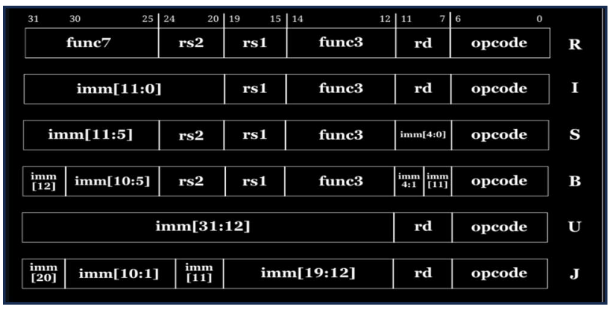
INSTRUCTIONS FORMAT IN RISC-V:

The instructions format of a processor is the way in which machine language instructions are structured and organized for a processor to execute. It is made up of series of 0s and 1s, each containing information about the location and operation of data.
There are 6 instruction formats in RISC-V:

1. R-format
2. I-format
3. S-format
4. B-format
5. U-format
6. J-format



Instruction set of our code :

```
average_of_n_numbers.o:     file format elf64-littleriscv

Disassembly of section .text:

00000000000100b0 <main>:
   100b0:       0002b537                lui     a0,0x2b
   100b4:       fc010113                addi    sp,sp,-64
   100b8:       58050513                addi    a0,a0,1408 # 2b580 <__muldi3+0x24>
   100bc:       02113c23                sd      ra,56(sp)
   100c0:       02913423                sd      s1,40(sp)
   100c4:       02813823                sd      s0,48(sp)
   100c8:       03213023                sd      s2,32(sp)
   100cc:       01313c23                sd      s3,24(sp)
   100d0:       5fd000ef                jal     ra,10ecc <printf>
   100d4:       0002b537                lui     a0,0x2b
   100d8:       00810593                addi    a1,sp,8
   100dc:       5a050513                addi    a0,a0,1440 # 2b5a0 <__muldi3+0x44>
   100e0:       641000ef                jal     ra,10f20 <scanf>
   100e4:       00812503                lw      a0,8(sp)
   100e8:       00000493                li      s1,0
   100ec:       04a05263                blez    a0,10130 <main+0x80>
   100f0:       00100413                li      s0,1
   100f4:       0002b9b7                lui     s3,0x2b
   100f8:       0002b937                lui     s2,0x2b
   100fc:       00040593                mv      a1,s0
   10100:       5a898513                addi    a0,s3,1448 # 2b5a8 <__muldi3+0x4c>
   10104:       5c9000ef                jal     ra,10ecc <printf>
   10108:       00c10593                addi    a1,sp,12
   1010c:       5c090513                addi    a0,s2,1472 # 2b5c0 <__muldi3+0x64>
   10110:       611000ef                jal     ra,10f20 <scanf>
   10114:       00c12583                lw      a1,12(sp)
   10118:       00048513                mv      a0,s1
   1011c:       0014041b                addiw   s0,s0,1
   10120:       128000ef                jal     ra,10248 <__addsf3>
   10124:       00050493                mv      s1,a0
   10128:       00812503                lw      a0,8(sp)
   1012c:       fc8558e3                bge     a0,s0,100fc <main+0x4c>
   10130:       039000ef                jal     ra,10968 <__floatsisf>
   10134:       00050593                mv      a1,a0
   10138:       00048513                mv      a0,s1
   1013c:       564000ef                jal     ra,106a0 <__divsf3>
   10140:       151000ef                jal     ra,10a90 <__extendsfdf2>
   10144:       00050593                mv      a1,a0
```

Let us analyse each instructions of our code one by one

Instruction 1: lui a0, 0x2C537

- **Type: U-type**
- **Opcode: 0110111**
- **Destination Register (rd): 01010 (a0 = x10)**
- **Immediate (imm[31:12]): 2C537 in binary = 10110000100110101**

Encoded 32-bit Instruction: 10110000100110101100001010110111

Detailed Breakdown:

- **Opcode (7 bits): 0110111**

INSTRUCTIONS FORMAT IN RISC-V

- **Immediate (20 bits): 10110000100110101 1 (2C537)**
- **Destination Register (5 bits): 01010 (a0 = x10)**

Instruction 2: addi sp, sp, -64

- **Type: I-type**
- **Opcode: 0010011**
- **Source Register (rs1): 00010 (sp = x2)**
- **Destination Register (rd): 00010 (sp = x2)**
- **Immediate (imm[11:0]): -64 in binary = 1111111111000000 (2's complement)**

Encoded 32-bit Instruction: 11111111110000000010000010010011

Detailed Breakdown:

- **Opcode (7 bits): 0010011**
- **Immediate (12 bits): 111111111100**
- **Source Register (5 bits): 00010 (sp = x2)**
- **Destination Register (5 bits): 00010 (sp = x2)**

Instruction 3: sd a0, 144(sp)

