Introduction to Computer Science: Final Exam

January 6, 2023. 9:30 - 11:00

Name: _____

Student ID: _____

Instructions (Must Read):

(1) Please fill in your name and student ID.

(2) This paper consists of 50 multiple choice questions. You need to answer all of them. Each question has only ONE correct answer and each question carries 2 marks.

(3) The questions are divided into two sections. Section A consists of 10 logical questions. Please follow the instructions described in the section to answer the questions. Section B consists of 40 questions.

(4) Please write down your answer (i.e. the option) on the left hand side close to the corresponding question number, by a blue or black ball pen. Writing the answer by pencil is accepted but not preferred.

(5) Some questions have an option '**None of the above.**'. If you have found that options (a), (b), (c) and so on are not correct, you should select this option. Let say, a question has five options and option (e) is '**None of the above.**'. You should opt (e) if you have found that options (a), (b), (c) and (d) are all incorrect.

(6) Dictionary, cell phone, computer, pad and the devices which are able to connect to the Internet are not allowed to use during the exam. If you are waiting for an urgent call, please inform Professor John Sum before the examination starts.

Please do not take away this paper. This paper has to be returned for marking.

SECTION A : Logical Questions

Instructions for Question 1 to Question 10: The questions below are logical questions. In each question, two statements X and Y are given. You have to identify from the following options what is their relation.

(a) Both statements are not true.

(b) Statement X is true. Statement Y is not true.

(c) Statement X is not true. Statement Y is true.

(d) Statement X is true. Statement Y is true. Statement X and Statement Y have no logical implication.

(e) Statement X is true. Statement Y is true. Statement X is a cause of Statement Y.

(f) Statement X is true. Statement Y is true. Statement Y is a cause of Statement X.

Question 1

With reference to the Program A in the appendix,

X: The program 'program01.cpp' can be complied successfully.

Y: There is no syntax error in 'program01.cpp'.

Question 2

With reference to the Program A in the appendix and assuming that the values input for x and y are 1 and 2 respectively,

X: The result shown for $x \times y$ is 2.

Y: $1.0_{10} \times 2.0_{10} = 2$.

Question 3

With reference to the Program A in the appendix and assuming that the values input for x and y are 1.5 and 2 respectively,

X: The result shown for $x \times y$ is 3.

Y: $1.5_{10} \times 2.0_{10} = 3$.

Question 4

With reference to the Program B in the appendix,

X: The program 'program02.cpp' will be complied successfully.

Y: There is no syntax error in 'program02.cpp'.

Question 5

Program C is a program for solving the abnormal-ball problems.

X: 'abnormalballs01.cpp' is able to be compiled successfully.

Y: 'abnormalballs01.cpp' is the only program that can solve the abnormal-ball problems.

Question 6

Program C is a program for solving the abnormal-ball problems.

X: 'abnormalballs01.cpp' is created by John Sum.

Y: John Sum is the only person who can create a program to solve the abnormal-ball problems.

Question 7

Data communication refer to transferring a piece of data from one device to another.

X: Internet technologies are developed to support data communications amongst the computers which are connected to the Internet.

Y: Telecommunication technologies are developed to support data/voice communications amongst the telephones which are connected to the telecommunication network.

Question 8

Data communication refer to transferring a piece of data from one device to another.

X: TCP/IP is a protocol (i.e. a set of technologies) developed to support secure data communication.

Y: Data communications amongst the computers connected to the Internet could now be securely protected by some encryption technologies.

Question 9

X: Telecommunication network is able to support data service.

Y: iPhone user is able to access a website if he/she has subscribed data service via a local telecommunication firm.

Question 10

X: Today, we are able to run multiple application software (equivalently, application systems) seemingly simultaneous in a computer.

Y: Today, Windows OS and MacOS are designed to handle multitasking.

