

Project - Cash Register

CS 0447 — Computer Organization & Assembly Language

See CourseWeb for Due Date

The purpose of this project is for you to practice writing functions in assembly language. The hardware for this project is a cash register for a wine store as shown below:

Wine List		
Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad		
7	8	9
4	5	6
1	2	3
	0	

Receipt
CS0447
Bad Wine & Topsy Spirits
University of Pittsburgh
Cabernet Sauvignon \$15.00
Merlot \$12.00
Syrah \$7.00
Pinot Noir \$9.00
Malbec \$11.00
Zinfandel \$8.00
Chadonnay \$10.00
Sauvignon Blanc \$11.00
Riesling \$9.00
Total: \$92.00
Cash: \$100.00
Change: \$8.00

The Cash Register (Mars Tool) can be found in `cash_register.zip` located in the CourseWeb under this project. Extract all files to your `[.]/mars4_5/mars/tools` directory. If you extract all files to the right directory, when you run the MARS program, you should see "Cash Register (Register) V0.1" under the "Tools" menu.

Introduction to the Cash Register (Mars Tool)

A cash register consists of a number of buttons, a simple display, and a receipt printer. When user press a button, the value of the register `$t9` will be changed as follows:

Button	Value in \$t9	Button	Value in \$t9
1	1	Merlot	12
2	2	Syrah	13
3	3	Pinot Noir	14
4	4	Malbec	15
5	5	Zinfandel	16
6	6	Chadonnay	17
7	7	Sauvignon Blanc	18
8	8	Riesling	19
9	9	Cancel	21
0	10	Total	22
Cabernet Sauvignon	11	Cash	23

Note that the value of the register `$t9` must be 0 before user press a button. Otherwise, the value will not be updated by the Cash Register's button.

A simple display is on top of buttons. This display is used to display value of each wine that user select. It also display the total after user press the Total button. When user press numeric keypad, this display also displays the current value. To display a value on the simply display, simply set the value of the register `$t8` to a desired value. For example, if you set the value of `$t8` to 15 (e.g., `addi $t8, $zero, 15`), the display will show \$15.00. Note that the dollar sign (\$) and decimal point (.00) will be displayed automatically. You simply need to set `$t8` to an integer value.

For the receipt printer, it simply displays a long null terminated string starting at the address 0xffff8000 until it encounters a null character (0). Again, the first character of the string must be stored at the address 0xffff8000, the second character must be stored at the address 0xffff8001, and so on. Note that there is no center/left/right alignment on the receipt. You have to align all characters manually. For example, to print a header just like the picture shown above, you simply have to store the following string:

```
"\n          CS0447\n  Bad Wine & Topsy Spirits\n  University of Pittsburgh"
```

starting at the address 0xffff8000. If user press the "Cabernet Sauvignon" button, the string starting at the address 0xffff8000 should be changed to:

```
"\n          CS0447\n  Bad Wine & Topsy Spirits\n  University of Pittsburgh\n\nCabernet Sauvignon  $15.00"
```

Note that the receipt display has a fix width and the font is a mono-spaced font (every character occupies the same width). I will be easy to align every item nicely.

Price of Each Wine

The price of each wine is shown in the table below:

Type of Wine	Price
Cabernet Sauvignon	\$15.00
Merlot	\$12.00
Syrah	\$7.00
Pinot Noir	\$9.00
Malbec	\$11.00
Zinfandel	\$8.00
Chardonnay	\$10.00
Sauvignon Blanc	\$11.00
Riesling	\$9.00

For simplicity, taxes are included.

Behavior of a Cash Register

The behavior of your cash register can be described as a series of states shown below:

State	Behavior
1	<ol style="list-style-type: none"> 1. Clear total to 0 2. Clear the simple display to \$0.00 3. Display the header on the receipt 4. Wait for user to press a wine button. Once user press a wine button, go to state 2.
2	<ol style="list-style-type: none"> 1. Display the price of the selected wine on the simple display screen 2. Display the name and the price of the selected wine on the receipt 3. Accumulate total with the price of the selected wine 4. Wait for user to press a button <ul style="list-style-type: none"> - If user press a wine button, go to back to state 2 - If user press the "Cancel" button, go to state 1 - If user press the "Total" button, go to state 3
3	<ol style="list-style-type: none"> 1. Display total on the simple display 2. Display total on the receipt 3. Clear the cash value to 0 4. Wait for user to press a button <ul style="list-style-type: none"> - If user press the "Cancel" button, go to state 1 - If user press a numeric button, go to state 4
4	<ol style="list-style-type: none"> 1. $\text{cash value} = (\text{cash value} * 10) + \text{selected button number}$ 2. Display cash value of the simple display 3. Wait for user to press a button <ul style="list-style-type: none"> - If user press the "Cancel" button, go to state 1 - If user press a numeric button, go to state 4 - If user press the "Cash" button, go to state 5
5	<ol style="list-style-type: none"> 1. Display cash value on the receipt 2. $\text{change} = \text{cash value} - \text{total}$ 3. Display change on the simple display 4. Display change on the receipt 5. Wait for user to press a wine button. Once user press a wine button, go to state 6.
6	<ol style="list-style-type: none"> 1. Clear total to 0 2. Display the header on the receipt 3. Go to state 2

We will assume that a sequence of user input will always follows the behavior shown above. Also assume that user always enter cash value more than the total value before pressing the "Cash" button.

Requirements

Your program must consists of a number of functions listed below:

1. `_strCopy`: Copy a source null-terminated string to a specific address.
 - Arguments:
 - `$a0`: An address of the first character of the first source string
 - `$a1`: An address of a buffer to store a null-terminated string
 - Return Value: None

Note that the copy of the source string should be a null-terminated string.

2. `_strConcat`: Concatenate the first source null-terminated string by the second source null-terminated string.
 - Arguments:
 - `$a0`: An address of the first character of the first source string
 - `$a1`: An address of the first character of the second source string
 - Return Value: None

Note that this function **MUST** use the function `_strCopy` as a subroutine. This can be done by first search for the address of the null character of the first source string. Then use it as the address of the buffer of the function `_strCopy`. **Note** that this function will help you simplify the process of display content on the receipt. For example, the code fragment below display a header and then an item on the receipt:

```
.data
    header: .ascii "\n      CS0447\n\n"
    merlot: .ascii "Merlot      $12.00"
.text
:
    lui $a0, 0xffff
    ori $a0, $a0, 0x8000
    la $a1, header
    jal _strConcat
:
    lui $a0, 0xffff
    ori $a0, 0x8000
    la $a1, merlot
    jal _strConcat
:
```

3. `_div10`: Returns a quotient and a remainder of a given positive integer divided by 10.

- Argument:
 - \$a0: A positive integer
 - Return Values:
 - \$v0: Quotient
 - \$v1: Remainder
4. `_getNumDigits`: Returns the number of digits of an given positive integer
- Argument:
 - \$a0: A given positive integer
 - Return Value:
 - \$v0: The number of digits of an given positive integer

Note that this function will help you align the receipt especially total, cash, and change. These value can have one or more digits. By knowing the number of digits will allow you to insert correct number of spaces before you print a value.

5. `_numToPriceString`: Convert a positive integer into a null-terminate string in the form of `$xx.00` where `xx` is the value of the given positive integer.
- Arguments:
 - \$a0: A given positive integer
 - \$a1: An address of a buffer to store a null-terminated string
 - Return Value: None

This function should help you simplify the process of display content on the receipt. **Note** that the function `_getNumDigits` should help you develop this function.

NOTE that you must follow all calling conventions (call a function, return to caller, arguments, return values, and register sharing). A test program named `functionTester.asm` is provided. Simply implement each function at its associated label. **DO NOT** modify the main or `.data` of the program. **Note** that you are allowed to have `.data` under your function. For example:

```
_myFunction:
.data
    msg1:  .asciiz  ".00"
.text
    :
    jr    $ra
```

Note that this tester program will use the regular MARS console to display result. The output of `functionTester.asm` should look like the following:

```
Test function _strCopy
Please enter a string: Hello
You just entered  : Hello
Result of _strCopy: Hello
If both strings are identical, your _strCopy works properly.

Please enter another string: World!!!
```

```

String 1: Hello
String 2: World!!!
Result of _strConcat: HelloWorld!!!
If the result looks fine, your _strConcat works properly.

Test function _div10
Please enter a positive integer: 12345
From your _div10, 12345 divided by 10 is 1234 and the remainder is 5
If the quotient and the remainder are correct, your function _div10 works properly.

Test function _getNumDigits
Your _getNumDigits says 12345 has 5 digits.
If the number of digits is correct, your function _getNumDigits works correctly.

Test function _numToPriceString
The output string from _numToPriceString with the number 12345 as the argument is the string "$12345.00"
If the output string looks correct, your function _numToPriceString works properly.

-- program is finished running --

```

Note that you are allowed to have more functions as you wish. But make sure all of your functions follow calling conventions discussed in class.

