 3.98524 \times 10^{-7}$$

$$y = 4.18130 \times 10^{-5}$$

Let x^* and y^* stand for the floating point representations of x and y respectively.

Give the floating point representation of $x^* + y^*$.

mantissa = _____

exponent = _____

(20) Give (a) one type of situation in which linear search succeeds but binary search does not, and (b) one type of situation in which you would prefer binary search to linear search.

(a) _____

(b) _____

Question 3 (Algorithms)

(10 marks)

Write an algorithm that describes how to wash up a pile of dirty dishes.

Your algorithm should include modules, selection and iteration.

(If you wish, you can consider such things as dish washers, double sinks, drying racks etc. However, the main thing is to use modules, selection and iteration properly and sensibly for the task.)

Question 4

(10 marks)

Write a C function which takes an integer parameter n (greater than 1) and returns the largest factorial which is less than n . Your function must not call any other functions.

Recall that the factorials are

$0! = 1! = 1$, $2! = 2$, $3! = 6$, etc, with

$$n! = n \times (n - 1) \times \dots \times 3 \times 2 \times 1$$

For example:

Given the parameter value **30**, your function should return **24** (4!).

Given the parameter value **24**, your function should return **6** (3!).

Given the parameter value **121**, your function should return **120** (5!).

Question 5.

(20 marks total)

Suppose an array of size N is being used to store sequentially a list of integers sorted in descending order. At some stage during the program execution, the array contains:

13	9	6	1	-3		
0	1	2	3	4	5	6

(a) Design an algorithm, using a flow diagram, for a module to delete an integer from the list, so that there are no gaps in the list and the list is kept in sorted order.

- Assume that the module is given the following parameters: the array, the index of the last item in the list (i.e. 4 for the example above), and the value of the integer to be deleted.
- The module should return the index of the integer's position in the list before it was deleted; if the integer is not found, the module should return -1.
- If the integer to be deleted is not in the list, the algorithm should print an error message, "Not in list".
- If the list is empty, the index passed to the module is -1.

Represent your algorithm using a flow diagram. (Note: pseudo-code for the algorithm is NOT required.) Your algorithm should be general enough to handle any valid input, but does not have to handle the case where the index value passed to the module is greater than or equal to N. (10 marks)

Question 5 (cont.)

(b) Write down a test data set which will fully test your algorithm for part (a). For each test, in the spaces provided, specify:

- the current status of the array,
- the index of the last item in the list
- the value, val, to be deleted from the list
- the integer value the module will return

(Add more test cases if your algorithm requires them.)

(8 marks)

<p>array</p> <table border="1" style="width: 100%; height: 30px; border-collapse: collapse;"> <tr> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> </tr> </table> <p style="text-align: center; font-size: small;">0 1 2 3 4 5 6</p>									<p>last index</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>val</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>returned value</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>
<p>array</p> <table border="1" style="width: 100%; height: 30px; border-collapse: collapse;"> <tr> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> </tr> </table> <p style="text-align: center; font-size: small;">0 1 2 3 4 5 6</p>									<p>last index</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>val</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>returned value</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>
<p>array</p> <table border="1" style="width: 100%; height: 30px; border-collapse: collapse;"> <tr> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> </tr> </table> <p style="text-align: center; font-size: small;">0 1 2 3 4 5 6</p>									<p>last index</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>val</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>returned value</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>
<p>array</p> <table border="1" style="width: 100%; height: 30px; border-collapse: collapse;"> <tr> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> <td style="width: 14.28%;"></td> </tr> </table> <p style="text-align: center; font-size: small;">0 1 2 3 4 5 6</p>									<p>last index</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>val</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>		<p>returned value</p> <div style="border: 1px solid black; width: 40px; height: 30px; margin: 0 auto;"></div>

Question 6

(15 marks)

There are many types of football games played around the world and in most countries football supporters collect *football cards*. Football cards typically show a picture of a player, plus information about the player such as name, height, weight, club, playing position, number of games played and so on.

Suppose you have a collection of 1000 football cards, and for each of these you want to store the player's name together with a number giving the card's monetary value (including cents).

(a) Define an appropriate data type in C to store this information (i.e. the player's name and the card's monetary value) for a single card. (3 marks)

(b) Declare a variable that is capable of storing all the information (i.e. name and value for every card) for your entire collection. (2 marks)

(c) Write a function which takes information on your collection, stored in such a variable, and prints out

- the average value of cards in your collection, and
- the name which comes first, in alphabetical order, of all the names of cards in your collection.

(10 marks)

Question 7

(20 marks)

A text file `customer.dat` contains a series of characters representing one person's address. You want to obtain their postcode (to help in compiling statistics on customers). Assume that their postcode is the *last* four-digit number in the file, and that it has white space on either side.

Write a program which reads in the address and prints out the postcode.

For example, if the address is

```
Mr Basil Fawlty
Year 2000 Task Force
Suite 26a, Fawlty Towers
4567 Wellington Road
Clayton, VIC 3168
Australia
```

then your program should print out 3168 .

Question 8

(10 marks total)

You have two arrays `alpha` and `beta`, each containing 50 floats, with each float being between 1.0 and 100.0. Your task is to calculate the product of all the elements of `alpha` divided by the product of all the elements of `beta`:

$$\frac{\text{alpha}[0] \times \text{alpha}[1] \times \dots \times \text{alpha}[49]}{\text{beta}[0] \times \text{beta}[1] \times \dots \times \text{beta}[49]}$$

Write a function which does this, and returns the answer.

Note: half the marks for the Question will be given for efforts to avoid overflow and underflow.

Question 9

(15 marks total)

Bingo Problem Description

Each player has a bingo card, a grid of cells which are assigned random numbers. The game master calls out numbers randomly. If the number called is on the card, the player places a marker on top of that cell. The winner is the first player on whose card appears a row, column or main diagonal filled with markers. Here is an example

Here is a possible algorithm for the main module of a software implementation of bingo. (Note: this is a simplified version of the algorithm.)

```
Module bingo (NoOfPlayers)
{
    loop {
        if processed all players then
            exit loop
        initialiseCard(card)
        printCard(card)
    }
    set winnerId to 0 i.e false
    loop {
        if winnerId then
            exit loop
        set nextNumber to callNumber()
        for each player
        {
            set card to be that players' card
            updateCard(card, nextNumber)
            printCard(card)
            set winnerId to checkWinner(card) /* returns playerId if winning card, else 0 */
        }
    }
    announceWinner(winnerId)
}
```

- (a) Draw a structure chart for the modules of the Bingo program as shown in the above algorithm (which will indicate the control coupling of the system).
- (b) Add labelled arrows to show the data passed to a module, and passed back from a module (i.e. the data coupling).
- (c) This problem requires a module for generating random numbers. Add that module to your structure chart. (Hint: consider which modules will call it.)