Section B: Normal Questions

Question 11

Given that a binary number is represented by the following 8-bit signed magnitude fixed-point format.

```
sxxxffff
```

Here the leading bit is the sign bit. The 2^{nd} to the 4^{th} bits are for integer part. The last four bits are for the fractional part.

What is the maximum number that can be represented by the above format?

Answer:

(a) $2^7 - 1$.

(b) $1 - 2^7$.

(c) $2^3 - 2^{-4}$.

(d) $2^{-4} - 2^3$.

(e) None of the above.

Question 12

With reference to the 8-bit signed magnitude format in Question 11, what is the difference between the largest and the smallest numbers that can be represented?

Answer:

(a) $2^3 - 2^{-3}$

(b) $2^4 - 2^{-3}$.

(c) $2^4 - 2^{-4}$.

(d) $2^5 - 2^{-4}$.

(e) None of the above.

Question 13

With reference to the 8-bit signed magnitude format in Question 11, how many discrete values can be represented by this format?

Answer:

(a) 2^8.

(b) $2^8 - 1$.

(c) 2^7.

(d) $2^7 - 1$.

(e) None of the above.

Question 14

With reference to the 8-bit signed magnitude format in Question 11, which of the following is the representation of the value +35/32 if the round-to-nearest is applied as the rounding method?

Answer:

(a) 00010001.

(b) 00010010.

(c) 00010011.

(d) 00010100.

(e) None of the above.

Question 15

With reference to the 8-bit signed magnitude format in Question 11, which of the following is the representation of the value +35/32 if the round-by-chopping is applied as the rounding method?

Answer:

(a) 00010001.

(b) 00010010.

(c) 00010011.

(d) 00010100.

(e) None of the above.

Question 16

With reference to the 8-bit signed magnitude format in Question 11, what is the difference between the N^{th} ($N \leq 100$) largest number and the $(N + M)^{th}$ ($0 \leq M \leq 100$) largest number?

Answer:

(a) $M/16$.

(b) $M/32$.

(c) $N/16$.

(d) $N/32$.

(e) None of the above.

Question 17

Which of the following logic gate is able to implement any logic circuit if it is implemented solely by using such logic gates?

Answer:

(a) AND gate.

(b) XOR gate.

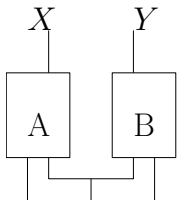
(c) OR gate.

(d) NAND gate.

(e) None of the above.

Question 18

The following circuit consists of two logic gates.



What are the output values X and Y if A is an XOR gate, B is an AND gate and the input (from left to right) is $1x1$, where x is an unknown.

Answer:

(a) $X = 0, Y = 0$.

(b) $X = 0, Y = 1$.

(c) $X = 1, Y = 0$.

(d) $X = 1, Y = 1$.

(e) None of the above.

Question 19

With reference to the logic circuit as shown in Question 18, what are the output values X and Y if A is an OR gate, B is an OR gate and the input (from left to right) is $1x1$, where x is an unknown.

Answer:

(a) $X = 0, Y = 0$.

(b) $X = 0, Y = 1$.

(c) $X = 1, Y = 0$.

(d) $X = 1, Y = 1$.

(e) None of the above.

Question 20

With reference to the logic circuit as shown in Question 18, what are the output values X and Y if A is an AND gate, B is an XOR gate and the input (from left to right) is $1x1$, where x is an unknown.

Answer:

(a) $X = 0, Y = 0$.

(b) $X = 0, Y = 1$.

(c) $X = 1, Y = 0$.

(d) $X = 1, Y = 1$.

(e) None of the above.

Question 21

With reference to the logic circuit as shown in Question 18, what are the output values X and Y if A is an OR gate, B is an NAND gate and the input (from left to right) is $1x1$, where x is an unknown.

Answer:

(a) $X = 0, Y = 0$.

(b) $X = 0, Y = 1$.

(c) $X = 1, Y = 0$.

(d) $X = 1, Y = 1$.

