

Problem 1:

In this problem we were asked to search for an element and print the index, else -1. What I've done is the function only which was what the question asked for. The array contains 5 elements for example from 1-5. The t0 is to be set to change the item to search for. This is assuming that this function would be used in a custom program. I have the main label and the function label, with jal to function to have the ra set to the printing instruction after that and then exit the program. The function will search by incrementing the pointer of array, and checks if it was found. If it was found then the s7 register is set to the index, and jr the ra to print what is in s7.

The screenshot shows the Mars IDE interface. The top panel displays the Data Segment with memory addresses from 0x10010000 to 0x10010100. The bottom panel shows the execution output, indicating that the program finished running with an index of 2.

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	0x55646e49	0x666f2078	0x0000203a
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Mars Messages: Run I/O

Index of: 2
-- program is finished running --

Reset: reset completed.

Index of: -1
-- program is finished running --

The screenshot shows the Mars IDE interface with assembly code and execution output. The code defines a function to search for an element in an array. The output shows the program finished running with an index of 2.

```

10  la $a0, array
11  syscall
12
13  li $v0, 1          #printing the index int that is defined from the function
14  move $a0, $s7      #printing what is in s7 which was found in function
15  syscall
16
17  li $v0, 10         #at the end of the function exit the program
18  syscall
19
20  function:
21  li $t0, 3          #item to search for (This is to be changed by the user)
22  la $t1, array      #pointing the t1 at the beginning of the array
23  li $t2, 5          #number of items in the array
24  li $t4, 0          #loading zero in t4 which is the counter
25  search:
26  lbu $t5, ($t1)      #loading the byte that t1 is pointing at
27  beq $t5, $t0, found #if it was found then branch to found
28  add $t4, $t4, 1     #increment the t4 by which is the counter
29  add $t1, $t1, 4     #increment the pointer by 4 to check the next item
30  beq $t4, $t2, notFound #at the end of the loop check jump to not found if it was not found

```

Line: 14 Column: 67 Show Line Numbers

Mars Messages: Run I/O

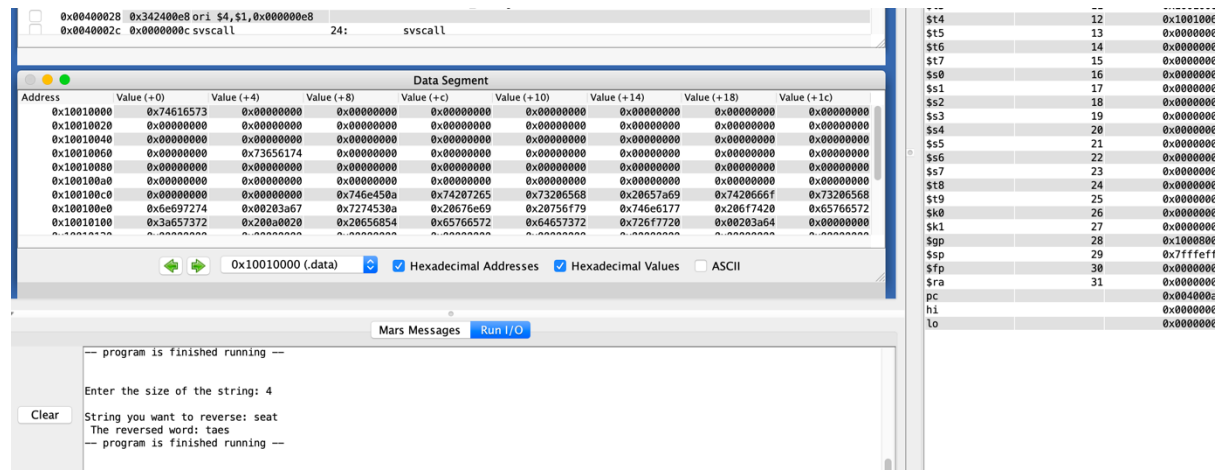
Index of:
-- program is finished running --

Reset: reset completed.

Index of: 2
-- program is finished running --

Problem 2:

In this problem, I simply had a pointer to the beginning of the string entered. I also reserved space for another string to be storing the reverse. I set the pointer of the second string to the end. In a loop, I load the byte in the first pointer of the first string, and then I store its item in the other pointer of the second string. I increment the pointer of the first string and decrement the second pointer, until the pointer points to zero, then print the reversed string.



