1. Task 1 riscv assembly

Sohaila Mohammad Kandil
ID: 120200062

```
void main(void) {
  1c:    1141                    addi    sp,sp,-16
  1e:    e406                    sd      ra,8(sp)
  20:    e022                    sd      s0,0(sp)
  22:    0800                    addi    s0,sp,16
  printf("%d %d\n", f(8)+1, 13);
  24:    4635                    li      a2,13
  26:    45b1                    li      a1,12
  28:    00000517                auipc   a0,0x0
  2c:    7a050513                addi    a0,a0,1952 # 7c8 <malloc+0xe8>
  30:    00000097                auipc   ra,0x0
  34:    5f8080e7                jalr    1528(ra) # 628 <printf>
  exit(0);
  38:    4501                    li      a0,0
  3a:    00000097                auipc   ra,0x0
  3e:    274080e7                jalr    628(ra) # 2ae <exit>

0000000000000042 <strcpy>:
#include "kernel/fcntl.h"
#include "user/user.h"
```

1. the value 13 is stored in register a2
2. the main doesn't really call the f function instead it loads the value 12 in a1 which is the result of f(8) + 1 in this line: 26: 45b1 li a1,12
3. we jump to the function printf in this line: 34: 5f8080e7 jalr 1528(ra) # 628 <printf> which starts at address 628 .. the printf function:

```
void
printf(const char *fmt, ...)
{
 628:   711d                    addi    sp,sp,-96
 62a:   ec06            sd      ra,24(sp)
 62c:   e822                    sd      s0,16(sp)
 62e:   1000                    addi    s0,sp,32
 630:   e40c            sd      a1,8(s0)
 632:   e810                    sd      a2,16(s0)
 634:   ec14            sd      a3,24(s0)
 636:   f018            sd      a4,32(s0)
 638:   f41c            sd      a5,40(s0)
 63a:   03043823                sd      a6,48(s0)
 63e:   03143c23                sd      a7,56(s0)
  va_list ap;
```

```
  va_start(ap, fmt);
 642:   00840613               addi    a2,s0,8
 646:   fec43423               sd      a2,-24(s0)
  vprintf(1, fmt, ap);
 64a:   85aa            mv     a1,a0
 64c:   4505                   li      a0,1
 64e:   00000097               auipc   ra,0x0
 652:   dce080e7               jalr    -562(ra) # 41c <vprintf>
}
 656:   60e2                   ld      ra,24(sp)
 658:   6442                   ld      s0,16(sp)
 65a:   6125                   addi    sp,sp,96
 65c:   8082                   ret
```

4. instruction aupc adds the upper 20 bits from the pc and adds 0 to it . So the value in a0 is 30.

5. code to test:

```c
#include "kernel/param.h"
#include "kernel/types.h"
#include "kernel/stat.h"
#include "user/user.h"


void main(void) {
  unsigned int i = 0x00646c72;
  printf("H%x Wo%s \n", 57616, &i);
  exit(0);
}
```

output:

```
$ test_asci_table
HE110 World
$
```

this mapping happened because we print H then we print the hexadecimal of the number 57616 which is E110 , then we print the two letters Wo then we print the asci representation of the string `0x00646c72 since in asci 72 → r  , 6c → 6 , 64 → d so the last part is rld and the result is HE110 World.`