Introduction to Computer Science: Final Exam

January 19, 2018. 9:30 - 11:00

Name: ______________________________

Student ID: _________________________

Instructions: This paper consists of 50 multiple choice questions. You have to answer all of them. Each carries two marks. Each question has only one correct option. Please put down the answer clearly next to the question number.

The information regarding the artificial CPU and the source codes of the programs for answering the questions are put in the end of the paper.

Reminder: Beware of careless mistake!

Question 1

In accordance with John Sum's viewpoint, which of the following systems is inevitable in an organization?

(a) Transaction processing system.

(b) Management information system.

(c) Decision support system.

(d) Executive information system.

Question 2

Which of the following firms has applied AI technologies in their services or products?

(i) Google

(ii) Apple

(iii) Facebook

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 3

Which of the following software has an interpreter inside?

(i) Window OS

(ii) Chrome

(iii) Matlab

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 4

Which of the following statement about the interaction between an application software and the operating system is *False*?

(a) Once an application software has a request for an operating system, it simply sends the request message on the service queue of the operating system.

(b) Once the operating system has completed a request, it simply sends the results on the receiving queue of the application software.

(c) Normally, an application software has no right to instruct the operating system to change the priority of its request.

(d) Normally, an operating system does not change the priority of a request from any application software.

(e) Normally, an operating system will serve the requests based on *first-come-first-serve* principle.

(f) None of them is false.

Question 5

How does an application software know that the service request to an operating system has completed?

(a) The application software from time to time will send a message to the operating system checking for the progress.

(b) The application software from time to time will check from its queue seeing if there is anything there.

(c) The operating system from time to time will send a message to the application software informing the progress.

(d) The operating system from time to time will check from its queue seeing if there is any *checking-for-progress* message coming.

Question 6

Which of the following statement about medium access control (MAC) is true?

(a) If a computer has detected that the medium is being in use, the computer will wait for a fixed period of time and then re-detect the medium seeing if it is in use. If the medium is not in use, it sends out the message to the medium.

(b) If a computer has detected that the medium is being in use, the computer will wait for a random period of time and then re-detect the medium seeing if it is in use. If the medium is not in use, it sends out the message to the medium.

(c) If a computer has detected that the medium is not in use, the computer will wait for a fixed period of time and then re-detect the medium seeing if it is in use. If the medium is not in use, it sends out the message to the medium.

(d) If a computer has detected that the medium is not in use, the computer will wait for a random period of time and then re-detect the medium seeing if it is in use. If the medium is not in use, it sends out the message to the medium.

Question 7

Which of the following statements about *port number* are true?

(i) Each application server (web server for instance) is usually assigned to a unique port number.

(ii) Once a message has arrived a computer, the network operating system will check the port number specified in the message and then pass the message to the corresponding application server queue.

(iii) Two different application servers, say email server and web server, can use the same port number.

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 8

Which of the following statements about *network protocol* are true?

(i) Network protocol is a collection of agreements for the network developers to follow so that every message sending from one computing device can be received and recovered by the another computing device if they are connected to the Internet.

(ii) The format of IP addresses is part of the protocol.

(iii) The physical signal, like electrical signal and radio signal, representing symbol ”0” (resp. ”1”) is part of the protocol.

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 9

Which of the following numbers is the protocol number of wireless communication?

(a) 802.3

(b) 802.4

(c) 802.8

(d) 802.11

Question 10

DNS refers to domain name server. Which of the following statements about DNS is incorrect?

(a) Each DNS consists of a table showing the mapping of the IP addresses of some domain names in the world.

(b) All the DNSs form a DNS network to facilitate the search of IP address of a domain name.

(c) Even if all the DNSs are down, we are still able to access a homepage.

(d) None of the above is incorrect.

Question 11

Which of the following are possible causes for not able to access www.nchu.edu.tw?

(i) The DNS your computer is linking to is down.

(ii) The cable connecting your computer and the Internet is disconnected.

(iii) Your ISP is currently unable to connect to the Internet.

Answer :

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 12

Suppose your cell phone (with enough battery power) has been set to the following conditions but your cell phone is not able to access www.nchu.edu.tw. However, you can still make a phone call. Which of the following is likely the cause of this connection failure?

```
WiFi              ON
Bluetooth         OFF
Cellular          ON
Personal Hotspot  OFF
Carrier           ON
```

(a) There is no available WiFi around.

(b) Personal Hotspot is OFF.

(c) Bluetooth is OFF.

(d) Web server www.nchu.edu.tw is down.

Question 13

If we would like to show the following text on a webpage, which of the following codes should be added?

statistical analysis.

