https://mail.google.com/mail/u/1?ik=a4dada916d&view=pt&search=all&permmsgid=msg-f%3A1711158940326187736&simpl=msg-f%3A17111589403... 3/3

09/21/2021

CSCI-212

Assignment-4

Section 8.6

Note:- Will come back to this section later

Section 9.4

1.Ans It works. Returns non-zero integer. Ran it with gdb. gdb uses octal to display the exit code.

2.Ans:-

```
@ f.s
@ Does nothing but return zero to caller.

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified          @modern syntax

@ Program code
        .text
        .align  2
        .global f
        .type   f, %function

f:
        str     fp, [sp, -4]!    @ save caller frame
                                 @ pointer
        add     fp, sp, 0        @ establish our frame
                                 @ pointer
        mov     r0, 0            @ return values go in r0

        sub     sp, fp, 0        @ delete allocated memory
        ldr     fp, [sp], 4      @ restore caller's frame pointer
        bx      lr               @ back to caller
```

review, dissemination, distribution, or copying of this message is strictly prohibited. If you have received this communication in error, please notify us immediately by e-mail, and delete the original message.

9/17/21, 7:55 PM Gmail - Fwd: First Advantage Corporation: RAKSHA SRINIVAS Screening Notification

P.T.O

(2)

3. Ans:

```
@ g.s
@ Returns 123 to caller.

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified         @ modern syntax
@ Program code
        .text
        .align  2
        .global g
        .type   g, %function

g:
        str     fp, [sp, -4]!   @ save caller frame
                                @ pointer
        add     fp, sp, 0       @ establish our frame pointer
        mov     r0, 123         @ return values go in r0
        sub     sp, fp, 0       @ delete allocated memory
        ldr     fp, [sp], 4     @ restore caller's frame
                                @ pointer
        bx      lr              @ back to caller
```

4. Ans:

```
/* checkRetNos.c
 * calls three assembly language functions and
 * prints their return numbers.
 */

#include <stdio.h>
int positiveNumber(void);
int negativeNumber(void);
int maxNumber(void);
```

P.T.O

Exp. No.	Experiment/Subject		Date
Name		Lab Partner	Locker/ Desk No.	Course & Section No.

```
int main()
{
    int x;
    x= positiveNumber();
    printf ("Here is a positive constant: %i, ", x);
    x= negativeNumber();
    printf ("a negative constant: %i, ", x);
    x= maxNumber();
    printf ("and the maximum number: %i.\n", x);
    return 0;
}

@ positiveNumber.s
@ Returns +100 to caller.
@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified                 @ modern syntax
@ Program code
        .text
        .align 2
        .global positiveNumber
        .type   positiveNumber, %function
positiveNumber:
        str     fp, [sp, -4]!   @ save caller frame pointer
        add     fp, sp, 0       @ establish our frame pointer
        mov     r0, 100         @ return +100
        sub     sp, fp, 0       @ restore stack pointer
        ldr     fp, [sp], 4     @ restore caller's frame pointer
        bx      lr              @ back to caller
```

Signature	Date	Witness/TA	Date

P.T.O

Exp. No.	Experiment/Subject	Date	
Name	Lab Partner	Locker Desk No.	Course & Section No.

```
@ negativeNumber.s
@ Returns -100 to caller.

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified         @ modern syntax

@ Program code
        .text
        .align  2
        .global negativeNumber
        .type   negativeNumber, %function
negativeNumber:
        str     fp, [sp, -4]!   @ save caller frame pointer
        add     fp, sp, 0       @ establish our frame pointer

        mov     r0, -100        @ return -100
        sub     sp, fp, 0       @ restore stack pointer
        ldr     fp, [sp], 4     @ restore caller's frame
                                @ pointer
        bx      lr              @ back to caller

@ maxNumber.s
@ Returns the largest "immediate data" constant to caller
@ which is 8 bits.

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified         @ modern syntax
```

Signature	Date	Witness/TA	Date

P.T.O

```
@ Program code
            .text
            .align 2
            .global maxNumber
            .type maxNumber, %function

maxNumber:
        str    fp, [sp, -4]!   @ save caller frame
                               @ pointer
        add    fp, sp, 0    @ establish our frame pointer
        mov    r0, 0xff  @only 8 bits available for
                         @ immediate
        sub    sp, fp, 0    @ restore stack pointer
        ldr    fp, [sp], 4  @restore caller's frame
                            @ pointer
        bx     lr        @back to caller
```

5. Ans:

```
#include <stdio.h>
int A(void);
int Z(void);
int hashtag(void);

int main()
{
     char aCharacter;
     aCharacter = A();
     printf("Here some characters : %c, ", aCharacter);
     aCharacter = Z();
     printf("%c, ", aCharacter);

     aCharacter = hashtag();
     printf("and %c\n", aCharacter);
     return 0;
}
```

P.T.O

```
@ A.s
@ Returns 'A' to caller.

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified         @ modern syntax
@ Program code
        .text
        .align  2
        .global A
        .type   A, % function
A:
    str     fp, [sp, -4]!   @ save caller frame pointer
    add     fp, sp, 0       @ establish our frame pointer
    mov     r0, 'A'         @ return 'A'
    sub     sp, fp, 0       @ restore stack pointer
    ldr     fp, [sp], 4     @ restore caller's frame pointer
    bx      lr              @ back to caller
@ z.s
@ Returns 'z' to caller

@ Define my Raspberry Pi
        .cpu    cortex-a53
        .fpu    neon-fp-armv8
        .syntax unified         @ modern syntax
@ Program code
```

P.T.O

```
        .text
        .align 2
        .global z
        .type   z, %function

z:

    str  fp, [sp, -4]! @ save caller frame pointer
    add  fp, sp, 0     @ establish our frame pointer
    mov  r0, 'z        @ return 'z'
    sub  sp, fp, 0     @ restore stack pointer
    ldr  fp, [sp], 4   @ restore caller's frame pointer
    bx   lr            @ back to caller
@ hashtag.s
@ Returns '#' to caller
@ Define my Raspberry Pi
        .cpu cortex-a53
        .fpu neon-fp-armv8
        .syntax unified        @ modern syntax
@ Program code
        .text
        .align 2
        .global hashtag
        .type hashtag, %function
hashtag:

    str  fp, [sp, -4]! @ save caller frame pointer
    add  fp, sp, 0     @ establish our frame pointer
    mov  r0, #'#       @ return hashtag
    sub  sp, fp, 0     @ restore stack pointer
    ldr  fp, [sp], 4   @ restore caller's frame
                       @ pointer
    bx   lr            @ back to caller
```

Section 8.6

1. Ans Written record of pc register :-

0x10464
0x1046c
0x10474
0x10478
0x10480
0x1048c
0x10498
0x104a4
0x104a8
0x7ee6718

8 bytes of machine code are in each of the C statements in this program.

2. Ans :- In my instance of the program, Register R4 is being used for reg.

(a) 0x1048c

(b) The contents of the register is one less (44) than the integer I entered when the program prompted me (45).

(c) The C statement uses 12 bytes of program memory