Report of lab01 109550027 紀竺均

Part 1

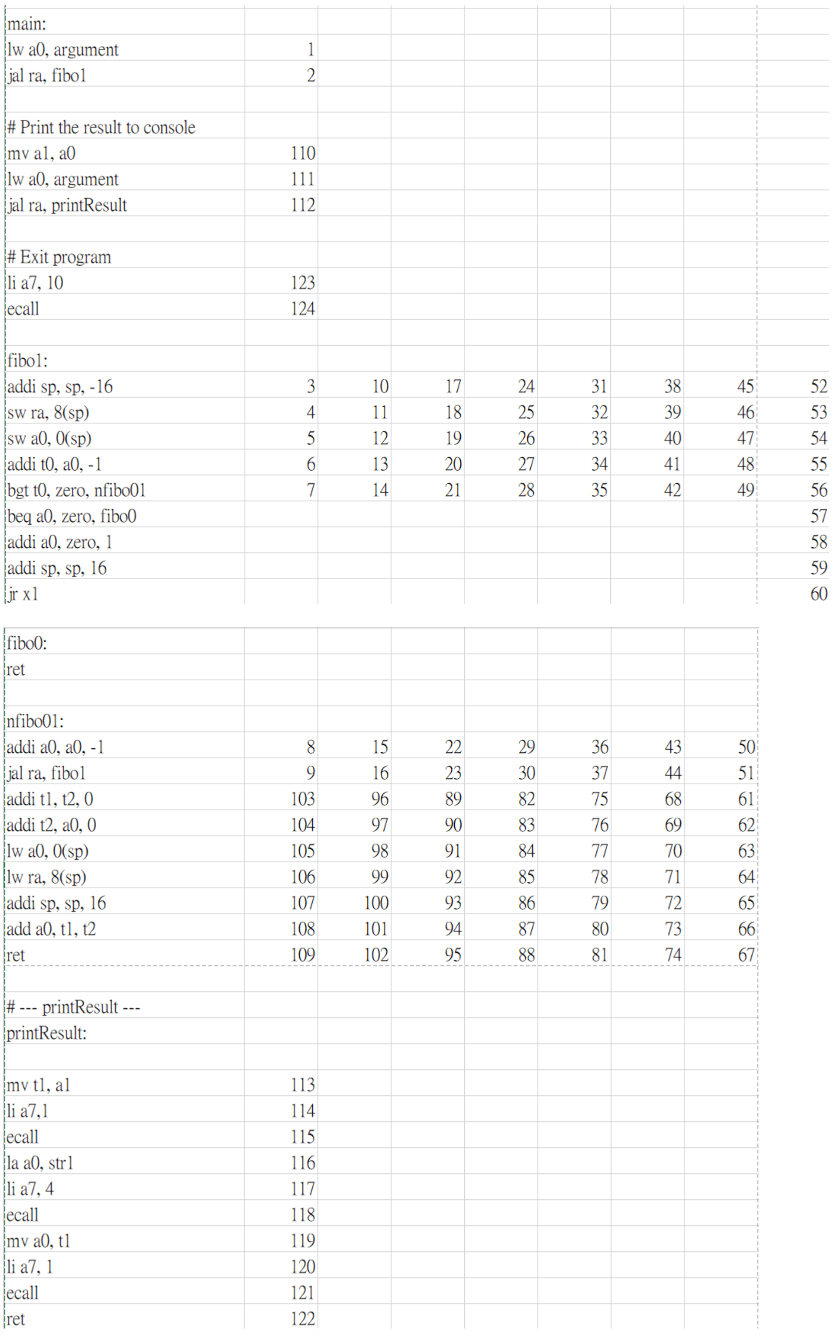
Q1: How many instructions are actually executed? You have to explain clearly how you calculate your instructions. There is no specific answer.

Q2: What is the maximum number of variable be pushed into the stack at the same time when your code execute? There is only one correct answer.

1. fibonacci:

   instruction executed: 124

   maximum number of variable be pushed into the stack: 16

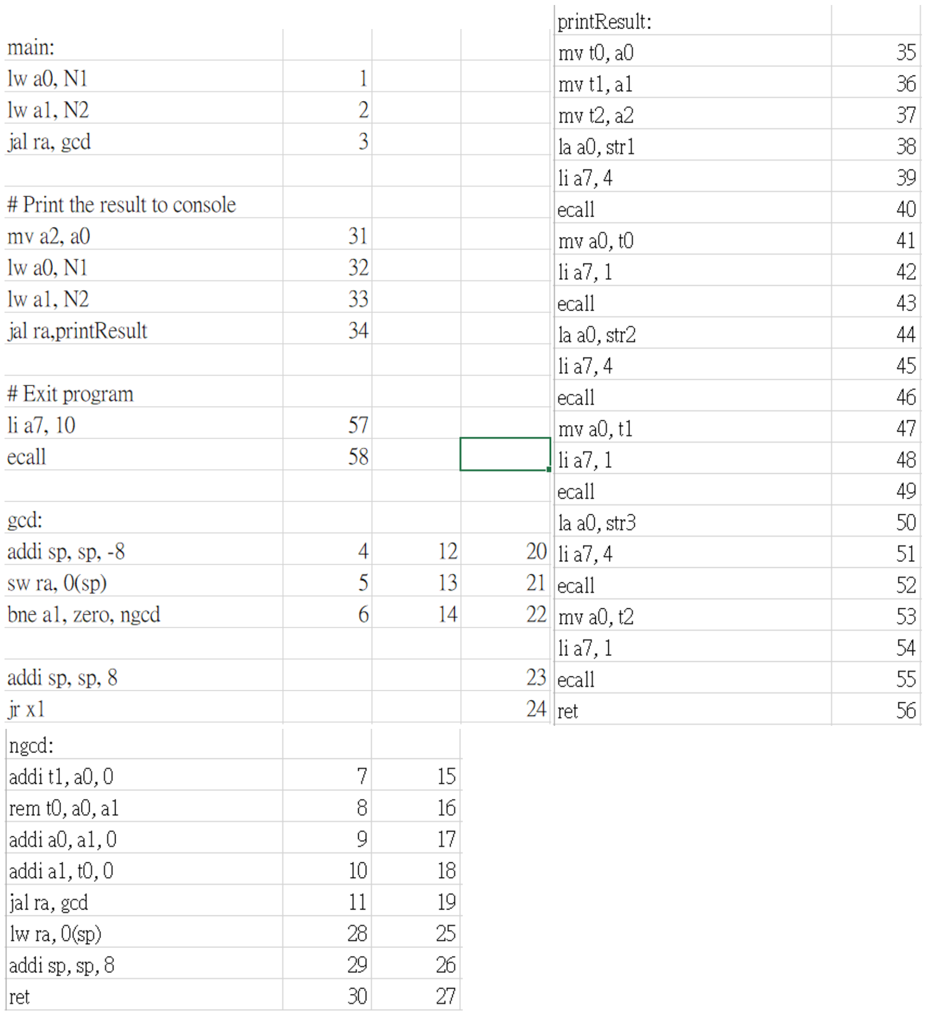


main:								
lw a0, argument	1							
jal ra, fibo1	2							
# Print the result to console								
mv a1, a0	110							
lw a0, argument	111							
jal ra, printResult	112							
# Exit program								
li a7, 10	123							
ecall	124							
fibo1:								
addi sp, sp, -16	3	10	17	24	31	38	45	52
sw ra, 8(sp)	4	11	18	25	32	39	46	53
sw a0, 0(sp)	5	12	19	26	33	40	47	54
addi t0, a0, -1	6	13	20	27	34	41	48	55
bgt t0, zero, nfibo01	7	14	21	28	35	42	49	56
beq a0, zero, fibo0								57
addi a0, zero, 1								58
addi sp, sp, 16								59
jr x1								60
fibo0:								
ret								
nfibo01:								
addi a0, a0, -1	8	15	22	29	36	43	50	
jal ra, fibo1	9	16	23	30	37	44	51	
addi t1, t2, 0	103	96	89	82	75	68	61	
addi t2, a0, 0	104	97	90	83	76	69	62	
lw a0, 0(sp)	105	98	91	84	77	70	63	
lw ra, 8(sp)	106	99	92	85	78	71	64	
addi sp, sp, 16	107	100	93	86	79	72	65	
add a0, t1, t2	108	101	94	87	80	73	66	
ret	109	102	95	88	81	74	67	
# --- printResult ---								
printResult:								
mv t1, a1	113							
li a7,1	114							
ecall	115							
la a0, str1	116							
li a7, 4	117							
ecall	118							
mv a0, t1	119							
li a7, 1	120							
ecall	121							
ret	122							