In the HTML file, what would be the correct coding for it?

(a) `<b></i>statistical analysis<i></b>`

(b) `<i></b>statistical analysis<i></b>`

(c) `</b><i>statistical analysis</i><b>`

(d) `<i><b>statistical analysis</i></b>`

(e) None of the above.

Question 14

Which of the following commands are valid to be added in a HTML file? Here, the files ”john2009.jpg” and ”poker01.cpp” are located in the same directory as the one for ”index.html”.

(i) `<a href="www.nchu.edu.tw">NCHU</a>`

(ii) `<a href="john2009.jpg">John</a>`

(iii) `<a href="poker01.cpp">Poker</a>`

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 15

Here is a segment of HTML code.

```
<table>
<tr> <td>Instructor:</td>
     <td>John Sum</td> </tr>
<tr> <td>Office:</td>
     <td>Room 821</td></tr>
<tr> <td width>Time:</td>
     <td>Friday 09:10-12:00 </td> </tr>
<tr> <td>Classroom:</td>
     <td>CS121</td> </tr>
</table>
```

Once the above program segment is interpreted, what will you see on the browser.

(a)
```
Instructor: John Sum
Office: Room 821
Time: Friday 09:10-12:00
Classroom: CS121
```

(b)
```
Instructor:     John Sum
Office:         Room 821
Time:           Friday 09:10-12:00
Classroom:      CS121
```

(c)
```
Instructor: John Sum
Office:     Room 821
Time:       Friday 09:10-12:00
Classroom:  CS121
```

(d)
```
Instructor: John Sum
    Office: Room 821
      Time: Friday 09:10-12:00
 Classroom: CS121
```

Question 16

In a HTML file, what information could be put in the header segment?

(i) Title of the webpage.

(ii) Keywords of the webpage.

(iii) Style of the webpage.

Answer:

(a) (i) only.

(b) (ii) only.

(c) (iii) only.

(d) (i) and (ii) only.

(e) (ii) and (iii) only.

(f) (i) and (iii) only.

(g) All of the above.

(h) None of the above.

Question 17

John encrypts a letter by using Mary's public key and then sends it to Mary. Somehow Peter gets the letter and destroys it. Besides, Peter acts on behalf of John to prepare another letter and encrypts it by using Mary's public key. Finally, Mary receives the letter from Peter. Which of the following will happen?

(a) Mary can read the letter and identify that it is not from John.

(b) Mary can read the letter and identify that it is from Peter.

(c) Mary can read the letter but she fails to identify the sender of the letter.

(d) None of the above.

Question 18

John encrypts a letter by using his own private key and Mary's public key. Then, the letter is sent to Mary. Somehow Peter gets the letter and destroys it. Besides, Peter acts on behalf of John to prepare another letter and encrypts it by using his own private key and Mary's public key. Finally, Mary receives the letter from Peter. Which of the followings will happen?

(a) Mary can read the letter and identify that it is from Peter.

(b) Mary cannot read the letter but she can identify it is from Peter.

(c) Mary can read the letter but fail to identify the sender of the letter.

(d) Mary cannot read the letter and fail to identify the sender of the letter.

Question 19

Which of the following methods is NOT a correct method for encryption?

(a) John encrypts the letter by using his public key. While Mary has received the message, she decrypts the message by using John's public key.

(b) John encrypts the letter by using Mary's public key. While Mary has received the message, she decrypts the message by using her private key.

(c) John encrypts the letter by using Mary's private key and then by using his public key. While Mary has received the message, she decrypts the message first by using John's private key and then by using her public key.

(d) John encrypts the letter by using Mary's public key and then by using his private key. While Mary has received the message, she decrypts the message first by using John's public key and then by using her private key.

Question 20

A CEO would like to send an official announcement to his colleagues. What he can do to let his colleagues to authenticate that the announcement is from him?

(a) He can encrypt the message by using his private key.

(b) He can encrypt the message by using his public key.

(c) He can encrypt the message first by using his private key and then re-encrypt the cipher text by using his public key.

(c) He can encrypt the message first by using his public key and then re-encrypt the cipher text by using his private key.

Question 21

Refer to the artificial CPU and its commands, what will be the content of $M4$ if the following commands are executed?

```
DEF M1 0
DEF M2 2
DEF M3 5

MOV IA M1
IF IA == 0
   MOV IA M2
   MOV IB M3
   ADD IA IB
   MOV M4 OUT
ELSE
   MOV IA M1
   ADD IB M2
   MOV M4 OUT
ENDIF
```

(a) 2.

(b) 7.

