Laboration 2

Mattias Karlsson
Max Niia
Johan Larsson

Task 1

```
.data
.balign 4
        string: .asciz ”\n%d + %d = %d\n”
.text
.global main
.extern printf

main:
        push {ip, lr}
        ldr r0, =string
        mov r1, #10   @ move immediate value 10 into r1
        mov r2, #10   @ move immediate value 10 into r2
        add r3, r1, r2 @ r3 = r1 + r2
        bl printf     @ prints r0 %d + %d = %d (r1 + r2 = r3)
        pop {ip, pc}
```

The output should be 10 + 10 = 20 (verification in screenshot below)

Task 2

[C] int_out function:

```
#include <stdio.h>

void int_out(int a){
    printf(“%X\n”, a);
}
```

[C] main function to test int_out:

```
int main() {
    int_out(4);
    return 0;
}
```

[ASM] Function to load immediate value 4 and 0xBD5B7DDE, bit shift the values arithmetically 1 bit to the right use external function int_out (above)

```
.data
.balign 4
string: .asciz “\n%X\n”
.text
.global main
.extern int_out
.extern printf

main:
    push {ip, lr}
    mov r1, #4
    mov r0, r1, ASR #1
    bl int_out
    mov r1, r0
    ldr r0, string
    bl printf
    ldr r1, =0xBD5B7DDE
    mov r0, r1, ASR #1
    bl int_out
    mov r1, r0
    ldr r0, string
    bl printf
    pop {ip, pc}
```

The output should be 2 and DEADBEEF (which can be seen below in the screenshot)

Task 3:

[C] xor function

```
int xor(int a, int b) {
    return a^b;
}

extern int axor(int a, int b);

int main(){
    printf("%d\n", xor(13, 9));
    printf("%d\n", axor(13, 9));
    return 0;
}
```

[ASM] xor function

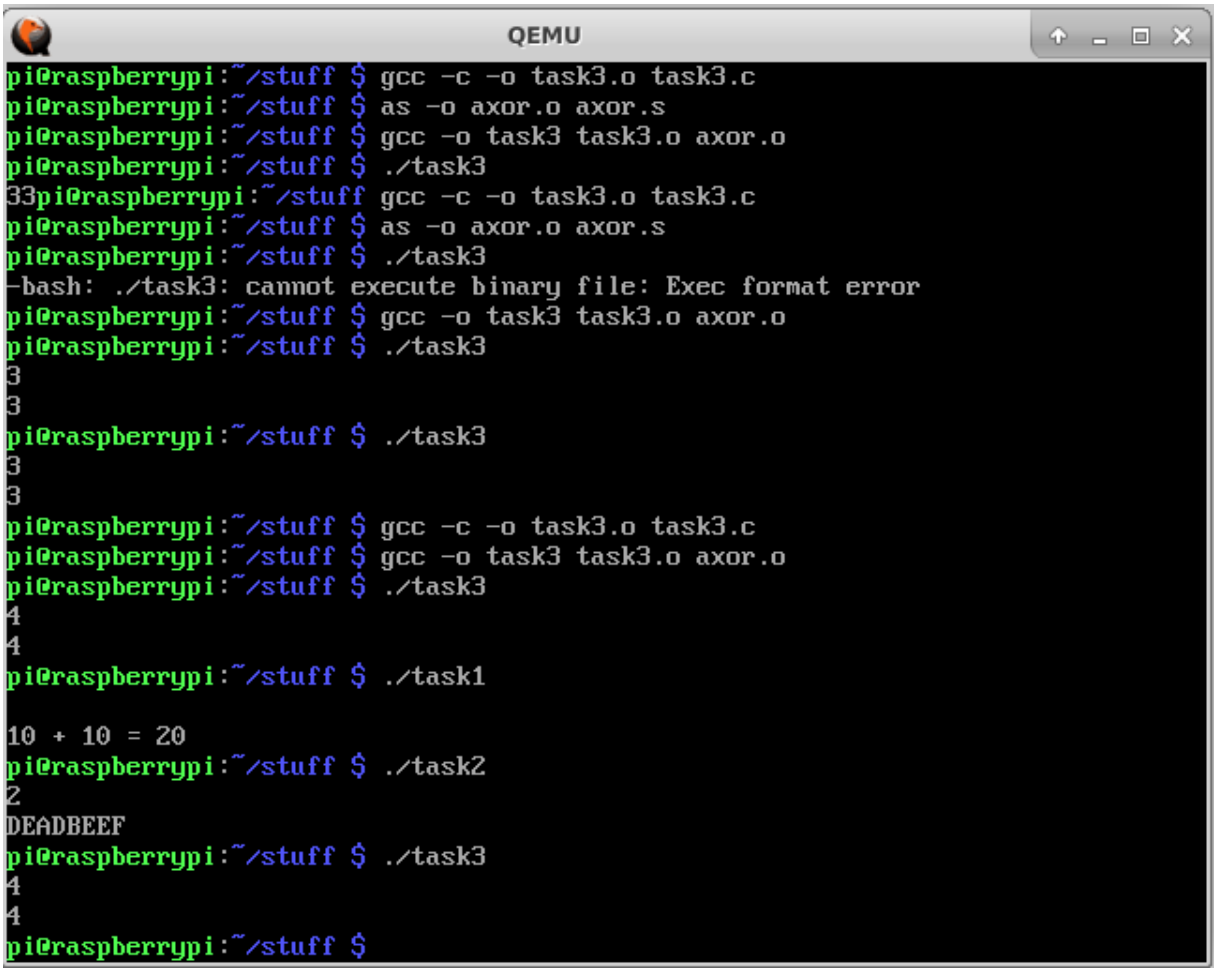
```
.global axor
.p2align 2
.type axor,%function


axor:
    .fnstart
    eor r0, r0, r1
    bx lr
    .fnend
```

The output should be 4 4

The code is well documented if any questions should arise

Output


QEMU
pi@raspberrypi:~/stuff $ gcc -c -o task3.o task3.c
pi@raspberrypi:~/stuff $ as -o axor.o axor.s
pi@raspberrypi:~/stuff $ gcc -o task3 task3.o axor.o
pi@raspberrypi:~/stuff $ ./task3
33pi@raspberrypi:~/stuff gcc -c -o task3.o task3.c
pi@raspberrypi:~/stuff $ as -o axor.o axor.s
pi@raspberrypi:~/stuff $ ./task3
-bash: ./task3: cannot execute binary file: Exec format error
pi@raspberrypi:~/stuff $ gcc -o task3 task3.o axor.o
pi@raspberrypi:~/stuff $ ./task3
3
3
pi@raspberrypi:~/stuff $ ./task3
3
3
pi@raspberrypi:~/stuff $ gcc -c -o task3.o task3.c
pi@raspberrypi:~/stuff $ gcc -o task3 task3.o axor.o
pi@raspberrypi:~/stuff $ ./task3
4
4
pi@raspberrypi:~/stuff $ ./task1
10 + 10 = 20
pi@raspberrypi:~/stuff $ ./task2
2
DEADBEEF
pi@raspberrypi:~/stuff $ ./task3
4
4
pi@raspberrypi:~/stuff $