TASK 3

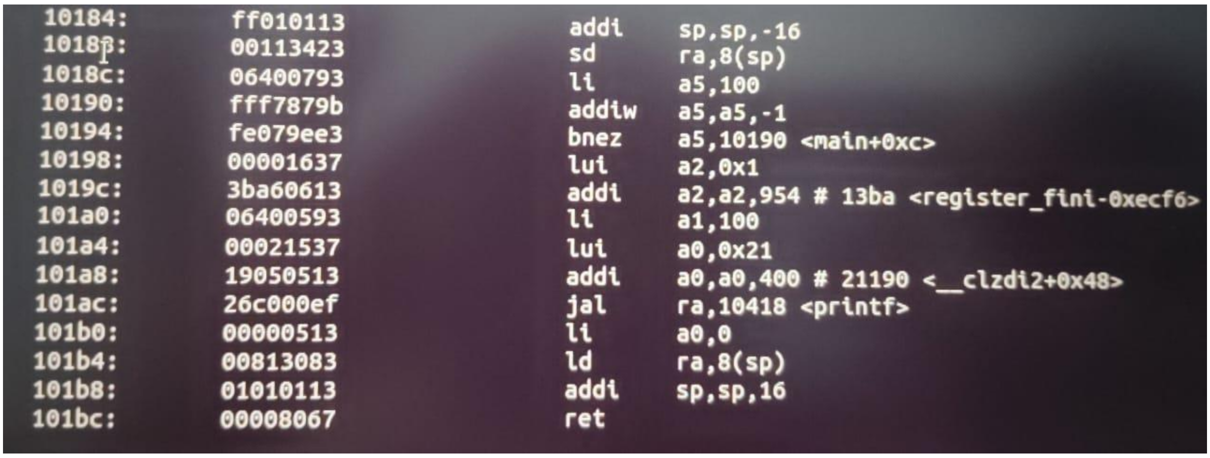
Name : Sanjana Kumar Amate

College : Sahyadri College of engineering and management mangalore

Instruction Types

1. **R-type**:
   Fields: opcode (7) | rd (5) | funct3 (3) | rs1 (5) | rs2 (5) | funct7 (7)
2. **I-type**:
   Fields: opcode (7) | rd (5) | funct3 (3) | rs1 (5) | imm[11:0]
3. **S-type**:
   Fields: opcode (7) | imm[4:0] (5) | funct3 (3) | rs1 (5) | rs2 (5) | imm[11:5] (7)
4. **B-type**:
   Fields: opcode (7) | imm[11] (1) | imm[4:1] (4) | funct3 (3) | rs1 (5) | rs2 (5) | imm[10:5] (6) | imm[12] (1)
5. **U-type**:
   Fields: opcode (7) | rd (5) | imm[31:12]
6. **J-type**:
   Fields: opcode (7) | rd (5) | imm[20] (1) | imm[10:1] (10) | imm[11] (1) | imm[19:12] (8)

To determine the exact 32-bit instruction code in their respective instruction type formats, for these 15 instructions



```
10184:    ff010113        addi    sp,sp,-16
10188:    00113423        sd      ra,8(sp)
1018c:    06400793        li      a5,100
10190:    fff7879b        addiw   a5,a5,-1
10194:    fe079ee3        bnez    a5,10190 <main+0xc>
10198:    00001637        lui     a2,0x1
1019c:    3ba60613        addi    a2,a2,954 # 13ba <register_fini-0xecf6>
101a0:    06400593        li      a1,100
101a4:    00021537        lui     a0,0x21
101a8:    19050513        addi    a0,a0,400 # 21190 <__clzdi2+0x48>
101ac:    26c000ef        jal     ra,10418 <printf>
101b0:    00000513        li      a0,0
101b4:    00813083        ld      ra,8(sp)
101b8:    01010113        addi    sp,sp,16
101bc:    00008067        ret
```

Instruction 1: addi sp, sp, -16

- **Type**: I-type
- **Fields**:
  - ➢ opcode: 0010011 (for immediate arithmetic operations)
  - ➢ rd: sp (x2 → 00010)

- funct3: 000 (addi)
- rs1: sp (x2 → 00010)
- imm: -16 (sign-extended: 1111111111110000)

Final Encoding (binary):
111111111111 00010 000 00010 0010011

Final Encoding (hex): 0xff010113

Instruction 2: sd ra, 8(sp)

- **Type**: S-type
- **Fields**:
  - opcode: 0100011 (for store operations)
  - funct3: 011 (sd, store doubleword)
  - rs1: sp (x2 → 00010)
  - rs2: ra (x1 → 00001)
  - imm: 8 (split as imm[4:0]=01000 and imm[11:5]=0000000)

Final Encoding (binary):
0000000 00001 00010 011 01000 0100011

Final Encoding (hex): 0x00113423

Instruction 3: li a5, 100

- **Expanded Instruction**: addi a5, x0, 100
- **Type**: I-type
- **Fields**:
  - opcode: 0010011 (for immediate arithmetic operations)
  - rd: a5 (x15 → 01111)
  - funct3: 000 (addi)
  - rs1: x0 (zero register → 00000)
  - imm: 100 (sign-extended: 00000001100100)