(c) 5.

(d) 0.

Question 22

What will be the content of $M4$ if the following program segment is executed?

```
DEF M1 16
DEF M2 22
DEF M3 10
MOV IA M1
MOV IB M2
CMP IA IB
MOV M4 OUT
MOV IA M2
MOV IB M3
CMP IA IB
MOV IA OUT
MOV IB M4
ADD IA IB
MOV M4 OUT
```

(a) 28.

(b) 30.

(c) 32.

(d) 34.

Question 23

Refer to the artificial CPU and its commands, what will be the content of $M4$ if the following commands are executed?

```
DEF M1 0
DEF M2 2
DEF M3 5

MOV IA M1
IF IA == 0
   MOV IA M2
   SHL IA 00000100
   MOV IA OUT
   MOV IB M2
   ADD IA IB
   MOV M4 OUT
ELSE
```

```
   MOV IA M3
   SHL IA 00000100
   MOV IA OUT
   MOV IB M3
   ADD IA IB
   MOV M4 OUT
ENDIF
```

(a) 4.

(b) 6.

(c) 8.

(d) 10.

Question 24

Three numbers have been stored in M1, M2 and M3. Which of the following program segments can correctly give the output of the following formulae?

$$M4 = M1 + M2 \times M3.$$

(a)
```
----------------
MOV IA M1
MOV IB M2
MUL IA IB
MOV IA OUT
MOV IB M3
ADD IA IB
MOV M4 OUT
----------------
```

(b)
```
----------------
MOV IA M1
MOV IB M2
ADD IA IB
MOV IA OUT
MOV IB M3
MUL IA IB
MOV M4 OUT
----------------
```

(c)
```
----------------
MOV IA M2
MOV IB M3
MUL IA IB
MOV IA OUT
MOV IB M1
ADD IA IB
MOV M4 OUT
----------------
```

(d)
```
----------------
MOV IA M2
MOV IB M3
ADD IA IB
MOV IA OUT
MOV IB M1
MUL IA IB
MOV M4 OUT
----------------
```

Question 25

Given that there are five memories M1, M2, M3, M4 and M5. Here is the program segment to instruct the circuit.

```
----------------
MOV IA M1
MOV IB M2
MUL IA IB
MOV M5 OUT
MOV IA M3
MOV IB M4
MUL IA IB
MOV IA OUT
MOV IB M5
ADD IA IB
MOV M5 OUT
----------------
```

Which of the following mathematical equation is identical to the operation of the following program segment?

(a) $M5 = M1 + M2 \times M3 + M4.$

(b) $M5 = (M1 + M2) \times M3 + M4$

(c) $M5 = M1 \times (M2 + M3) \times M4.$

(d) $M5 = M1 \times M2 + M3 \times M4.$

Question 26

Given that there are five memories M1, M2, M3, M4 and M5. Here is the program segment to instruct the circuit.

```
----------------
MOV IA M1
MOV IB M2
MUL IA IB
MOV IA OUT
MOV IB M3
MUL IA IB
MOV IA OUT
MOV IB M4
SUB IA IB
MOV M5 OUT
----------------
```

which of the following mathematical equation is identical to the operation of the following program segment?

(a) $M5 = M4 - M1 \times M2 \times M3$.

(b) $M5 = M4 - (M1 + M2) \times M3$

(c) $M5 = M1 \times M2 \times M3 - M4$.

(d) $M5 = (M1 + M2) \times M3 - M4$.

Question 27

If the content in $M1$ is either ’1’ or ’0’, which of the following mathematical equation is identical to the operation of the following program segment?

```
MOV IA M1
IF IA == 0
   MOV IA M2
   SHL IA 00000100
   MOV IA OUT
   MOV IB M2
   ADD IA IB
   MOV M4 OUT
ELSE
   MOV IA M3
   SHL IA 00000100
   MOV IA OUT
   MOV IB M3
   ADD IA IB
   MOV M4 OUT
ENDIF
```

(a) $M1 \times (9 \times M2) + (1 - M1) \times (9 \times M3)$.

(b) $M1 \times (5 \times M2) + (1 - M1) \times (5 \times M3)$.

(c) $(1 - M1) \times (9 \times M2) + M1 \times (9 \times M3)$.

(d) $(1 - M1) \times (5 \times M2) + M1 \times (5 \times M3)$.