(e) None of the above.

Question 22

With reference to the logic circuit as shown in Question 18, what are the output values X and Y if A is an NAND gate and B is an NAND gate and the input (from left to right) is $1x1$, where x is an unknown.

Answer:

(a) $X = 0, Y = 0$.

(b) $X = 0, Y = 1$.

(c) $X = 1, Y = 0$.

(d) $X = 1, Y = 1$.

(e) None of the above.

Question 23

Which of the following factors will affect the efficiency of a CPU?

(i) The clock speed.

(ii) The design of the microprograms.

(iii) The architecture of the CPU.

Answer:

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

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

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

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

(e) None of the above.

Question 24

Which of the following factors determining the efficiency of an application software, like MS Word and DevC?

(i) The program design of the application software.

(ii) The program design of the operating system running in the computer.

(iii) The design of the microprograms in the CPU.

Answer:

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

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

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

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

(e) None of the above.

Question 25

Suppose you have already subscribed the voice and data services from a local telecommunication firm for your cell phone. Happen to be, you have encountered that your cell phone is unable to access both `yahoo.com` and `gmail.com`.

Which of the following reason(s) is(are) the possible cause(s) for this? Note that your cell phone is full of battery and its setting is under normal condition. Your cell phone is not set to WiFi ON.

(i) Both `yahoo.com` and `gmail.com` web servers are down or being attacked.

(ii) The DNS of your telecommunication network being connected is down.

(iii) Your cell phone has been set to Bluetooth ON.

Answer:

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

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

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

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

(e) None of the above.

Question 26

Which of the following OS families are commonly found in today computing devices (including computers and cell phones)?

(i) MacOS.

(ii) Windows.

(iii) Unix.

Answer:

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

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

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

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

(e) None of the above.

Question 27

Once a computer has been powered on, the first instruction the CPU to be executed is stored in which device?

Answer:

(a) RAM.

(b) Harddisk.

(c) BIOS.

(d) Cache in CPU.

(e) None of the above.

Question 28

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 29

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 30

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 31

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 32

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 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 33

Which of the following statement about TCP/IP is incorrect?

Answer:

(a) TCP/IP are two protocols (i.e. two collections of technologies) for the Internet.

(b) The purpose of TCP/IP is to ensure that a message sending from one computer (with corresponding IP address) can arrive to another computer (with corresponding IP address) without any error.

(c) The purpose of TCP/IP is to ensure that a message sending from one computer (with corresponding IP address) can arrive to another computer (with corresponding IP address) securely without any error.

(d) TCP stands for *transmission control protocol*, while IP stands for *internetworking protocol*.

Question 34

The following is a simple C program.

```
/*****************************
test01.cpp
*****************************/
#include<stdio.h>

main()
{
    int x, y, z1, z2;

    x = 7; y = 5;
    z1 = (x+y)%3;
    z2 = (x%3 + y%3);

    printf("z1 = %d and z2 = %d.", z1, z2);
}
```

Once the above program has been executed, what will you see on the command window?

Answer:

(a)
```
C:\>test01
z1 = 4 and z2 = 3.
C:\>
```

(b)
```
C:\>test01
z1 = 4 and z2 = 4.
C:\>
```

(c)
```
C:\>test01
z1 = 0 and z2 = 3.
C:\>
```

(d)
```
C:\>test01
z1 = 4 and z2 = 3.

C:\>
```

(e) None of the above.

Question 35

The following is a simple C program.

```
/***************************
test02.cpp
***************************/
#include<stdio.h>

main()
{
    int x, y, z1, z2;

    x = 7; y = 5;
    z1 = (x+y)/3;
    z2 = (x/3 + y/3);

    printf("z1 = %d and z2 = %d.", z1, z2);
}
```

Once the above program has been executed, what will you see on the command window?

