Lab 2

Instructions

Complete each task and demonstrate the working program to your tutor. Tasks should be demonstrated using AVR Studio's simulator. You will have two lab sessions to work on this lab, and all questions must be marked by the end of the second session.

Part A – Reverse String (2 Marks)

Implement a program that loads a null-terminated string from program memory and pushes it onto the stack, then writes out the reversed string into data memory.

Eg "abc",0 will be stored in RAM as "cba",0

Dictionary Macro

The following macro can be used to define a linked list of null-terminated strings in program memory.

```
.set NEXT_STRING = 0x0000
.macro defstring ; str
      .set T = PC           ; save current position in program memory
      .dw NEXT_STRING << 1 ; write out address of next list node
      .set NEXT_STRING = T ; update NEXT_STRING to point to this node

      .if strlen(@0) & 1 ; odd length + null byte
            .db @0, 0
      .else ; even length + null byte, add padding byte
            .db @0, 0, 0
      .endif
.endmacro
```

The macro is used like this:

```
.cseg
rjmp start
defstring "macros"
defstring "are"
defstring "fun"
```

The first argument to the macro is the string to store.

In memory, the first word will contain (as a byte address) a pointer to the next entry, or 0x0000 if it is the last item in the list. NEXT_STRING can be used to get the address of the first entry. Any code that accesses NEXT_STRING must be after the last use of defstring in the source file.

The remaining bytes will contain the characters in the string and a null character.

Part B – Recursive Linked-list search (4 Marks)

Define a list of strings using the "defstring" macro, and write a recursive function to find the address of the longest string in the list. If there are multiple strings with the same length, return the address of the first one. You may use multiple helper functions if you want, but the main search function must be recursive.

The search function must be passed the byte address of the first entry to search in the Z pointer, and return the byte address of the found string in the Z pointer, and the length of the string in r16. If a null pointer is passed to the function it should return null in Z and 0 in r16.

Part C – Min/max (4 Marks)

Define your own macro called “defint” that defines a linked list of signed 16-bit integers in program memory. Write a recursive function that takes the byte address of the first list entry in the Z pointer, and calculates the largest and smallest integers in the list. The largest integer should be returned in XH:XL, and the smallest in YH:YL.