2. gcd:

   instruction executed: 58

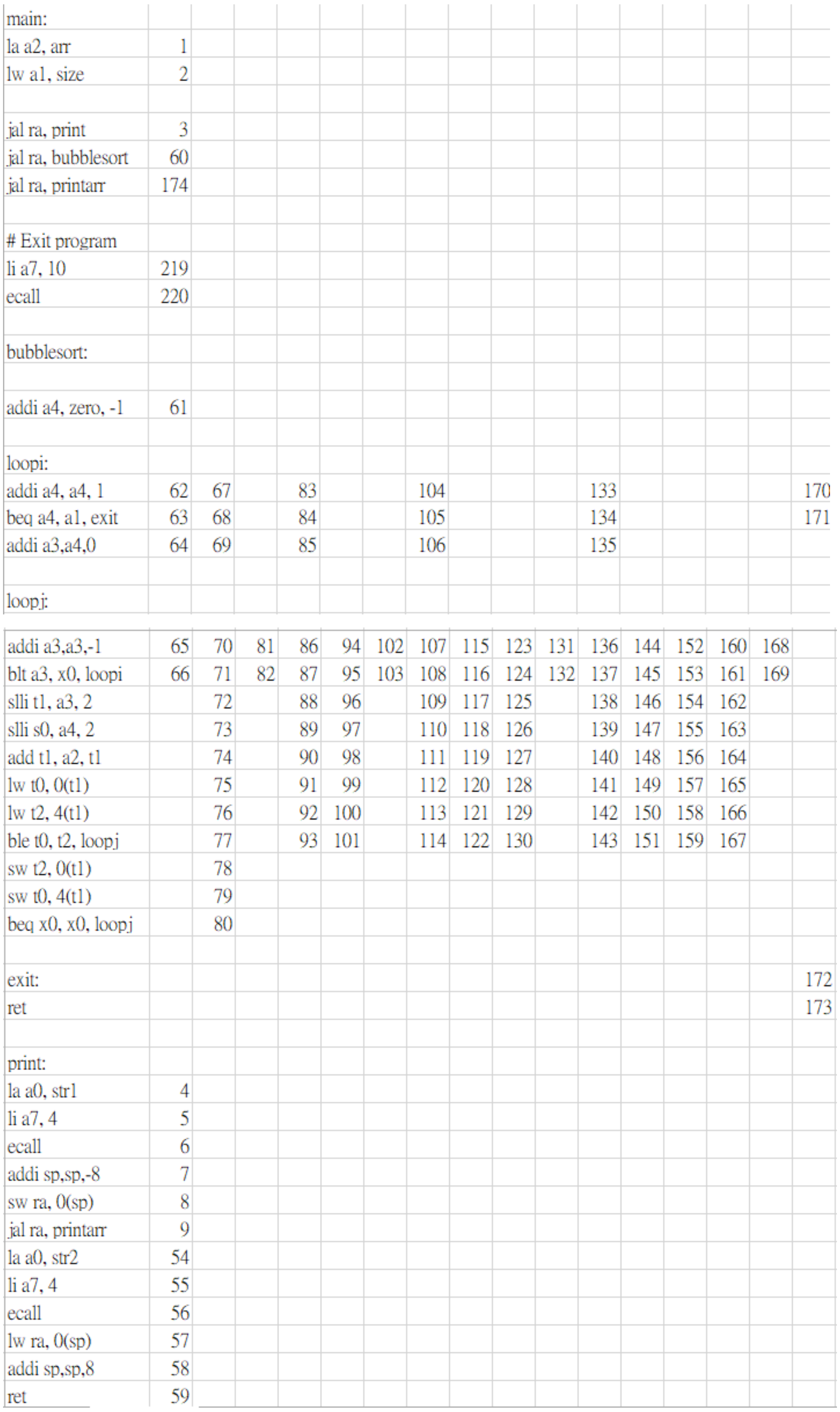
   maximum number of variable be pushed into the stack: 3



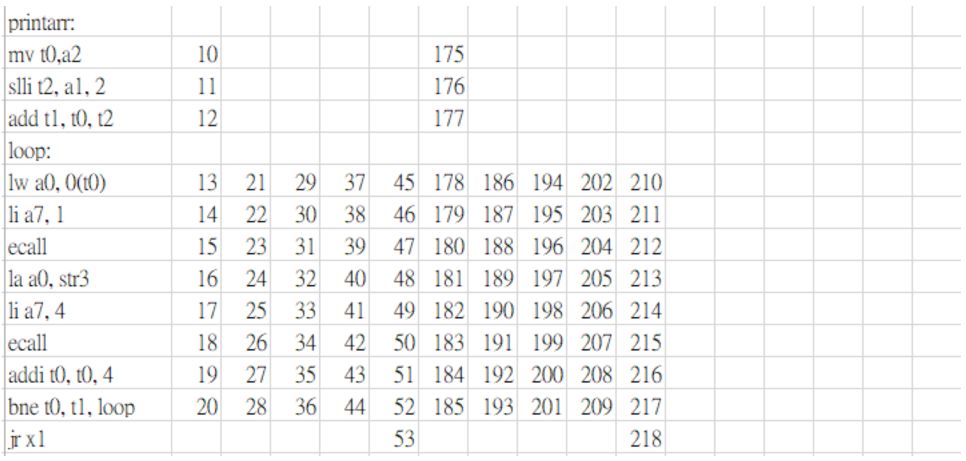
main:			
lw a0, N1	1		
lw a1, N2	2		
jal ra, gcd	3		
# Print the result to console			
mv a2, a0	31		
lw a0, N1	32		
lw a1, N2	33		
jal ra,printResult	34		
# Exit program			
li a7, 10	57		
ecall	58		
gcd:			
addi sp, sp, -8	4	12	20
sw ra, 0(sp)	5	13	21
bne a1, zero, ngcd	6	14	22
addi sp, sp, 8			23
jr x1			24
ngcd:			
addi t1, a0, 0	7	15	
rem t0, a0, a1	8	16	
addi a0, a1, 0	9	17	
addi a1, t0, 0	10	18	
jal ra, gcd	11	19	
lw ra, 0(sp)	28	25	
addi sp, sp, 8	29	26	
ret	30	27	