What to Do?

For this project, write a MIPS assembly program named `cashRegister.asm` such that when the program is running, the Cash Register (Mars Tool) will behave just like an actual cash register hardware. Simply copy and paste all five functions from the tester program to your `cashRegister.asm`. **Note** that since registers `$t8` and `$t9` are used for sending command and receiving button input, do not use these registers for any other purpose.

Example

This example shows the cash register after a sequence of buttons being pressed. The sequence of buttons is as follows:

[Merlot] → [Malbec] → [Cancel] → [Cabernet Sauvignon] → [Melbec] → [Riesling] → [Total] → [4] → [0] → [Cash] → [Syrah]

1. When your program is just started, the cash register should look like the following:

Wine List		
Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad		
7	8	9
4	5	6
1	2	3
	0	

2. After the "Merlot" button is pressed, your cash register should look like the following:

The Cash Register interface shows a total of \$12.00. The Wine List table has 'Merlot' selected. The Keypad is visible. The Receipt panel shows the transaction details.

Wine List			Keypad		
Cabernet Sauvignon	Merlot	Syrah	7	8	9
Pinot Noir	Malbec	Zinfandel	4	5	6
Chadonnay	Sauvignon Blanc	Riesling	1	2	3
Cancel	Total	Cash		0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Merlot \$12.00

3. After the "Malbec" button is pressed, your cash register should look like the following:

The Cash Register interface shows a total of \$11.00. The Wine List table has 'Malbec' selected. The Keypad is visible. The Receipt panel shows the transaction details.

Wine List			Keypad		
Cabernet Sauvignon	Merlot	Syrah	7	8	9
Pinot Noir	Malbec	Zinfandel	4	5	6
Chadonnay	Sauvignon Blanc	Riesling	1	2	3
Cancel	Total	Cash		0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Merlot \$12.00
Malbec \$11.00

4. After the "Cancel" button is pressed, your cash register should look like the following:

The Cash Register interface shows a total of \$0.00. The Wine List table is empty. The Keypad is visible. The Receipt panel shows the transaction details.

Wine List			Keypad		
Cabernet Sauvignon	Merlot	Syrah	7	8	9
Pinot Noir	Malbec	Zinfandel	4	5	6
Chadonnay	Sauvignon Blanc	Riesling	1	2	3
Cancel	Total	Cash		0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

5. After buttons "Cabernet Sauvignon", "Malbec", and "Riesling" are pressed (in that order), your cash register should look like the following:

Cash Register

\$9.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Cabernet Sauvignon \$15.00
Malbec \$11.00
Riesling \$9.00

6. After the "Total" button is pressed, your cash register should look like the following:

Cash Register

\$35.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Cabernet Sauvignon \$15.00
Malbec \$11.00
Riesling \$9.00
Total: \$35.00

7. After the numeric button 4 is pressed, your cash register should look like the following:

Cash Register

\$4.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Cabernet Sauvignon \$15.00
Malbec \$11.00
Riesling \$9.00
Total: \$35.00

8. After the numeric button 0 is pressed, your cash register should look like the following:

Cash Register

\$40.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Cabernet Sauvignon \$15.00
Malbec \$11.00
Riesling \$9.00
Total: \$35.00

9. After the "Cash" button is pressed, your cash register should look like the following:

Cash Register

\$5.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Cabernet Sauvignon \$15.00
Malbec \$11.00
Riesling \$9.00
Total: \$35.00
Cash: \$40.00
Change: \$5.00

10. After the "Syrah" button is press (after "Cash"), this represents a new transaction. Your cash register should look like the following:

Cash Register

\$7.00

Wine List

Cabernet Sauvignon	Merlot	Syrah
Pinot Noir	Malbec	Zinfandel
Chadonnay	Sauvignon Blanc	Riesling
Cancel	Total	Cash

Keypad

7	8	9
4	5	6
1	2	3
	0	

Receipt

CS0447
Bad Wine & Tippy Spirits
University of Pittsburgh

Syrah \$7.00

Submission

The due date of this project is on the CourseWeb. Late submissions will not be accepted. You should submit the file `cashRegister.asm` and `functionTester.asm` via CourseWeb.