Question 28

Which of the following program segments perform exactly the same thing as the program segment below?

```
for (i=0,i<3, i++)
  printf("*\n");
printf("Program end!");
```

(a)
```
i = 0;
while( i<3)
{ printf("*\n");
printf("Program end!");
i = i + 1; }
```

(b)
```
i = 0;
while( i<3)
{ printf("*\n"); i = i + 1; }
printf("Program end!");
```

(c)
```
i = 0;
while(i<3)
printf("*\n"); i = i + 1;
printf("Program end!");
```

(d)
```
i = 1;
while(i<3)
{ printf("*\n"); i = i + 1; }
printf("Program end!");
```

Question 29

If the following C program segment is executed, what will you see on the screen?

```
i=0;
While (i<=5)
{
  printf("*");
  i = i+1;
}
printf("%d\n", i+1);
```

(a)
```
*****5
c:\>
```

(b)
```
*****5

c:\>
```

(c)
```
*****6
c:\>
```

(d)
```
*****6

c:\>
```

(e) None of the above.

Question 30

Which of the following statements about comment delimiters ”/*” and ”*/” in C program are true?

(i) Whenever the compiler has encountered the comment delimiters, all the content inside the delimiters will be ignored.

(ii) Comment delimiters are used for adding the change history in a source code.

(iii) Adding comments in a source code is a way to help the programmer easily understand or recall the flow of the program.

Answer:

(a) (i) and (ii) only.

(b) (ii) and (iii) only.

(c) (i) and (iii) only.

(d) (i), (ii) and (iii).

Question 31

What will you see on the console if the following program segment is executed?

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

main()
{
int i;

for(i=0; i<4; i++)
    printf("%d ", i%3);
printf("\n");
}
```

(a)
```
0 1 2 3
c:\>
```

(b)
```
0 1 2 3

c:\>
```

(c)
```
0 1 2 0
c:\>
```

(d)
```
0 1 2 0

c:\>
```

Question 32

What will you see on the console if the following program segment is executed?

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

main()
{
int i, tmp, A[5];

A[0]=6; A[1]=2; A[2]=8;
A[3]=1; A[4]=5;

for(i=0; i<4; i++)
    if(A[i] > A[i+1])
    {
    tmp = A[j];
    A[j] = A[j+1];
    A[j+1] = tmp;
    }

for(i=0; i<5; i++)
    printf("%d ", A[i]);
}
```

(a)
```
1 2 5 6 8
c:\>
```

(b)
```
8 6 5 2 1
c:\>
```

(c)
```
2 6 8 1 5
c:\>
```

(d)
```
6 8 2 5 1
c:\>
```

Question 33

What will you see on the console if the following program segment is executed?

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

main()
{
int i, A[5];
A[0]=6; A[1]=2; A[2]=8;
A[3]=1; A[4]=5;

for(i=0; i<4; i++)
    if(A[i] < A[i+1])
    {
    tmp = A[j];
    A[j] = A[j+1];
    A[j+1] = tmp;
    }
for(i=0; i<5; i++)
    printf("%d ", A[i]);
}
```

(a)
```
1 2 5 6 8
c:\>
```

(b)
```
8 6 5 2 1
c:\>
```

(c)
```
2 6 8 1 5
c:\>
```

(d)
```
6 8 2 5 1
c:\>
```

Question 34

Refer to Program A. What will you see after you have entered ”a”?

(a)
```
You have successfully registered MKT.
You are now the 45 number of student.
c:\>
```

(b)
```
You have successfully registered MKT.
You are now the 46 number of student.

c:\>
```

(c)
```
You have successfully registered ACCT.
You are now the 35 number of student.
c:\>
```

(d)
```
You have successfully registered ACCT.
You are now the 36 number of student.

c:\>
```

(e) None of the above.

Question 35

Refer to Program A. What will you see if you have miss-typed ”a” by ”s”?

(a)
```
You have successfully registered MKT.
You are now the 45 number of student.
c:\>
```

(b)
```
You have successfully registered MKT.
You are now the 46 number of student.

c:\>
```

(c)
```
You have successfully registered ACCT.
You are now the 35 number of student.
c:\>
```

(d)
```
You have successfully registered ACCT.
You are now the 36 number of student.

c:\>
```

(e) None of the above.

Question 36

Refer to Program B. If the following command is typed on the console, what will you see on the screen?

```
c:\>bubblesort ascend 3 6 5 4 7 9 11
```

(a)
```
a 3 4 5 6 7 9 11

c:\>
```

(b)
```
a 3 4 5 6 7 9 11
c:\>
```

(c)
```
3 4 5 6 7 9 11

c:\>
```

(d)
```
3 4 5 6 7 9 11
c:\>
```

(e) None of the above.

Question 37

Refer to Program B. If the following command is typed on the console, what will you see on the screen?