printResult:	
mv t0, a0	35
mv t1, a1	36
mv t2, a2	37
la a0, str1	38
li a7, 4	39
ecall	40
mv a0, t0	41
li a7, 1	42
ecall	43
la a0, str2	44
li a7, 4	45
ecall	46
mv a0, t1	47
li a7, 1	48
ecall	49
la a0, str3	50
li a7, 4	51
ecall	52
mv a0, t2	53
li a7, 1	54
ecall	55
ret	56

3. gcd: for array={5,3,6,7,31}, size = 5
   instruction executed: 220
   maximum number of variable be pushed into the stack: 1 (ra)



main:																
la a2, arr	1															
lw a1, size	2															
jal ra, print	3															
jal ra, bubblesort	60															
jal ra, printarr	174															
# Exit program																
li a7, 10	219															
ecall	220															
bubblesort:																
addi a4, zero, -1	61															
loopi:																
addi a4, a4, 1	62	67		83			104				133					170
beq a4, a1, exit	63	68		84			105				134					171
addi a3,a4,0	64	69		85			106				135					
loopj:																
addi a3,a3,-1	65	70	81	86	94	102	107	115	123	131	136	144	152	160	168	
blt a3, x0, loopi	66	71	82	87	95	103	108	116	124	132	137	145	153	161	169	
slli t1, a3, 2		72		88	96		109	117	125		138	146	154	162		
slli s0, a4, 2		73		89	97		110	118	126		139	147	155	163		
add t1, a2, t1		74		90	98		111	119	127		140	148	156	164		
lw t0, 0(t1)		75		91	99		112	120	128		141	149	157	165		
lw t2, 4(t1)		76		92	100		113	121	129		142	150	158	166		
ble t0, t2, loopj		77		93	101		114	122	130		143	151	159	167		
sw t2, 0(t1)		78														
sw t0, 4(t1)		79														
beq x0, x0, loopj		80														
exit:																172
ret																173
print:																
la a0, str1	4															
li a7, 4	5															
ecall	6															
addi sp,sp,-8	7															
sw ra, 0(sp)	8															
jal ra, printarr	9															
la a0, str2	54															
li a7, 4	55															
ecall	56															
lw ra, 0(sp)	57															
addi sp,sp,8	58															
ret	59															



printarr:										
mv t0,a2	10					175				
slli t2, a1, 2	11					176				
add t1, t0, t2	12					177				
loop:										
lw a0, 0(t0)	13	21	29	37	45	178	186	194	202	210
li a7, 1	14	22	30	38	46	179	187	195	203	211
ecall	15	23	31	39	47	180	188	196	204	212
la a0, str3	16	24	32	40	48	181	189	197	205	213
li a7, 4	17	25	33	41	49	182	190	198	206	214
ecall	18	26	34	42	50	183	191	199	207	215
addi t0, t0, 4	19	27	35	43	51	184	192	200	208	216
bne t0, t1, loop	20	28	36	44	52	185	193	201	209	217
jr x1					53					218

Part 2 experience

一開始在理解 factorial.s 的時候花了最多時間，也不太能理解 sp 一直加加減減是什麼意思，後來經過同學的指點終於豁然開朗，我感覺我和 risc v 變成好朋友一般，剩下的題目也可以漸漸地融會貫通。
遇到的挫折：我以為 run 過就會自動存檔，結果沒有，但我還是一直忘記，有時候不小心讓他出現無限迴圈，整個當掉，我就要全部重打，好慘。
心情總結：覺得這個 lab 只有一個禮拜好少喔，因為光理解就花了很多時間，網路上也找不太到相關的教學，很多都是亂推測的，然後還有一大堆奇奇怪怪的版本都不能用，但最後還是很幸運的親手把它刻出來了，也慢慢的發現有些寫法可以讓他的 instruction 少跑幾個，或許對於以後寫 code 有幫助！