

- REPORT -Name: **Đào Minh Châu**ID: **24560025****1. Classwork: Answer all question in the instruction.****Question 1.1.****The program read a number from user and then print it on the screen.****li \$v0, 5 :** load the immediate 5(the service number to read an integer) to register \$v0.**Syscall:** tell the computer to get number from user.**add \$a0, \$v0, 0:** set value of \$a0 equal to value of \$v0.**li \$v0, 1:** load immediate 1(the service number to print an integer) to register \$v0.**syscall:** tell the computer to print the number.**Question 1.2.**

	H	e	l	l	o
ASCII value(hex)	0x48	0x65	0x6C	0x6C	0x6F

Adress	Value(+0)				Value(+4)			
0x10010000	+3	+2	+1	+0	+7	+6	+5	+4
	l	l	e	H				o
	0x6C	0x6C	0x65	0x48				0x6F

Question 1.3.

The directive `.space` is used to declare the space of the answer.

Question 1.4.

The service `8` is used to read a string.

Question 1.5.

Input data is stored at address `0x10010000` and is next to the “What’s your name: \n” and is at `0x10010000 +16` or more depending on your input.

Question 2.1.

There are 5 labels:
+Start
+Child
+Teen
+Adult
+End

Question 2.2.

Register `$v0` is used to store input data.

Question 2.3.

The instruction `bltz` is used to branch if the value of the chosen register is less than 0.

The value of `$t1` must be less than 0 in order to jump to child.

The value of `$t1` plays a role to compare the value of `$v0` and 15 since it is the result of the subtraction between `$t0` value and 15.

Question 2.4.

The instruction `bgtz` is used to branch if the value of the chosen register is greater than 0.

The value of `$t1` must be greater than 0 in order to jump to `adult`.

`Value($t1)` plays a role to compare `value($v0)` and 17 since it is the result of the subtraction between `value($t0)` and 17.

Question 2.5.

The program jumps to `teen` only when the previous conditions are not satisfied, the program will jump straight to label `teen`.

Question 2.6.

We need to put a `j end` at the end of each label to prevent the program from executing other labels after getting the final conclusion.

2.Exercise

2.1 Given the six first notes of the Happy Birthday song below, read service 31, 33 in MARS syscall document, write a program to play the song (6 first notes)?

Happy Birthday: C C D C F E

MIDI out	31	<code>\$a0</code> = pitch (0-127) <code>\$a1</code> = duration in milliseconds <code>\$a2</code> = instrument (0-127) <code>\$a3</code> = volume (0-127)	Generate tone and return immediately. <i>See note below table</i>
sleep	32	<code>\$a0</code> = the length of time to sleep in milliseconds.	Causes the MARS Java thread to sleep for (at least) the specified number of milliseconds. This timing will not be precise, as the Java implementation will add some overhead.

.data

text: .ascii "sieu nhac pham"

.text

C1:

li \$v0, 33

li \$a0, 60

li \$a1, 1000

li \$a2, 0

li \$a3, 100

syscall

S1:

li \$v0, 32

li \$a0, 50

syscall

C2:

li \$v0, 33

li \$a0, 60

li \$a1, 1000

li \$a2, 0

li \$a3, 100

syscall

S2:

li \$v0, 32

li \$a0, 50

syscall

D:

li \$v0, 33

li \$a0, 62

li \$a1, 1000

li \$a2, 0

li \$a3, 100

syscall

S3:

li \$v0, 32

li \$a0, 50

syscall

C3:

li \$v0, 33

li \$a0, 60

li \$a1, 1000

li \$a2, 0

li \$a3, 100

syscall

S4:

li \$v0, 32

li \$a0, 50

syscall

F:

li \$v0, 33

li \$a0, 65

li \$a1, 1000

li \$a2, 0

li \$a3, 100

syscall

S10:

li \$v0, 32

li \$a0, 50

syscall

E:

li \$v0, 33

li \$a0, 64

li \$a1, 2000

li \$a2, 0

li \$a3, 100

syscall

2.2 Write a program which satisfies below requirements?

- Ask user to input two number A and B
- If $A < B$, ask user to re-input A and B
- If $A \geq B$, calculate $A - B$ and print to the screen

.data

A: .asciiz "Enter number A: \n"

B: .asciiz "Enter number B: \n"

result: .asciiz "A-B="

.text

start:

li \$v0, 4

la \$a0, A

syscall

li \$v0, 5

syscall

add \$t0, \$v0, \$zero

li \$v0, 4

la \$a0, B

syscall

li \$v0, 5

syscall

add \$t1, \$v0, \$zero

sub \$t2, \$t0, \$t1

bltz \$t2, start

li \$v0, 4

la \$a0, result

syscall

li \$v0, 1

la \$a0, (\$t2)

syscall