Answer:

(a)
```
C:\>test02
z1 = 4 and z2 = 3.
C:\>
```

(b)
```
C:\>test02
z1 = 4 and z2 = 4.
C:\>
```

(c)
```
C:\>test02
z1 = 0 and z2 = 3.
C:\>
```

(d)
```
C:\>test02
z1 = 4 and z2 = 3.

C:\>
```

(e) None of the above.

Question 36

The following is a simple C program.

```
/***************************
test03.cpp
***************************/
#include<stdio.h>

main()
{
    int i, x = 10;

    for(i=0; i<3; i++)
    {
        printf("*\n");
        x = x - i;
    }
    printf("x is %d.\n", x);
}
```

Once the above program has been executed, what will you see on the command window?

Answer:

(a)
```
C:\>test03
***7.
C:\>
```

(b)
```
C:\>test03
***7.

C:\>
```

(c)
```
C:\>test03
*
*
*
7
C:\>
```

(d)
```
C:\>test03
*
*
*
7

C:\>
```

(e) None of the above.

Question 37

The following is a simple C program.

```
/***************************
test04.cpp
***************************/
#include<stdio.h>
#include<stdlib.h>

main(int argc, char *argv[])
{
 printf("argv[2][1] is %c.\n", argv[2][1]);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test04 john sum 001 002 003
```

Answer:

(a)
```
C:\>test04 john sum 001 002 003
argv[2][1] is j.
C:\>
```

(b)
```
C:\>test04 john sum 001 002 003
argv[2][1] is j.

C:\>
```

(c)
```
C:\>test04 john sum 001 002 003
argv[2][1] is u.
C:\>
```

(d)
```
C:\>test04 john sum 001 002 003
argv[2][1] is u.

C:\>
```

(e) None of the above.

Question 38

The following is a simple C program.

```
/*****************************
test05.cpp
*****************************/
#include<stdio.h>
#include<stdlib.h>

main(int argc, char *argv[])
{
    int i;

    while(argv[1][0] == 'j')
    {
        for(i=0; i<3; i++)
            printf("3");
    }
    printf("%d", i);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test05 john
```

Answer:

(a)
```
C:\>test05 john
3332
C:\>
```

(b)
```
C:\>test05 john
3333
C:\>
```

(c)
```
C:\>test05 john
3
3
3
2
C:\>
```

(d)
```
C:\>test05 john
3
3
3
3
C:\>
```

(e) None of the above.

Question 39

The following is a simple C program.

```
/*****************************
test06.cpp
*****************************/
#include<stdio.h>
#include<stdlib.h>

main(int argc, char *argv[])
{
    int i;

    do
    {
        for(i=0; i<3; i++)
            printf("3");
    } while(argv[1][0] == 'j');

    printf("%d", i);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test06 john
```

Answer:

(a)
```
C:\>test06 john
3332
C:\>
```

(b)
```
C:\>test06 john
3333
C:\>
```

(c)
```
C:\>test06 john
3
3
3
2
C:\>
```

(d)
```
C:\>test06 john
3
3
3
3
C:\>
```

(e) None of the above.

Question 40

The following is a simple C program.

```
/*****************************
test07.cpp
*****************************/
#include<stdio.h>

#define PI 3.14159265

main()
{
    printf("PI is %.4f.", PI);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test07
```

Answer:

(a)
```
C:\>test07
PI is 3.141.
C:\>
```

(b)
```
C:\>test07
PI is 3.1415.
C:\>
```

(c)
```
C:\>test07
PI is 3.142.
C:\>
```

(d)
```
C:\>test07
PI is 3.1416.
C:\>
```

(e) None of the above.

