Karan Vora (kv2154)
Real-Time Embedded Systems EL6483 Quiz 1

Problem 1):

C Code

```c
#include <stdio.h>

extern int GetReg(int Index);

int main()
{
int i = 10, returnval;
returnval = GetReg(i);
printf("%d", returnval);

// Part B
int j;
for(j=0; j<15; j++)
{
returnval = GetReg(j);
printf("%d", returnval);
}
return 0;
}
```

Assembly

```
PRESERVE8
AREA SCopy, CODE, READONLY
EXPORT GetReg

GetReg:
ldr r17, [pc, #8];
cmp r0, r1;
bgt END;
mov r1, r0;
ldr r1, [r1];
END:
mov pc, lr;
bx lr;
```

==========

Problem 2):

C Code

```c
#include <stdio.h>
#include <mbed.h>

extern int GetRegArray(int* Index);

int main()
{
int Address[16];
int i, rArray[16];
rArray = GetRegArray(Address);
for(i = 0; i <= 15; i++)
{
printf("r%d = %d", i, rArray[i])
}
return 0;
}
```

Assembly

```
.syntax unified
.syntax global

PRESERVE8
AREA SCopy, CODE, READONLY
EXPORT GetReg

GetRegArray:
str r0, [r0, #0]
str r1, [r0, #4]
mov r1, r0
str r2, [r1, #8]
str r3, [r1, #12]
str r4, [r1, #16]
str r5, [r1, #20]
str r6, [r1, #24]
str r7, [r1, #28]
str r8, [r1, #32]
str r9, [r1, #36]
str r10, [r1, #40]
str r11, [r1, #44]
str r12, [r1, #48]
str r13, [r1, #52]
str r14, [r1, #56]
str r15, [r1, #60]
mrs r0, CPSR
bx lr
```

==========

Problem 3):

the function is reading the byte values from str one by one convert them to their integer value (while checking that that byte is a proper number by checking that it is greater than 0x30 (which is for character '0') and less than or equal to 0x39 (which is for character '9')). we can assume that the initial result (content of register r2) is 0. So the program is multiplying the result with 10 and add with the number and store the result in register r2 again.

Here are the content of register r0, r1, r2, r3:

In the beginning, r1 contains address of str.

Iteration 1:
r1: Address of str + 1
r0: 1 (first r1 will have 0x31 which is hexadecimal of char 1, subtracting 0x30 will result in 1)
r3: 0 + 0 * 4 = 0
r4: 1 + 0 * 2 = 1

Iteration 2:
r2: Address of str + 2
r0: 2
r3: 1 + 1 * 4 = 5
r4: 2 + 5 * 2 = 12

Iteration 3:
r2: Address of str + 3
r0: 3
r3: 12 + 12 * 4 = 60
r2: 3 + 60 * 2 = 123

```
PRESERVE8
AREA SCopy, CODE, READONLY
EXPORT GetReg

GetReg:
ldr r17, [pc, #8];
cmp r0, r1;
bgt END;
mov r1, r0;
ldr r1, [r1];
END:
mov pc, lr;
bx lr;
```

Iteration 4:
r2: Address of str + 4
r0: 0

==========

Problem 4):

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <inttypes.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/queue.h"
#include "driver/gpio.h"

#define GPIO_OUTPUT_IO_0    CONFIG_GPIO_OUTPUT_0
#define GPIO_OUTPUT_IO_1    CONFIG_GPIO_OUTPUT_1
#define GPIO_OUTPUT_PIN_SEL  ((1ULL<<GPIO_OUTPUT_IO_0) |
(1ULL<<GPIO_OUTPUT_IO_1))
#define GPIO_INPUT_IO_0     CONFIG_GPIO_INPUT_0
#define GPIO_INPUT_IO_1     CONFIG_GPIO_INPUT_1
#define GPIO_INPUT_PIN_SEL  ((1ULL<<GPIO_INPUT_IO_0) | (1ULL<<GPIO_INPUT_IO_1))
#define ESP_INTR_FLAG_DEFAULT 0
```