```
c:\>bubblesort 3 6 5 4 7 9 11
```

(a)
```
The sorted numbers are 11 9 7 6 5 4.
c:\>
```

(b)
```
The sorted numbers are 11 9 7 6 5 4.

c:\>
```

(c)
```
The sorted numbers are 4 5 6 7 9 11.
c:\>
```

(d)
```
The sorted numbers are 4 5 6 7 9 11.

c:\>
```

(e) None of the above.

Question 38

Refer to Program B. What is the maximum number of numbers the program can sort?

(a) 16

(b) 20

(c) 24

(d) 28

(e) None of the above.

Question 39

Refer to Program B. What is the value of ”argc” if the following command is typed?

```
c:\>bubblesort ascend 3 8 4 7 9 11
```

(a) 8

(b) 9

(c) 10

(d) 11

(e) None of the above.

Question 40

Refer to Program B. What is the content of ”argv[0]” if the following command is typed?

```
c:\>bubblesort ascend 3 8 4 7 9 11
```

(a) bubblesort

(b) ascend

(c) 3

(d) 8

(e) None of the above.

Question 41

Refer to Program C. If your option of sorting style is ”a” the the numbers entered are ”4 5 2 1 6”, what will you see below the numbers you entered?

(a)
```
The sorted numbers are 1 2 4 5 6.
c:\>
```

(b)
```
The sorted numbers are 1 2 4 5 6.

c:\>
```

(c)
```
The sorted numbers are 6 5 4 2 1.
c:\>
```

(d)
```
The sorted numbers are 6 5 4 2 1.

c:\>
```

(e) None of the above.

Question 42

Refer to Program C. If your option of sorting style is ”ascend” the the numbers entered are ”4 5 2 1 6”, what will you see below the numbers you entered?

(a)
```
The sorted numbers are 1 2 4 5 6.
c:\>
```

(b)
```
The sorted numbers are 1 2 4 5 6.

c:\>
```

(c)
```
The sorted numbers are 6 5 4 2 1.
c:\>
```

(d)
```
The sorted numbers are 6 5 4 2 1.

c:\>
```

(e) None of the above.

Question 43

Refer to Program C. If your option of sorting style is ”descend” the the numbers entered are ”4 5 2 1 6”, what will you see below the numbers you entered?

(a)
```
The sorted numbers are 1 2 4 5 6.
c:\>
```

(b)
```
The sorted numbers are 1 2 4 5 6.

c:\>
```

(c)
```
The sorted numbers are 6 5 4 2 1.
c:\>
```

(d)
```
The sorted numbers are 6 5 4 2 1.

c:\>
```

(e) None of the above.

Question 44

Refer to Program C. If your option of sorting style is ”s” the the numbers entered are ”4 5 2 1 6”, what will you see below the numbers you entered?

(a)
```
The sorted numbers are 1 2 4 5 6.
c:\>
```

(b)
```
The sorted numbers are 1 2 4 5 6.

c:\>
```

(c)
```
The sorted numbers are 6 5 4 2 1.
c:\>
```

(d)
```
The sorted numbers are 6 5 4 2 1.

c:\>
```

(e) None of the above.

Question 45

Refer to Program D. If your option of sorting style is "d" the the numbers entered are "4 5 2 1 6", what will you see below the numbers you entered?

(a)
```
The sorted numbers are 1 2 4 5 6.
c:\>
```

(b)
```
The sorted numbers are 6 5 4 2 1.
c:\>
```

(c)
```
The sorted numbers are 1 2 4 5 6.

c:\>
```

(d)
```
The sorted numbers are 6 5 4 2 1.

c:\>
```

(e) None of the above.

Question 46

Refer to Program D. If your option of sorting style is "s", what will you see on the console?

(a)
```
Please enter five numbers.
Enter the 1 number:
```

(b)
```
Please enter five numbers.

Enter the 1 number:
```

(c) The screen will show "Select a sorting style, 'a' for ascending and 'd' for descending." and then the "Option :" will appear on the line just below this sentence.

(d) The screen only show the following thing.