Question 41

The following is a simple C program.

```
/*****************************
test08.cpp
*****************************/
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

main()
{
    float x, y;

    srand(time(NULL));
    x = rand()/RAND_MAX;
    y = rand()/RAND_MAX;
    printf("x = %.2f and y = %.4f.", x, y);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test08
```

Answer:

(a)
```
C:\>test08
x = 0.00 and y = 0.0000.
C:\>
```

(b)
```
C:\>test08
x = 0.00 and y = 0.00.

C:\>
```

(c)
```
C:\>test08
x = 0.0000 and y = 0.0000.
C:\>
```

(d)
```
C:\>test08
x = 0.0000 and y = 0.00.

C:\>
```

(e) None of the above.

Question 42

The following is a simple C program.

```
/*****************************
test09.cpp
*****************************/
#include<stdio.h>
#include<stdlib.h>

main()
{
    float x, y;

    x = rand()/RAND_MAX;
    y = rand()/RAND_MAX;
    printf("x = %.2f and y = %.4f.", x, y);
}
```

Once the following command has been entered on the command prompt, what will you see?

```
C:\>test09
```

Answer:

(a)
```
C:\>test09
x = 0.00 and y = 0.0000.
C:\>
```

(b)
```
C:\>test09
x = 0.00 and y = 0.00.

C:\>
```

(c)
```
C:\>test09
x = 0.0000 and y = 0.0000.
C:\>
```

(d)
```
C:\>test09
x = 0.0000 and y = 0.00.

C:\>
```

(e) None of the above.

Question 43

Which of the following programs can be compiled successfully by a C compiler?

(i)
```
main()
{
    int i = 1, j = 2, k, z;

    z = i+j;

    return z;
}
```

(ii)
```
#include<stdio.h>

main()
{
    int i = 1, j = 2, k, z;

    z = i+j+k;
    printf("%d", z);

    return z;
}
```

(iii)
```
#include<stdio.h>

main()
{
    int i = 1;
    int j = 2;
    int k = 3;

    z = i+j+k;

    return z;
}
```

Answer:

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

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

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

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

(e) None of the above.

Question 44

Refer to the artificial CPU and its commands in the Appendix. Note that the number is represented in 8-bit 2's complement format.

```
----------------
MOV IA M1
MOV IB M2
ADD IA IB
MOV IA OUT
MOV IB M3
SUB IA IB
MOV M4 OUT
----------------
```

Suppose the initial contents of M1, M2, M3 and M4 are given as below.

```
M1 = 00000011
M2 = 00000010
M3 = 00001000
M4 = 00000000
```

What is the content of M4 once the program is finished?

Answer:

(a) 10000010.

(b) 10000011.

(c) 11111101.

(d) 11111100.

Question 45

Refer to the artificial CPU and its commands in the Appendix. Which one of the following programs performs the function equivalent to that

$$M3 = M1 - M2.$$

Answer:

(a)
```
----------------
MOV IA M2
MOV IB M1
SUB IA IB
MOV IA OUT
----------------
```

(b)
```
----------------
MOV IA M1
MOV IB M2
SUB IB IA
MOV IA OUT
----------------
```

(c)
```
----------------
MOV IA M2
MOV IB M1
SUB IB IA
MOV M3 OUT
----------------
```

(d)
```
----------------
MOV IA M1
MOV IB M2
SUB IB IA
MOV M3 OUT
----------------
```

(e) None of the above.

Question 46

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
SHL IA 00000010
MOV IA OUT
SHL IB 00000100
MOV IB OUT
ADD IA IB
MOV M5 OUT
MOV IA M3
MOV IB M4
SHL IA 00000100
MOV IA OUT
ADD 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?

Answer:

(a) $M5 = 2 \times (M1 + M2 + M3) + M4$.

(b) $M5 = 2 \times (M1 + 2 \times M2 + M3) + M4$

(c) $M5 = 2(\times M1 + 2 \times (M2 + M3)) + M4$.

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

(e) None of the above.

Question 47

With reference to the simple processor as shown in the Appendix, Figure 1, suppose that the registers are preset as $RA = 1$, $RB = 1$, $RZ = 0$, $R1 = R2 = R3 = R4 = 0$. What will be the contents of the registers RA and RB after the following micro-instructions (S1, S2, S3 and S4) have been executed?