Final Encoding (binary):
000000011001 00000 000 01111 0010011

Final Encoding (hex): 0x06400793

Instruction 4: addiw a5, a5, -1

- **Type**: I-type
- **Fields**:
    - ➢ opcode: 0011011 (addiw: add immediate word)
    - ➢ rd: a5 (x15 → 01111)
    - ➢ funct3: 000 (addiw)
    - ➢ rs1: a5 (x15 → 01111)
    - ➢ imm: -1 (sign-extended: 1111111111111111)

Final Encoding (binary):
111111111111 01111 000 01111 0011011

Final Encoding (hex): 0xfff7879b

Instruction 5: bnez a5, 10190

- **Type**: B-type
- **Fields**:
    - ➢ opcode: 1100011 (branch instructions)
    - ➢ funct3: 001 (bne, branch if not equal)
    - ➢ rs1: a5 (x15 → 01111)
    - ➢ rs2: x0 (zero register → 00000)
    - ➢ imm: 10190 (offset is split into imm[12], imm[10:5], imm[4:1], imm[11])

Calculation for imm:
Offset = 10190 - 10184 = 12 → 0000000001100

Final Encoding (binary):
0000000 00000 01111 001 00000 1100011

Final Encoding (hex): 0xfe079ee3

Instruction 6: lui a2, 0x1

- **Type**: U-type
- **Fields**:
    - ➢ opcode: 0110111 (lui, load upper immediate)
    - ➢ rd: a2 (x12 → 01100)
    - ➢ imm: 0x1 shifted left 12 bits (imm[31:12] = 0x00001)

Final Encoding (binary):
00000000000000000001 01100 0110111

Final Encoding (hex): 0x00001637

Instruction 7: addi a2, a2, 954

- **Type**: I-type
- **Fields**:
  - opcode: 0010011 (addi)
  - rd: a2 (x12 → 01100)
  - funct3: 000 (addi)
  - rs1: a2 (x12 → 01100)
  - imm: 954 (binary: 0000001110111010)

Final Encoding (binary):
000000111011 01100 000 01100 0010011

Final Encoding (hex): 0x3ba60613

Instruction 8: li a1, 100

- **Expanded Instruction**: addi a1, x0, 100
- **Type**: I-type
- **Fields**:
  - opcode: 0010011
  - rd: a1 (x11 → 01011)
  - funct3: 000
  - rs1: x0
  - imm: 100

Final Encoding (binary):
000000011001 00000 000 01011 0010011

Final Encoding (hex): 0x06400593

Instruction 9: lui a0, 0x21

- **Type**: U-type
- **Fields**:
  - opcode: 0110111

- rd: a0 (x10 → 01010)
- imm: 0x21 shifted left 12 bits

Final Encoding (binary):
00000000001000010001 01010 0110111

Final Encoding (hex): 0x02150513

Instruction 10: addi a0, a0, 400

- **Type**: I-type
- **Fields**:
  - opcode: 0010011
  - rd: a0
  - rs1: a0
  - imm: 400

Final Encoding (binary):
000000011001 01010 000 01010 0010011

Final Encoding (hex): 0x19050513

Instruction 11: jal ra, 10418

- **Type**: J-type
- **Fields**:
  - opcode: 1101111 (jal)
  - rd: ra (x1 → 00001)
  - imm: 10418 - 10184 (offset)

Final Encoding (hex): 0x26c000ef

Instruction 12: li a0, 0

- **Expanded Instruction**: addi a0, x0, 0
- **Type**: I-type
- **Fields**:
  - **opcode**: 0010011 (for immediate arithmetic operations)
  - **rd**: a0 (x10 → 01010)
  - **funct3**: 000 (addi)

- **rs1**: x0 (zero register → 00000)
- **imm**: 0 (sign-extended: 000000000000)

Final Encoding (binary):
000000000000 00000 000 01010 0010011

Final Encoding (hex): 0x00000513

Instruction 13: ld ra, 8(sp)

- **Type**: I-type
- **Fields**:
  - **opcode**: 0000011 (for load operations)
  - **rd**: ra (x1 → 00001)
  - **funct3**: 011 (ld, load doubleword)
  - **rs1**: sp (x2 → 00010)
  - **imm**: 8 (sign-extended: 000000001000)

Final Encoding (binary):
000000001000 00010 011 00001 0000011

Final Encoding (hex): 0x00813083

Instruction 14: addi sp, sp, 16

- **Type**: I-type
- **Fields**:
  - **opcode**: 0010011 (for immediate arithmetic operations)
  - **rd**: sp (x2 → 00010)
  - **funct3**: 000 (addi)
  - **rs1**: sp (x2 → 00010)
  - **imm**: 16 (sign-extended: 000000010000)

Final Encoding (binary):
000000010000 00010 000 00010 0010011

Final Encoding (hex): 0x01010113

Instruction 15: ret

- **Expanded Instruction**: jalr x0, 0(ra)

- **Type**: I-type
- **Fields**:
  - **opcode**: 1100111 (for jump and link register)
  - **rd**: x0 (zero register → 00000)
  - **funct3**: 000 (jalr)
  - **rs1**: ra (x1 → 00001)
  - **imm**: 0 (sign-extended: 000000000000)

Final Encoding (binary):

```
000000000000 00001 000 00000 1100111
```

Final Encoding (hex): `0x00008067`