```
Option :
```

Question 47

Refer to Program E. Which number is assigned to "Heart 4" during the initialization step?

(a) 8.

(b) 16.

(c) 32.

(d) No one knows.

Question 48

Refer to Program E. What will be the last four cards to be displayed?

(a) No one knows, as the cards are shuffled randomly.

(b) S10, S11, S12 and S13.

(c) C10, C11, C12 and C13.

(b) D10, D11, D12 and D13.

Question 49

Refer to Program E. What will you see on the console once the program "poker01.exe" is executed? (Note that the actual cards to be displayed for each player are unknown. Here, I put them arbitrarily. Owing to save space, the cards for the Player 3 to Player 11 are omitted.)

(a)
```
The card are being shuffled now.
Here are the cards.
Player 1: S5 S1 S6 C6
Player 2: S8 H11 C3 H13
......
Player 12: S13 C12 C10 C8
Player 13: D10 C2 H12 S9
c:\>
```

(b)
```
The card are being shuffled now.
Here are the cards.
Player 1: S5 S1 S6 C6
Player 2: S8 H11 C3 H13
......
Player 12: S13 C12 C10 C8
Player 13: D10 C2 H12 S9

c:\>
```

(c)
```
The card are being shuffled now.
Here are the cards.
Player  1: S5 S1 S6 C6
Player  2: S8 H11 C3 H13
```

```
......
Player 12: S13 C12 C10 C8
Player 13: D10 C2 H12 S9
c:\>
```

(d)
```
The card are being shuffled now.
Here are the cards.
Player  1: S5 S1 S6 C6
Player  2: S8 H11 C3 H13
......
Player 12: S13 C12 C10 C8
Player 13: D10 C2 H12 S9

c:\>
```

Question 50

Refer to Program E. What will be the last four cards to be displayed if the "Card Shuffling" step is implemented by the following program segment?

```
for(i=0; i<MAX; i++)
{
    tmp1 = rand() % (RANGE - 8);
    tmp2 = rand() % (RANGE - 8);
    tmp = CARD[tmp1];
    CARD[tmp1] = CARD[tmp2];
    CARD[tmp2] = tmp;
}
```

(a) No one knows, as the cards are shuffled randomly.

(b) S10, S11, S12 and S13.

(c) C10, C11, C12 and C13.

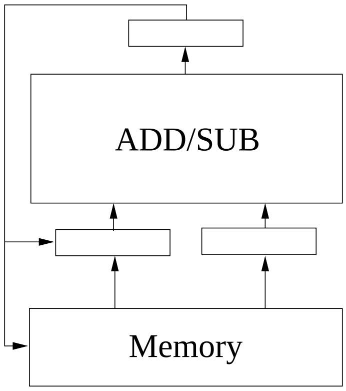
(b) D10, D11, D12 and D13.

APPENDIX

In this appendix, it includes the information about the artificial CPU and the source codes of five programs. Please read them carefully!

Artificial CPU

Below is a simple circuit. It consists of a **memory** with 16 memory spaces (from M1 to M16), an **ALU block**, 2 **input registers** (IA and IB) and one **output register** (OUT). M1 to M16, IA, IB and OUT are all 8 bits long. Numbers are represented in *2's compliment* format.



Eleven commands (MOV, ADD, SUB, MUL, DIV, CMP, SHL, SHR, DEF, MSK and IF) are provided for instructing the above circuit. The syntax and the descriptions of these commands are depicted in Table 1.

Table 1: Commands for using the CPU.

Syntax	Description
MOV X Y	Copy the content of Y to X
ADD X Y	$OUT = X + Y$.
SUB X Y	$OUT = X - Y$.
MUL X Y	$OUT = X \times Y$.
DIV X Y	$OUT = X/Y$.
CMP X Y	$OUT = b_1b_2b_3b_4b_5b_6b_7b_8$. $b_i = 0$ if $X_i = Y_i$. $b_i = 1$ if $X_i \neq Y_i$.
SHL X Y	OUT is the content of X shifting left Y bits.
SHR X Y	OUT is the content of X shifting right Y bits.
DEF X N	Define X as the number N.
MSK X M	Mask the value of X by M.
IF ELSE	Condition statement.

Notes on CPU Commands

1. For the "CMP" command, if $X = 0110$ and $Y = 1101$, $OUT = 1011$.

2. For "SHL" and "SHR" commands, the content of Y can only be one of the following.

Y	Meaning
10000000	(Shift 7 bits)
01000000	(Shift 6 bits)
00100000	(Shift 5 bits)
00010000	(Shift 4 bits)
00001000	(Shift 3 bits)
00000100	(Shift 2 bits)
00000010	(Shift 1 bits)
00000001	(No shift)

   For example, if

   $$X = 00011000, Y = 00000100,$$

   the OUT of "SHL X Y" is 01100000 and the OUT of "SHR X Y" is 00000110.

3. For the "DEF" command, N must be a number in *decimal* form. X can only be a memory location. "DEF" command is not applicable for assigning values to a register. It is used to assign a value to a memory location. For example, "DEF M1 12" means that memory location $M1$ will be assigned with a value 12. Therefore, $M1 = 00001100$.

4. For the "MSK" command, it is used for masking a register (either IA or IB) by the mask M (in binary). The mask must be 8 bits long.

   Suppose that the content of IA and M are defined as follows :

   $$IA = 01001001, M = 11110000.$$

   Then, the output OUT will be "01000000". The last four bits are masked. Here is an example.

   ```
   ------------------------
   DEF M1 45
   MOV IA M1
   MSK IA 00001111
   MOV M2 OUT
   ------------------------
   ```

   Initially, $M1$ is assigned with value 45. In binary form, the content reads "00101101". Thus, the output OUT is "00001101".

5. The "IF-ELSE" command is an advanced level command. It is for conditional statement. Once it is executed, the CPU will performs multiple steps in order to make it works. You do not need to know the detail how it works. In term of its usage, it is simple. Here is an example.

   ```
   ------------------------
   DEF M1 1
   DEF M2 2
   DEF M3 1