S1: $S_1 = S_8 = 1$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S2: $S_{14} = 01$. $S_{12} = S_{15} = 10$. The control signals to all connectors are set to 0. The signals to other two-way switches are set to 00.

S3: $S_2 = S_5 = S_9 = 1$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

S4: $S_{12} = 10$. $S_{14} = 01$. $S_{16} = 10$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

Answer:

(a) $RA = 0, RB = 0$.

(b) $RA = 0, RB = 1$.

(c) $RA = 1, RB = 0$.

(d) $RA = 1, RB = 1$.

Question 48

With reference to the simple processor as shown in the Appendix, Figure 1, suppose that the registers are preset as $RA = 1$, $RB = 0$, $RZ = 0$, $R1 = R2 = R3 = R4 = 0$. What will be the contents of the registers $R1$ and $R2$ after the following micro-instructions (S1, S2, S3 and S4) have been executed?

S1: $S_1 = S_8 = 1$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S2: $S_{14} = 01$. $S_{12} = S_{15} = 10$. The control signals to all connectors are set to 0. The signals to other two-way switches are set to 00.

S3: $S_2 = S_5 = S_9 = 1$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

S4: $S_{12} = 10$. $S_{14} = 01$. $S_{16} = 10$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

Answer :

(a) $R1 = 0, R2 = 0$.

(b) $R1 = 0, R2 = 1$.

(c) $R1 = 1, R2 = 0$.

(d) $R1 = 1, R2 = 1$.

Question 49

With reference to the simple processor as shown in the Appendix, Figure 1, suppose that the registers are preset as $RA = 0$, $RB = 1$, $RZ = 0$, $R1 = R2 = R3 = R4 = 0$. What will be the contents of the registers $R3$ and $R4$ after the following micro-instructions (S1, S2, S3 and S4) have been executed?

S1: $S_1 = S_8 = 1$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S2: $S_{14} = 01$. $S_{12} = S_{15} = 10$. The control signals to all connectors are set to 0. The signals to other two-way switches are set to 00.

S3: $S_2 = S_5 = S_9 = 1$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

S4: $S_{12} = 10$. $S_{14} = 01$. $S_{16} = 10$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

Answer :

(a) $R3 = 0, R4 = 0$.

(b) $R3 = 0, R4 = 1$.

(c) $R3 = 1, R4 = 0$.

(d) $R3 = 1, R4 = 1$.

Question 50

With reference to the simple processor as shown in the Appendix, Figure 1, it is assumed that the data in the memory locations $M1$ and $M2$ have already been copied to the registers RA and RB. What will be the logical operation once the following micro-instructions have been executed?

S1: $S_{12} = 01$ and $S_{15} = 10$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S2: $S_{12} = 10$ and $S_{13} = 01$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S3: $S_1 = S_8 = 1$. The control signals to other connectors are set to 0. The signals to all two-way switches are set to 00.

S4: $S_{12} = 10$, $S_{14} = 01$ and $S_{16} = 10$. The control signals to all connectors are set to 0. The signals to other two-way switches are set to 00.

S5: $S_{13} = 10$ and $S_{15} = 01$. The control signals to all connectors are set to 0. The signals to other two-way switches are set to 00.

S6: $S_2 = S_5 = S_9 = 1$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

S7: $S_{12} = 10$, $S_{14} = 01$ and $S_{15} = 10$. The control signals to other connectors are set to 0. The signals to other two-way switches are set to 00.

What of the following instruction have been executed?

Answer :

(a) $R1 = (\neg RA)RB$ and $R2 = \neg RA$.

(b) $R1 = RA(\neg RB)$ and $R2 = \neg RA$.

(c) $R1 = (\neg RA)RB$ and $R2 = \neg RB$.

(d) $R1 = RA(\neg RB)$ and $R2 = \neg RB$.

(e) None of the above.

