香港公開大學 科技學院
THE OPEN UNIVERSITY OF HONG KONG
SCHOOL OF SCIENCE AND TECHNOLOGY

# ELEC S212

# Network Programming and Design

---

**2018 Autumn Presentation**

**Assignment 2**

Please e-submit this assignment via the OLE
by **27 Jan 2019, 23:59**

## Question 1    (20 marks)

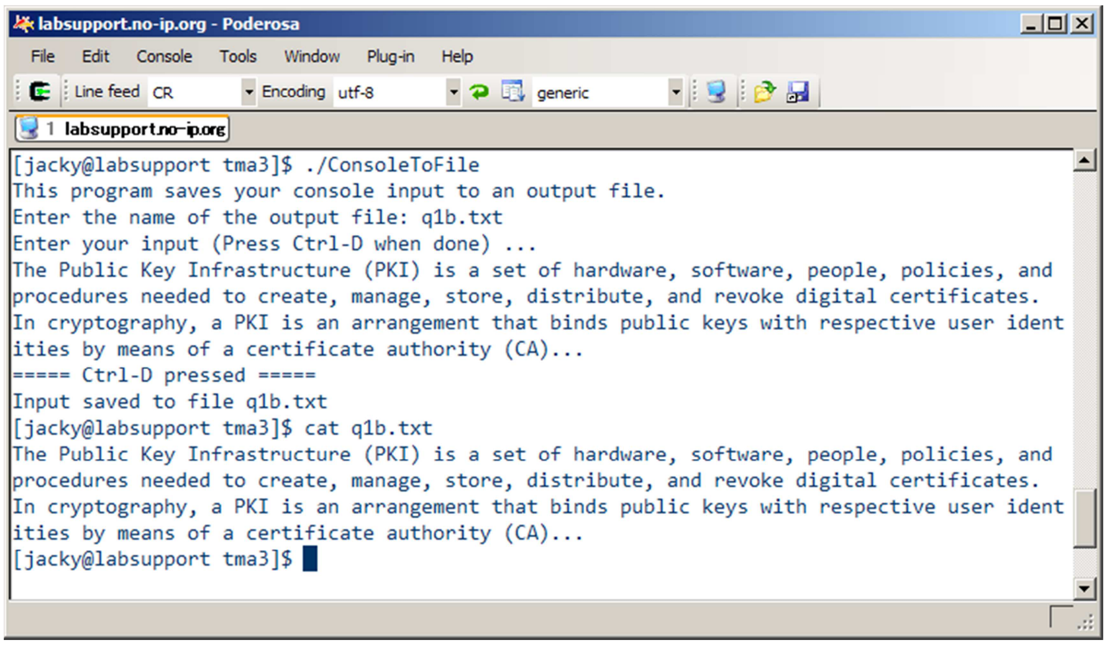
(a) In compiling a C program into an executable file, you will need to deal with the following items (either directly or indirectly). Briefly explain the meaning or function of each of them. Use an example to illustrate its meaning or function where appropriate. **[12 marks]**

1) Source files
2) Preprocessor
3) Compiler
4) Assembler
5) Linker
6) Object code and object files
7) Executable file
8) Header files and libraries

(b) The following program contains **5 errors (bugs)**.

```
#include stdio.h
#include stdlib.h

int main(void)
{
  FILE *outputFile;
  char file_name[];
  int ch;

  printf("This program saves your console input to an output file.\n");
  printf("Enter the name of the output file: ");
  scanf("%s", file_name);
  flush_input_buffer();

  if (outputFile = fopen(file_name, "w") == NULL) {
       printf("Cannot open %s for writing.\n", file_name);
       return EXIT_FAILURE;
  }

  printf("Enter your input (Press Ctrl-D when done) ...\n");

  while ((ch = getc(stdin)) != EOF) ; {
       putc(ch, outputFile);
  }

  fclose(outputFile);

  printf("===== Ctrl-D pressed ===== \n");
  printf("Input saved to file %s\n", outputFile);

  return EXIT_SUCCESS;
}

void flush_input_buffer() {
  while (getchar() != '\n') {
       continue;
  }
}
```

The following figure shows an example run of the correct program.



```
[jacky@labsupport tma3]$ ./ConsoleToFile
This program saves your console input to an output file.
Enter the name of the output file: q1b.txt
Enter your input (Press Ctrl-D when done) ...
The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and
procedures needed to create, manage, store, distribute, and revoke digital certificates.
In cryptography, a PKI is an arrangement that binds public keys with respective user ident
ities by means of a certificate authority (CA)...
===== Ctrl-D pressed =====
Input saved to file q1b.txt
[jacky@labsupport tma3]$ cat q1b.txt
The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and
procedures needed to create, manage, store, distribute, and revoke digital certificates.
In cryptography, a PKI is an arrangement that binds public keys with respective user ident
ities by means of a certificate authority (CA)...
[jacky@labsupport tma3]$
```

1) Locate the five errors. Explain why they are errors and how they should be corrected. **[5 marks]**

2) Correct the program and save it as “ConsoleToFile.c” inside a directory named “tma2” under your home directory. Compile the program to produce an executable file named “ConsoleToFile” in the same directory. **[3 marks]**

## Question 2 (35 marks)

Complete Lab 2.2 – Internet application development using C.

