PUSH AND POP WITH STACK

CODE

```
.data
    array: .word 1, 2, 3, 4, 5    # Initialize original array with 5 elements
    array_len: .word 5            # Length of the array
    new_array: .space 20          # Space for new array to store popped elements

.text
    .globl main

main:
    # Load the address of the original array and its length
    la $t0, array
    lw $t1, array_len

    # Loop to push each array element onto the stack
    li $t2, 0  # Initialize loop counter
push_loop:
    beq $t2, $t1, end_push_loop  # If loop counter equals array length, exit loop
    lw $t3, 0($t0)  # Load array element into $t3

    # Push $t3 onto the stack
    subu $sp, $sp, 4  # Decrease stack pointer
    sw $t3, 0($sp)    # Store $t3 value onto stack

    addi $t0, $t0, 4  # Move to the next array element
    addi $t2, $t2, 1  # Increment loop counter
    j push_loop
end_push_loop:

    # Reset loop counter and load the address of the new array
    li $t2, 0
    la $t4, new_array

    # Loop to pop each element off the stack into new_array
pop_loop:
    beq $t2, $t1, end_program  # If loop counter equals array length, exit loop

    # Pop from the stack to $t3
    lw $t3, 0($sp)  # Load value from stack to $t3
    addu $sp, $sp, 4  # Increase stack pointer
```

```
        # Store popped element into new_array
        sw $t3, 0($t4)

        addi $t4, $t4, 4  # Move to the next position in new_array
        addi $t2, $t2, 1  # Increment loop counter
        j pop_loop

end_program:
        # Exit
        li $v0, 10
        syscall
```

Stack pointer starts with 268468224 and after pushing elements into it it begins decreasing by 4 bits.

$gp	28	268468224
$sp	29	2147479548
$fp	30	0

$gp	28	268468224
$sp	29	2147479544
$fp	30	0
$ra	31	0

$gp	28	268468224
$sp	29	2147479540
$fp	30	0
$...	21	0

$gp	28	268468224
$sp	29	2147479536
$fp	30	0

$gp	28	268468224
$sp	29	2147479532
$fp	30	0

$gp	28	268468224
$sp	29	2147479528
$fp	30	0

Then, I started popping the elements from the stack. And the values went up by 4 bits everytime

$gp		28	268468224
$sp		29	2147479532
$fp		30	0

$gp		28	268468224
$sp		29	2147479536
$fp		30	0

$gp		28	268468224
$sp		29	2147479540
$fp		30	0

$gp		28	268468224
$sp		29	2147479544
$fp		30	0

$gp		28	268468224
$sp		29	2147479548
$fp		30	0

The elements were just 1 to 5.

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x7fffefe0	0	0	5	4	3	2	1	0