APPENDIX

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

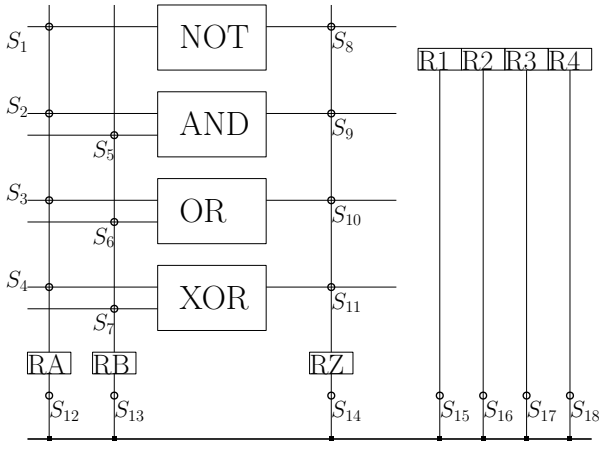
A. Processor with 4 Logic Gates

A simple processor, with a sector of four logic gates and a sector of four registers, shown in Figure 1. Each register is associated with a two-way switch. The signals to be fed to the switch and the corresponding actions are depicted in the following table.

S_i	Action
00	Disconnect.
01	Read from register.
10	Write to register.

For each connector, its control signal is either '0' (for disconnection) and '1' (for connection).

$$\text{Connection} = \begin{cases} \text{Connect} & \text{if } S_i = 1, \\ \text{Disconnect} & \text{if } S_i = 0. \end{cases}$$



Two-Way Switches: Disconnected (00); Down (01), Up (10).

$(S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_{17}, S_{18})$

Connectors: Disconnected (0), Connected (1).

$(S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11})$

Figure 1: A processor with four logic gates. Switches S_1 to S_{11} are simple switches (i.e. connectors). S_{12} to S_{18} are two-way switches.

B. Processor with a NAND Gate

Figure 2 shows a simple processor with single NAND gate inside. Switches S_1, S_2 and S_3 are simple switches (i.e. connectors). S_{12}, S_{13} and S_{14} are two-way switches. The signals sending to $A1$, $A2$ and R/W together with the corresponding actions are depicted in the following table.

$A1$	$A2$	R/W	Action
0	0	01	Read data from $R1$
0	1	01	Read data from $R2$
1	0	01	Read data from $R3$
1	1	01	Read data from $R4$
0	0	10	Write data to $R1$
0	1	10	Write data to $R2$
1	0	10	Write data to $R3$
1	1	10	Write data to $R4$
x	x	00	Disconnection

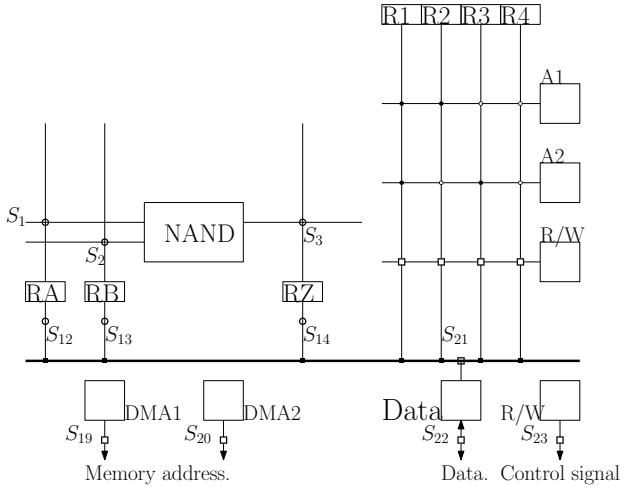
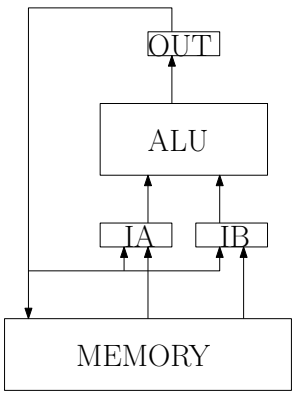


Figure 2: A processor with single NAND gate.

