

BÀI BÁO CÁO KIỂM TRA GIỮ KỲ

Họ và tên: Nguyễn Trọng Khánh Duy

MSSV: 20210284

Assignment 1: A_1

1. Phân tích cách thực hiện :

- Bước 1: Yêu cầu người dùng nhập vào số nguyên dương N, Kiểm tra điều kiện nhập xem có đúng là số nguyên dương hay không. Nếu không đúng thì báo lỗi vào yêu cầu nhập lại.
- Bước 2: Khởi tạo biến n chạy từ 1 đến N-1
- Bước 3: Kiểm tra n lần lượt chia hết cho 3 và chia hết cho 5. Nếu 1 trong 2 trường hợp thỏa mã thì in n ra màn hình.

2. Ý nghĩa của chương trình con nếu có (Không có):

3. Ảnh chụp kết quả thực hiện:

Edit Execute
Registers Coproc 1 Coproc 2

Text Segment										Labels	
Bkpt	Address	Code	Basic	Source						Label	Address
	0x00400000	0x24100001	addiu \$t6,\$0,3	7:	11	\$s0, 1	# n = 1	Nhap_n	0x0040000c		
	0x00400004	0x24120003	addiu \$t6,\$0,3	8:	11	\$s2, 3	# \$s2 = 3	Print1	0x00400044		
	0x00400008	0x24130005	addiu \$t9,\$0,5	9:	11	\$s3, 5	# \$s3 = 5	Loop	0x00400054		
	0x0040000c	0x24020004	addiu \$t2,\$0,4	11:	Nhap_n: 11	\$v0, 4	# Thong B...	Chia_3	0x00400060		
	0x00400010	lui \$t1,\$1,4097	12:	1a	\$a0, message1			Chia_5	0x0040006c		
	0x00400014	0x34240000	ori \$4,\$t1,0					Print2	0x00400090		
	0x00400018	0x0000000c	syscall	13:				Exit	0x004000a4		
	0x0040001c	0x24020005	addiu \$t2,\$0,5	15:	11	\$v0, 5	# Nhap so...	message1	0x10010000		
	0x00400020	0x0000000c	syscall	16:				message2	0x10010014		
	0x00400024	0x00028821	addiu \$t7,\$0,\$2	18:		move \$s1, \$v0	# \$s1 = N	message3	0x1001001e		
	0x00400028	0x2a2a0001	slli \$t0,\$t7,1	20:	slli	\$t2, \$s1, 1	# if (n < ...)	message4	0x10010057		
	0x0040002c	0x11400005	bneq \$t0,\$0,5	21:	bneq	\$t2, \$zero, Print1					
	0x00400030	0x24020004	addiu \$t2,\$0,4	22:	11	\$v0, 4	# Bao loi...				

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	1346455630	542069536	1498761038	1310740037	2112032	1380253728	978472786	544166688
0x10010020	1885431918	1868658208	1869114144	1881171822	543777128	1847619443	1702458727	1969496174
0x10010040	174550639	543782230	1735290732	1634233888	1634476144	1124076137	1931502433	1752440943
0x10010060	1830838639	975203937	32	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0
0x10010100	0	0	0	0	0	0	0	0
0x10010120	0	0	0	0	0	0	0	0
0x10010140	0	0	0	0	0	0	0	0

☒ Data ☒ Text

0x10010000 (.data)

☒ Hexadecimal Addresses
 ☐ Hexadecimal Values
 ☐ ASCII

Mars Messages Run I/O

```

NHAP SO NGUYEN N : 26
Cac so thoa man : 3 5 6 9 10 12 15 18 20 21 24 25
-- program is finished running --
          
```