   MOV IA M1
   IF IA == 0
      MOV IA M2
      MOV IB M3
      ADD IA IB
      MOV M4 OUT
   ELSE
      MOV IA M1
      ADD IB M2
      MOV M4 OUT
   ENDIF
   ------------------------
   ```

   Command "IF" checks if the content of IA is identical to "0". If it is, it will perform $M2+M3$ and output the result to $M4$. Otherwise, it will perform $M1+M2$ and output the result to $M4$.

   ```
   ------------------------
   DEF M1 1
   DEF M2 2
   DEF M3 1

   MOV IA M1
   IF IA == 0
      MOV IA M2
      MOV IB M3
      ADD IA IB
      MOV M4 OUT
   ENDIF
   ------------------------
   ```

   In this example, the CPU performs $M2 + M3$ only if IA is zero. Otherwise, it performs nothing.

6. For the "IF-ELSE" command, the following conditions are allowed for you to define. Here NUM must be stated in decimal form but not in binary.

```
------------------------
IA == NUM
IA > NUM
IA >= NUM
IA < NUM
IA <= NUM
------------------------
```

Program A

The filename of the following program is ”registration.cpp”.

```
#include<stdio.h>

main() /* Main function. */
{
int SNM; /* Current no. of students in Marketing. */
int SNA; /* Current no. of students in Accounting. */
char Choice; /* Define a character variable Choice. */

SNM = 45; SNA = 35;
printf("Which course you would like to register?\n");
printf("a: Marketing, b: Accounting\n");
printf("Please enter your choice (a or b): ");
scanf("%c", &Choice);

if (Choice == ’a’)
{
  SNM = SNM + 1;
  printf("You have successfully registered MKT.\n");
  printf("You are now the %d number of student.", SNM);
}
else
{
  SNA = SNA + 1;
  printf("You have successfully registered ACCT.\n");
  printf("You are now the %d number of student.", SNA);
}

}
```

Program B

The filename of the following program is ”bubblesort.cpp”.

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

int main(int argc, char *argv[])
{
    int A[24]; /* Define integer array. */
    int i,j; /* Define indices. */
    int tmp; /* Define dummy variable for sorting. */
    char Opt = ’c’; int SNUM;

    /* Step 1: Read input from the command line. */
    SNUM = argc - 2;
    Opt = argv[1][0];
    for(i = 0; i<argc-2; i++)
        A[i] = atoi(argv[i+2]);

    /* Step 2: Sorting the numbers */
    if(Opt == ’a’)
        for(i=0; i<SNUM-1; i++)
        for(j=0; j<SNUM-i-1; j++)
        {
                 if(A[j] > A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        }
    else
        for(i=0; i<SNUM-1; i++)
        for(j=0; j<SNUM-i-1; j++)
        {
                 if(A[j] < A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        }


    /* Step 3: Show the results */
    printf("The sorted numbers are");
    for(i=0; i<SNUM; i++)
        printf(" %d", A[i]);
    printf(".\n");
}
```

