Course Examination
1st Term 2020-21

ESTR1002

2020-2021 Fall

ESTR1002 Problem Solving by Programming

Online Test

Dec 23, 2020

Time Allowed: 2 hours, 3:30pm to 5:30pm.

Note: 5:30pm to 5:40pm is just buffer time.

INSTRUCTIONS

1. Six questions over **ten** pages.

2. Answer **ALL** questions.

3. Open-book Open-note BUT
   - No discussion in any form.
   - Always unmute yourself
   - Always open camera without virtual background
   - Use the Internet only for Zoom and Blackboard
   - Share your whole desktop screen in the breakout room
   - No public message in Zoom
   - Raise hand (Zoom feature) if you have question
   - Go to restroom before the online test starts
   - Use WeChat/WhatsApp for emergency contact

4. Please prepare one .c file for each problem. Name it as `<student ID>.q<question number>.c` (e.g., "1155100472.q1.c"), and then zip all .c files and submit the zip file to the submission box on our course website on Blackboard: https://blackboard.cuhk.edu.hk

5. Please strictly follow the regulations in "`Guideline_Online_Test.pdf`"

6. Note that even if your program cannot be compiled, we may still give you partial marks to your program. So, please submit "code" for every question.

7. **Time management** is important! Don't use too much time on a single question!

Problem 1 Mobile Plan Payment (15 points)

The Hau Hau mobile phone company has two plans for their users to pay for the mobile data usage in each month:

- Plan A: $0.1 per MB for all data usage below or equal 100MB. If the data usage exceeds 100MB, the price for data usage above 100MB is $0.05 per MB; and
- Plan B: a fixed charge of $15 per month and 300MB data usage. If the data usage exceeds 300MB, the price for data usage above 300MB is $0.075 per MB.

Write a C program that requests the data usage (in MB) in a month, and then prints out the cheaper plan that the consumer should choose.

Input: a nonnegative double number that represents the data usage in a month.

Output: the cheaper plan to be chosen and how much money to be paid (with two decimal points). If both plans lead to the same amount of payment, then print "`equal`".

Note: For accuracy, please use "`double`" for all the variables.

Sample Run #1 (the bold characters in blue are the input entered by the user):

```
10
A 1.00
```

Sample Run #2 (the bold characters in blue are the input entered by the user):

```
200
equal
```

Sample Run #3 (the bold characters in blue are the input entered by the user):

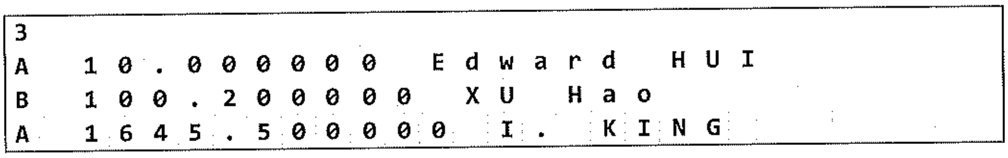
```
201
B 15.00
```

Problem 2 File Write and Read + Data Structure (18 points)

Below shows an incomplete C program with a structure called “`Client`”.

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    char   mobile_plan ;     // A or B
    double data_usage  ;
    char   client_name[102] ;
} Client ;

void printClient( const Client * c )
{
    printf( "%c %f %s\n" , c->mobile_plan , c->data_usage , c->client_name );
}

void writeClientFile( const char * filename, const Client clients[], int num )
{
    // please complete this function (9 points) to
    // - first print out the number of clients to the file
    // - then print out the data of each client to the file
}

int readClientFile( const char * filename , Client clients[] )
{
    // please complete this function (9 points) to
    // - read the client data from the file
    // - update the array "clients"
    // - return the number of client data read
}

void main() {
    Client clients_set1[3] = { { 'A' , 10     , "Edward HUI" } ,
                               { 'B' , 100.2  , "XU Hao"     } ,
                               { 'A' , 1645.5 , "I. KING"    } } ;
    Client clients_set2[100] ;
    int    num_clients ;

    writeClientFile( "client_data.txt" , clients_set1 , 3 );
    num_clients = readClientFile( "client_data.txt" , clients_set2 );

    printClient( & (clients_set2[0]) );
    printClient( & (clients_set2[1]) );
    printClient( & (clients_set2[2]) );
}
```

The contents inside `"client_data.txt"` should look like this:



```
3
A 10.000000 Edward HUI
B 100.200000 XU Hao
A 1645.500000 I. KING
```

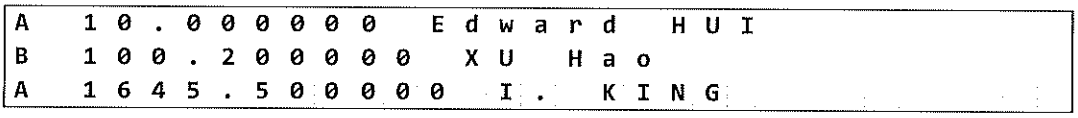
Your task in this question is to complete the two function in the above program to write the client data to a text file and then read back the data back and print it out.

Note that:

- Please submit a complete program.
- If the file writing or the file reading fails, your program should print out an error message to `stderr` and then call `exit(1)`. See sample outputs below.
- The above program template can be downloaded from the Blackboard website after 335pm.

Sample Run #1:

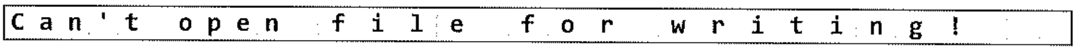
- Successfully finished



```
A 10.000000 Edward HUI
B 100.200000 XU Hao
A 1645.500000 I. KING
```

Sample Run #2:

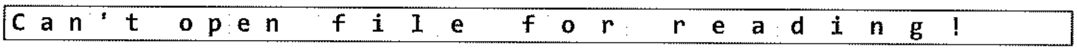
- Upon file writing error



```
Can't open file for writing!
```

Sample Run #3:

- Upon file reading error



```
Can't open file for reading!
```

Problem 3 Phone Numbers in Hong Kong (15 points)

Common phone numbers in Hong Kong are generally eight-digit, for which fixed land line numbers start with 2 or 3 and mobile phone numbers start with 5, 6, 7 or 9. Write a program to read a phone number (as a string) and check if the number is valid, i.e.,

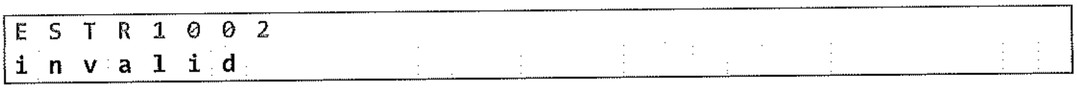
i. the number has exactly eight digits;
ii. the number can only start with "2", "3", "5", "6", "7", or "9"; and
iii. if the number is from the CUHK (Chinese University of Hong Kong), it should start with "3943".

Note that a valid phone number may contain a hyphen in the middle, e.g., 3943-7000.

Input: a string, which may contain letters, digits, and punctuations.

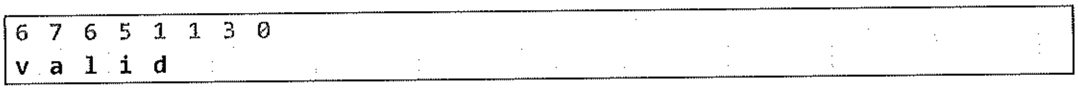
Output: If the input is a valid CUHK number, print "`CUHK`"; if it is valid but not from the CUHK, print "`valid`". Otherwise, print "`invalid`".

Sample Run #1 (the bold characters in blue are the input entered by the user):



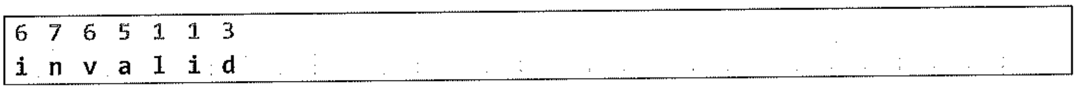
```
ESTR1002
invalid
```

Sample Run #2 (the bold characters in blue are the input entered by the user):



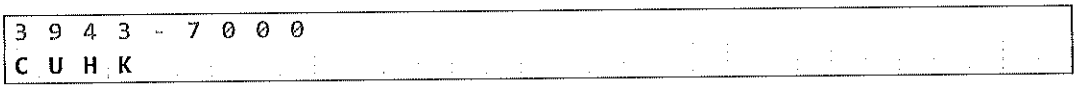
```
67651130
valid
```

Sample Run #3 (the bold characters in blue are the input entered by the user):