Assignment 2: B_8

1. Phân tích cách thực hiện :

- Bước 1: Yêu cầu người dùng nhập vào số phần tử của mảng bằng N. Kiểm tra xem N có phải số âm hay không. Nếu là số âm thì báo lỗi và yêu cầu người dùng nhập lại.
- Bước 2: Sau đó người dùng nhập lần lượt từng giá trị vào mảng
- Bước 3: Khởi tạo 2 biến tong_chan = 0 và tong_le = 0 để tính tổng các số chẵn, lẻ trong mảng
- Bước 4: Khởi tạo i = 0 và thực hiện vòng lặp đến N-1. Với mỗi giá trị A[i] ta kiểm tra xem A[i] là chẵn hay lẻ.
- Bước 5: Nếu A[i] là chẵn thì cộng vào tong_chan hoặc nếu A[i] là lẻ thì cộng vào tong_le
- Bước 6: In kết quả tổng chẵn và tổng lẻ ra màn hình

2. Ý nghĩa của chương trình con nếu có (Không có):

3. Ảnh chụp kết quả thực hiện:

The screenshot displays a MIPS simulator interface with three main panels:

- Text Segment:** Shows assembly code with addresses, codes, basic instructions, and source comments. Key instructions include `lui $t1, 4097`, `ori $t7, $t1, 0`, `addiu $t1, $t0, 2`, `addiu $t2, $t0, 4`, `ori $t1, $t1, 4097`, `ori $t4, $a0, message1`, `syscall`, `addiu $t2, $t0, 5`, `syscall`, `move $a0, $v0`, `li $t0, 0`, `sllt $t5, $a0, 0`, and `beq $t5, $zero, Nhap_a`.
- Data Segment:** A table showing memory addresses and their corresponding values. The values are mostly 0, with some non-zero values at higher addresses (e.g., 268500992, 2147479548, 4194588).
- Registers:** A table showing the state of MIPS registers. The `$a0` register contains the value 4, which corresponds to the input N. Other registers like `$t0`, `$t1`, `$t2`, `$t3`, `$t4`, `$t5`, `$t6`, `$t7`, `$s0`, `$s1`, `$s2`, `$s3`, `$s4`, `$s5`, `$s6`, `$s7`, `$s8`, `$s9`, `$s10`, `$s11`, `$gp`, `$sp`, `$fp`, `$ra`, `$pc`, `$hi`, and `$lo` are also shown.

At the bottom, the **Mars Messages** panel shows the output of the program:

```
Nhap so phan tu cho mang : 4
Nhap phan tu thu 0 : 1
Nhap phan tu thu 1 : 2
Nhap phan tu thu 2 : 3
Nhap phan tu thu 3 : 4
Tong phan tu le trong mang la : 4
Tong phan tu chan trong mang la : 6
```

Assignment 3: C_4

1. Phân tích cách thực hiện :

- Bước 1: Yêu cầu người dùng nhập vào một ký tự, nếu là xâu rỗng hoặc chỉ chứa ký tự cách thì thông báo lỗi và yêu cầu người dùng nhập lại.
- Bước 2: Tìm độ dài dài nhất của từ trong xâu.
- Bước 3: Lần lượt đếm độ dài của từng chữ trong xâu và kiểm tra với độ dài lớn nhất tìm được ở bước 2. Nếu bằng thì in ra từ đó ra màn hình.

2. Ý nghĩa của chương trình con nếu có (Không có):

3. Ảnh chụp kết quả thực hiện:

The screenshot displays a debugger window with several panels:

- Text Segment:** Shows assembly instructions with addresses, codes, and comments. For example, at address 0x00400000, there is an instruction `addiu $9,$0,0` with comment `9: li $t1, 0 # Khoi t...`.
- Data Segment:** Shows memory addresses and their corresponding values. For example, at address 0x10010000, the value is 1752066413.
- Registers:** A table showing the state of various registers. The 'Registers' tab is selected, showing registers like \$zero, \$at, \$v0, \$v1, etc., with their current values.
- Mars Messages:** A text area showing the output of the program. It displays the input string "Nhap xau ky tu: minh la hoc sinh bach khoa" and the output "Ky tu dai nhat trong chuoai vua nhap la: 'minh', 'sinh', 'bach', 'khoa'".