(a) You are required to place your source files, executables files, and web pages as follows:

**Step 1:**

Source file "`fibonacci.c`" and executable file "`fibonacci`" in the "`tma2/lab2.2`" directory under your home directory.

Source file "`fibonacci.cgi.c`" and executable file "`fibonacci.cgi`" in the "`public_html/cgi-bin`" directory.

**Step 2:**

Web page "`program2.html`" in the "`public_html`" directory under your home directory.

Source file "`fibonacci2.cgi.c`" and executable file "`fibonacci2.cgi`" in the "`public_html/cgi-bin`" directory.

**Step 3:**

Web page "`program3.html`" in the "`public_html`" directory under your home directory.

Source file "`fibonacci3.cgi.c`" and executable file "`fibonacci3.cgi`" in the "`public_html/cgi-bin`" directory.

**[10 marks]**

(b) Answer questions 1 to 6 at the end of the Lab. Place all your C source files and executable files in the "tma2/lab2.2" directory under your home directory and all CGI programs in the "public_html/cgi-bin" directory. **[25 marks]**

## Question 3 (20 marks)

In Unit 5, you learn about the structures in C. "A structure is a collection of one or more variables, possibly of different types, grouped together under a single name for convenient handling… Structures help to organize complicated data, particularly in large programs, because they permit a group of related variables to be treated as a unit instead of as separate entities."

In Reading 5.13, the author introduces the `struct date`, as an example to illustrate the use of structures. Study the programs in this reading carefully and make sure you understand how to make use of structures and pointers to structures.

Your task is to write a C program `printdate` that can print out a date in the American date format (e.g., December 25, 2016). The year, month and day of the date are supplied as command line arguments. If they represent a correct date, the date is printed out in the American date format in the console display. Otherwise, an error message is printed. If an incorrect number of command line arguments are supplied, the program prints a message showing its proper usage syntax.

You must also consider the leap year in order to earn full marks for this question.

The following shows the expected output of the correctly implemented program executed with different command line arguments:

```
[joe@labsupport tma2]$ ./printdate
Usage: printdate year month day
[joe@labsupport tma2]$ ./printdate 1
Usage: printdate year month day
[joe@labsupport tma2]$ ./printdate 1 2
Usage: printdate year month day
[joe@labsupport tma2]$ ./printdate 1 2 3
February 3, 1
[joe@labsupport tma2]$ ./printdate 2008 12 25
December 25, 2008
[joe@labsupport tma2]$ ./printdate 2010 2 29
Error: invalid day.
[joe@labsupport tma2]$ ./printdate 2000 2 29
February 29, 2000
[joe@labsupport tma2]$ ./printdate a 1 2
Error: year must be a number.
[joe@labsupport tma2]$ ./printdate 1 a 2
Error: month must be a number..
[joe@labsupport tma2]$ ./printdate 1 2 a
Error: day must be a number.
[joe@labsupport tma2]$ ./printdate 1997 7 1
July 1, 1997
[joe@labsupport tma2]$ ./printdate 1997 7 32
Error: invalid day.
[joe@labsupport tma2]$
```

The following listing shows the incomplete program code of **`printdate`**. You may develop your program based on the code in this listing:

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

#define PROGRAM_NAME            "printdate"
#define CORRECT_INPUT_COUNT     4

typedef struct
{
        int year;
        int month;
        int day;
} Date;

const char *MONTH_NAMES[] = {
        "Invalid month",
        "January", "February", "March", "April",
        "May", "June", "July", "August",
        "September", "October", "November", "December"
};

const int DAYS_PER_MONTH[] = {
        0,
        31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31
};

// Checks whether the correct number of command line arguments have been supplied.
// If not, prints the usage message and exits the program.
void check_input_count(int);

// Prints the usage message.
void print_usage(void);

// Prints the supplied Date structure.
void print_date(Date*);

// Checks whether the supplied date is a leap year.
bool is_leap_year(Date*);

// Checks whether the supplied date has a valid year.
bool check_year(Date*);

// Checks whether the supplied date has a valid month.
bool check_month(Date*);

// Checks whether the supplied date has a valid day.
bool check_day(Date*);

// Checks whether the supplied date is valid.
bool check_date(Date*);

// Sets the fields of the supplied date structure from the command line parameters.
bool get_date(Date*, char**);
```

```

// The main program
int main(int argc, char *argv[])
{
    // Your code starts here
```

You are required to name your source file and executable file “**`printdate.c`**” and “**`printdate`**” respectively. Place them in the “**`tma2`**” directory under your home directory on the server.

## Question 4 (25 marks)

**Answer the following questions in your own words.**

(a) A client program must take several steps to communicate with a server program. Describe with an illustration all the basic steps in creating a socket, setting up the destination host address, establishing the channel, sending and receiving data, and finally terminating the connection. **[10 marks]**

(b) What does the socket function `bind()` do? Explain why, in general, server programs need to use the `bind()` function but client programs do not. Can you describe a situation in which the client program really needs to use the `bind()` function? **[8 marks]**

(c) Write a C program `hostname2ip` that can look up the IP address of a given host name specified by the command line argument (Hint: use the function `gethostbyname()`).Place your work in the "tma2" directory under your home directory. **[7 marks]**