```
6765113
invalid
```

Sample Run #4 (the bold characters in blue are the input entered by the user):



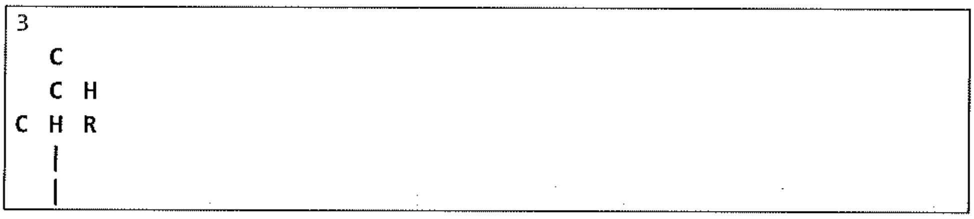
```
3943-7000
CUHK
```

Problem 4 Christmas Tree (15 points)

Write a program that reads a positive integer n from the user (assuming no error in the user input), and then print out the Christmas Tree pattern as shown in the sample runs below. Note that n can be assumed to be an integer from 3 to 20. The top part of the tree is made up of the nine-letter word "CHRISTMAS" in a cyclic manner. Also, do not forget the trunk of the tree with two '|'.

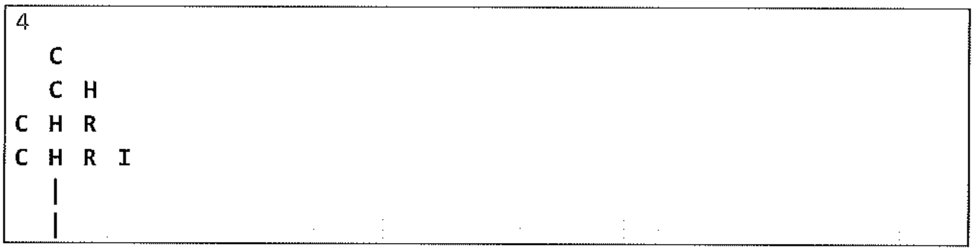
Hint: you may define function(s) to help to simplify your code.

Sample Run #1 (the bold characters in blue are the input entered by the user):



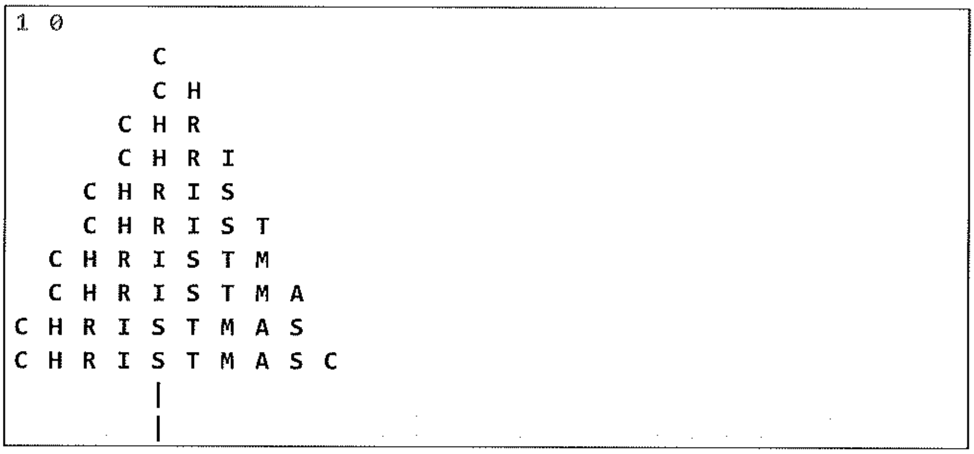
```
3
  C
  C H
C H R
  |
  |
```

Sample Run #2 (the bold characters in blue are the input entered by the user):