Problem 3:

In this problem, I made the conversion into 6 main function to convert from a number to another, instead of 12. I made any number entered converted to a decimal number. This decimal number is then converted to binary or hexa or octal based on the user's desired input. First of all I retrieve the user input size which is very important in my program to keep looping over the string plus which makes in consideration the two bytes for the prefixes. This string is parsed from the beginning and checked if it contains b, O, or X in their second element in the string. The handlers are activated based in the that, otherwise it goes to the decimal handler immediately. The handlers makes the string ready to enter the decimal conversion function by increasing the pointer and decreasing to the size. After the number goes to the decimal conversion, the user is then asked to enter the prefix related to the desired conversion. The conversion is made through the check1 function and goes to either binary, or octal, or hexa functions. After the conversion is made the result is stored in the bin_string, or octal_string, or hexa_string, or dec_string, and then printed. The conversion from the number to another was made by parsing the string and then loading them into registers which are then compared with their appropriate signs. The hexa letter are translated into their equivalent ascii numbers such that is works. **We assumed the user is entering the exact numbers as the characters in the string, otherwise the program will go into an infinite loop.** The conversion from the number to another was made by parsing the string and then loading them into registers which are then compared with their appropriate signs.

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x65746e45	0x68742072	0x69732065	0x6f20657a	0x68742066	0x74732065	0x676e6972	0x756c7020
0x10010020	0x0a322073	0x746e4500	0x74207265	0x73206568	0x6e697274	0x69622067	0x7972616e	0x30203d20
0x10010040	0x4f202c62	0x6c617463	0x30203d20	0x48202c4f	0x20617865	0x5830203d	0x6e61202c	0x68742064
0x10010060	0x65642065	0x616d6963	0x756e206c	0x7265626d	0x73797320	0x206d6574	0x6c6c6977	0x76616820
0x10010080	0x6f6e2065	0x65727020	0x3a786966	0x68540020	0x756e2065	0x7265626d	0x73797320	0x206d6574
0x100100a0	0x20756f79	0x74616e77	0x206f7420	0x766e6f63	0x20747265	0x6e696228	0x20797261	0x6230203d
0x100100c0	0x634f202c	0x206c6174	0x4f30203d	0x6548202c	0x3d206178	0x2c583020	0x646e6120	0x63656420
0x100100e0	0x6c616d69	0x30203d20	0x203a2964	0x00313400	0x30316230	0x31303031	0x0000000a	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

Reset: reset completed.
Enter the size of the string plus 2
8
Enter the string binary = 0b, Octal = 00, Hexa = 0X, and the decimal number system will have no prefix: 0b101001
The number system you want to convert (binary = 0b, Octal = 00, Hexa = 0X, and decimal = 0d): 0d
41
— program is finished running (dropped off bottom) —

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x65746e45	0x68742072	0x69732065	0x6f20657a	0x68742066	0x74732065	0x676e6972	0x756c7020
0x10010020	0x0a322073	0x746e4500	0x74207265	0x73206568	0x6e697274	0x69622067	0x7972616e	0x30203d20
0x10010040	0x4f202c62	0x6c617463	0x30203d20	0x48202c4f	0x20617865	0x5830203d	0x6e61202c	0x68742064
0x10010060	0x65642065	0x616d6963	0x756e206c	0x7265626d	0x73797320	0x206d6574	0x6c6c6977	0x76616820
0x10010080	0x6f6e2065	0x65727020	0x3a786966	0x68540020	0x756e2065	0x7265626d	0x73797320	0x206d6574
0x100100a0	0x20756f79	0x74616e77	0x206f7420	0x766e6f63	0x20747265	0x6e696228	0x20797261	0x6230203d
0x100100c0	0x634f202c	0x206c6174	0x4f30203d	0x6548202c	0x3d206178	0x2c583020	0x646e6120	0x63656420
0x100100e0	0x6c616d69	0x30203d20	0x203a2964	0x41453000	0x36330030	0x0a323135	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

Reset: reset completed.
Enter the size of the string plus 2
5
Enter the string binary = 0b, Octal = 00, Hexa = 0X, and the decimal number system will have no prefix: 36512
The number system you want to convert (binary = 0b, Octal = 00, Hexa = 0X, and decimal = 0d): 0X
8EA0
— program is finished running (dropped off bottom) —