Program C

The filename of the following program is ”bubblesort02.cpp”.

```
#include<stdio.h>
main()
{
    int A[5]; /* Define integer array. */
    int i,j; /* Define index. */
    int tmp; /* Define dummy variable for sorting. */
    char OPT[8];  char Opt = ’c’;

    printf("This program demonstrates how to do sorting.\n\n");
    while((Opt != ’a’)&&(Opt != ’d’))
    {
    printf("Select a sorting style, ’a’ for ascending and ’d’ for descending.\n");
    printf("Option : ");
    scanf("%s", OPT);
    Opt = OPT[0];
    }
    printf("\n");
    printf("Please enter five numbers.\n");

    for(i=0; i<5; i++)
    {
    printf("Enter the %d number: ", i+1);
    scanf("%d", &A[i]);
    }

    if(Opt == ’a’)
        {
        for(i=0; i<4; i++)
        for(j=0; j<4-i; j++)
        {
                 if(A[j] > A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        }
        printf("The sorted numbers are %d %d %d %d %d.", A[0], A[1], A[2], A[3], A[4]);
        }
    else
        {
        for(i=0; i<4; i++)
        for(j=0; j<4-i; j++)
        {
                 if(A[j] < A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        }
        printf("The sorted numbers are %d %d %d %d %d.", A[0], A[1], A[2], A[3], A[4]);
        }
}
```

Program D

The filename of the following program is ”bubblesort03.cpp”.

```
#include<stdio.h>
main()
{
    int A[5], i, j, tmp;
    char OPT[8];  char Opt = 'c';

    printf("This program demonstrates how to do sorting.\n\n");
    while((Opt != 'a')&&(Opt != 'd'))
    {
    printf("Select a sorting style, 'a' for ascending and 'd' for descending.\n");
    printf("Option : ");
    scanf("%s", OPT);
    Opt = OPT[0];
    }
    printf("\n");
    printf("Please enter five numbers.\n");

    for(i=0; i<5; i++)
    {
    printf("Enter the %d number: ", i+1);
    scanf("%d", &A[i]);
    }

    if(Opt == 'a')
        for(i=0; i<4; i++)
        for(j=0; j<4-i; j++)
                 if(A[j] > A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        printf("The sorted numbers are");
        for(i=0; i<5; i++)
            printf(" %d", A[i]);
        printf(".");
    else
        for(i=0; i<4; i++)
        for(j=0; j<4-i; j++)
                 if(A[j] < A[j+1])
                 {
                 tmp = A[j];
                 A[j] = A[j+1];
                 A[j+1] = tmp;
                 }
        printf("The sorted numbers are");
        for(i=0; i<5; i++)
            printf(" %d", A[i]);
        printf(".");
}
```

Program E

The filename of the following program is ”poker01.cpp”.

```
/***********************************************
poker01.cpp
************************************************
This program demonstrates how to use rand() and
for-loop to implement card shuffling. Besides,
it demonstrates how to use switch-case to select
the card number output style.

Pseudo code
===========
Step 1: (Card Initialization)
  Initialize CARD[0]=0, CARD[1]=1 and so on.

Step 2: (Card Shuffling)
  Step 2.1: Pick up any two locations tmp1, tmp2.
  Step 2.2: Swap CARD[tmp1] and CARD[tmp2].
  Step 2.3: Allocate CARD[] to PLAYER[][].

Step 3: (Output Results)
  For the i-th player, find out the 'hand'
  (i.e. quotient) and 'number' (i.e. remainder)
  of the card number.
      HAND = PLAYER[i][] / 13;
      NUMBER = PLAYER[i][] % 13;
  Use the information 'HAND' to determine
  the output style.

Created on: December 29, 2017.
***********************************************/

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

main()
{
      int MAX = 10000;
      int RANGE = 52;
      int CARD[52];
      int PLAYER[13][4];
      int HAND, NUMBER;
      int i, j;
      int tmp, tmp1, tmp2;

      /* Step 1: Card Initialization */
      for(i=0; i<RANGE; i++)
        CARD[i] = i;

      /* Step 2: Card shuffling */
      printf("The card are being shuffled now.\n");
      for(i=0; i<MAX; i++)
      {
               tmp1 = rand() % RANGE;
               tmp2 = rand() % RANGE;
               tmp = CARD[tmp1];
               CARD[tmp1] = CARD[tmp2];
               CARD[tmp2] = tmp;
      }
      for(i=0; i<13; i++)
        for(j=0; j<4; j++)
          PLAYER[i][j] = CARD[i*4+j];

      /* Step 3: Output Results */
      printf("Here are the cards.\n");
      for(i=0; i<13; i++)
      {
          printf("Player %2d: ", i+1);
          for(j=0; j<4; j++)
            {
              tmp = PLAYER[i][j];
              HAND = tmp/13;
              NUMBER = tmp % 13;

              switch ( HAND )
              {
                     case 0: printf("S%d ", NUMBER+1); break;
                     case 1: printf("H%d ", NUMBER+1); break;
                     case 2: printf("C%d ", NUMBER+1); break;
                     case 3: printf("D%d ", NUMBER+1); break;
              }
            }
          printf("\n");
      }
      /* END */
}
```