```
4
  C
  C H
C H R
C H R I
  |
  |
```

Sample Run #3 (the bold characters in blue are the input entered by the user):



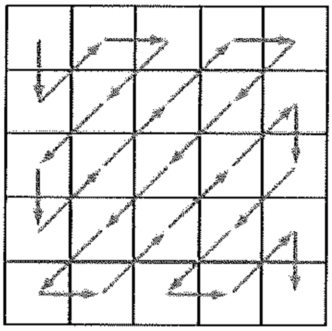
```
1 0
        C
        C H
      C H R
      C H R I
    C H R I S
    C H R I S T
  C H R I S T M
  C H R I S T M A
C H R I S T M A S
C H R I S T M A S C
        |
        |
```

Sample Run #4 (the bold characters in blue are the input entered by the user):

```
1 3
            C
            C H
          C H R
          C H R I
        C H R I S
        C H R I S T
      C H R I S T M
      C H R I S T M A
    C H R I S T M A S
    C H R I S T M A S C
  C H R I S T M A S C H
  C H R I S T M A S C H R
C H R I S T M A S C H R I
            |
            |
```

Problem 5 Number Pattern (18 points)

Write a program that reads a positive integer n from the user (assuming no error in the user input), and then arranges and prints integers 1 to n^2 in the form of a 2D matrix following the pattern below (which shows the case of n=5):



Here, n ranges 1 to 12. Also, note the whitespaces padded to align the numbers in the sample outputs below.

Sample Run #1 (the bold characters in blue are the input entered by the user):

```
3
   1   3   4
   2   5   8
   6   7   9
```

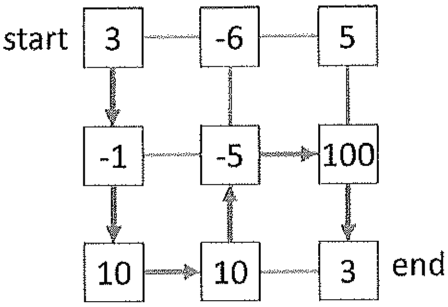
Sample Run #2 (the bold characters in blue are the input entered by the user):

```
6
   1   3   4  10  11  21
   2   5   9  12  20  22
   6   8  13  19  23  30
   7  14  18  24  29  31
  15  17  25  28  32  35
  16  26  27  33  34  36
```

Problem 6 Maximum Profit (19 points)

The input to this problem has two parts:

i. a positive integer n, ranged from 1 to 15; and
ii. a 2D array of n x n integers (each can be zero, positive, or negative).



The figure above shows an example input of 3x3 cities (n=3). We aim to find a path **(with only down, right, and/or up moves)** to traverse from start to end with the maximum profit that one can get. In the example shown above, the best path has a total profit of 120.

You may view the 2D array as a 2D grid of n by n cities, in which you start from the top left corner and aim to go to the bottom right corner with only three kinds of movements: **down, up, or right.** Note also that you **cannot visit the same city more than once!**

Write a C program that first reads the data (i.e., integer n followed by n x n integers) and then finds out the maximum profit that one can obtain.

Sample Run #1 (the bold characters in blue are the input entered by the user):

```
2
1    3
2    4
8
```

Sample Run #2 (the bold characters in blue are the input entered by the user):

```
3
3    - 6    5
- 1    - 5    1 0 0
1 0    1 0    3
1 2 0
```

A program code template is given below.

```
#include <stdio.h>

#define MAX_DIM 15

void printMat( int data[MAX_DIM][MAX_DIM] , int n )
{
    int i , j ;
    for ( j = 0 ; j < n ; j ++ )
    {
        for ( i = 0 ; i < n ; i ++ )
            printf(" %d",data[j][i]);
        printf("\n");
    }
}


// you may create more functions here


void main()
{
    int data[MAX_DIM][MAX_DIM] , results[MAX_DIM][MAX_DIM] , n , i , j ;

    // read from users
    scanf("%d",&n);
    for ( j = 0 ; j < n ; j ++ )
        for ( i = 0 ; i < n ; i ++ )
            scanf("%d",&(data[j][i]));
    //printf("input:\n");
    //printMat(data,n);


    // your code here


    // report the result
    printf("%d\n",results[n-1][n-1]);
}
```

Note that the above template can be downloaded from the Blackboard website after 335pm. You may start with your own code or start with the above code to save time.

END OF PAPER