C. 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.

D. 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
------------------------
```

E. Program A

```
/************************
program01.cpp
************************/
#include<stdio.h>

main()
{
    int x,y;

    printf("x value: ");
    scanf("%d", &x);
    printf("y value: ");
    scanf("%d", &y);

    printf("x+y is %d.\n", x+y);
    printf("x-y is %d.\n", x-y);
    printf("x*y is %d.", x*y);
}
```

F. Program B

```
/************************
program02.cpp
************************/
#include<stdio.h>

main()
{
    int x,y;

    printf("x value: ");
    scanf("%d", &x);
    printf("y value: ");
    scanf("%d", &y);

    for(x = 0; x < 2; x++)
    {
        printf("x+y is %d.\n", x+y);
        printf("x-y is %d.\n", x-y);
        printf("x*y is %d.", x*y);
    }
}
```

G. Program C

```
/*********************************************************************
abnormalballs01.cpp

   Description and usage:
   c:\>abnormalballs01 200 200 200 199 200 201 200

   The number of balls should be abitrary.

   The program will show the following.
   ----------------------------------------
   Ball 0 is a normal ball.
   Ball 1 is a normal ball.
   Ball 2 is a normal ball.
   Ball 3 is a lighter ball.
   Ball 4 is a normal ball.
   Ball 5 is a heavier ball.
   Ball 6 is a normal ball.
   ----------------------------------------

   History:
   December 22, 2022. Created by John Sum
*********************************************************************/

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

main(int argc, char *argv[])
{
    int A[256]; /* Define integer array. */
    int B199 = 0, B200 = 0, B201 = 0; /* Number of balls with weight 199, 200 and 201. */
    int C[256]; /* Integer array for pairwise comparison. */
    int i,j, itmp; /* Define indices. */
    int H, E, L, PE = 0, NB, RB, NAB;

    /* Step 1: Read input from the command line. */
    NB = argc - 1;

    for(i=0; i<NB; i++)
    {
        A[i] = atoi(argv[i+1]);
        if (A[i] == 199) B199 = B199 + 1;
        if (A[i] == 200) B200 = B200 + 1;
        if (A[i] == 201) B201 = B201 + 1;
    }
    NAB = NB - B200;

    /* Step 2: Finding the abnormal balls. */
    if(NAB < B200)
    {
        itmp = 1; RB = NB;
        while((PE == 0)&&(itmp < NB))
        {
            H = 0; E = 0; L = 0;
            for(i=itmp; i<NB; i++)
            {
                if(A[itmp-1] > A[i])  { C[i] = 1;  H = H + 1;  }
                if(A[itmp-1] == A[i]) { C[i] = 0;  E = E + 1;  }
```

```
            if(A[itmp-1] < A[i])  { C[i] = -1; L = L + 1; }
        }

        if (E >= RB/2)
        {
            printf("Ball %d is a normal ball.\n", itmp-1);
            for (j=itmp; j<NB; j++)
            {
                if(C[j] == 1)  printf("Ball %d is a lighter ball.\n", j);
                if(C[j] == 0)  printf("Ball %d is a normal ball.\n", j);
                if(C[j] == -1) printf("Ball %d is a heavier ball.\n", j);
            }
            PE = 1;
        }
        if (PE == 0)
        {
            if (L == 0) printf("Ball %d is a heavier ball.\n", itmp-1);
            if (H == 0) printf("Ball %d is a lighter ball.\n", itmp-1);
        }
        RB = RB - 1;
        itmp = itmp + 1;
    }
  }
  else
  {
      printf("Total number of abnormal balls is larger than normal balls.\n");
      printf("Rerun the program by entering a new set of weights.");
  }
}
```