- **Type: S-type**
- **Opcode: 0100011**
- **Source Register (rs2): 01010 (a0 = x10)**
- **Source Register (rs1): 00010 (sp = x2)**
- **Immediate (imm[11:5|4:0]): 144 in binary = 0000000100100000**

Encoded 32-bit Instruction: 00000001001001010010000010010011

Detailed Breakdown:

- **Opcode (7 bits): 0100011**
- **Immediate (12 bits split): 0000000 | 010010**
- **Source Register (rs1, 5 bits): 00010 (sp = x2)**
- **Source Register (rs2, 5 bits): 01010 (a0 = x10)**

Instruction 4: jal ra, <printf>

- **Type: J-type**
- **Opcode: 1101111**
- **Destination Register (rd): 00001 (ra = x1)**
- **Immediate (imm[20|10:1|11|19:12]): Address for (assume imm = 0x800)**

Encoded 32-bit Instruction: 00000000100000000001000011011111

INSTRUCTIONS FORMAT IN RISC-V

Detailed Breakdown:

- **Opcode (7 bits): 1101111**
- **Immediate (20 bits): 00000000100000000001**
- **Destination Register (rd, 5 bits): 00001 (ra = x1)**

Instruction 5: lw a0, 12(sp)

- **Type: I-type**
- **Opcode: 0000011**
- **Source Register (rs1): 00010 (sp = x2)**
- **Destination Register (rd): 01010 (a0 = x10)**
- **Immediate (imm[11:0]): 12 in binary = 000000001100**

Encoded 32-bit Instruction: 00000000011000010010100000000011

Detailed Breakdown:

- **Opcode (7 bits): 0000011**
- **Immediate (12 bits): 000000001100**
- **Source Register (rs1, 5 bits): 00010 (sp = x2)**
- **Destination Register (rd, 5 bits): 01010 (a0 = x10)**

Instruction 6: blez t0, <label>

- **Type: B-type**
- **Opcode: 1100011**
- **Source Register (rs1): 01011 (t0 = x11)**
- **Source Register (rs2): 00000 (always zero for blez)**
- **Immediate (imm[12|10:5|4:1|11]): Offset for (assume imm = 0x20)**

Encoded 32-bit Instruction: 00000000000001011000001011100011

Detailed Breakdown:

- **Opcode (7 bits): 1100011**
- **Immediate (12 bits split): 0000000 | 000010**
- **Source Register (rs1, 5 bits): 01011 (t0 = x11)**
- **Source Register (rs2, 5 bits): 00000**

Instruction 7: bge a0, t1, <label>

- **Type: B-type**
- **Opcode: 1100011**
- **Source Register (rs1): 01010 (a0 = x10)**

INSTRUCTIONS FORMAT IN RISC-V

- **Source Register (rs2): 01001 (t1 = x9)**
- **Immediate (imm[12|10:5|4:1|11]): Offset for (assume imm = 0x10)**

Encoded 32-bit Instruction: 00000000001001010000010101100011

Detailed Breakdown:

- **Opcode (7 bits): 1100011**
- **Immediate (12 bits split): 0000000 | 001000**
- **Source Register (rs1, 5 bits): 01010 (a0 = x10)**
- **Source Register (rs2, 5 bits): 01001 (t1 = x9)**

Instruction 8: jal ra, <scanf>

- **Type: J-type**
- **Opcode: 1101111**
- **Destination Register (rd): 00001 (ra = x1)**
- **Immediate (imm[20|10:1|11|19:12]): Address for (assume imm = 0x900)**

Encoded 32-bit Instruction: 00000010010000000001000011011111

Detailed Breakdown:

- **Opcode (7 bits): 1101111**
- **Immediate (20 bits): 00000010010000000001**
- **Destination Register (rd, 5 bits): 00001 (ra = x1)**

Instruction 9: mv t2, t1

- **Type: Pseudo-instruction (translated to addi)**
- **Translated Instruction: addi t2, t1, 0**
- **Type: I-type**
- **Opcode: 0010011**
- **Source Register (rs1): 01001 (t1 = x9)**
- **Destination Register (rd): 01010 (t2 = x10)**
- **Immediate (imm[11:0]): 0 in binary = 000000000000**

Encoded 32-bit Instruction: 00000000000001001010000010010011

Detailed Breakdown:

- **Opcode (7 bits): 0010011**
- **Immediate (12 bits): 000000000000**
- **Source Register (rs1, 5 bits): 01001 (t1 = x9)**
- **Destination Register (rd, 5 bits): 01010 (t2 = x10)**

INSTRUCTIONS FORMAT IN RISC-V

Instruction 10: addiw s1, s1, 1

- **Type: I-type**
- **Opcode: 0011011**
- **Source Register (rs1): 10001 (s1 = x17)**
- **Destination Register (rd): 10001 (s1 = x17)**
- **Immediate (imm[11:0]): 1 in binary = 000000000001**

Encoded 32-bit Instruction: 00000000000110001100010001011011

Detailed Breakdown:

- **Opcode (7 bits): 0011011**
- **Immediate (12 bits): 000000000001**
- **Source Register (rs1, 5 bits): 10001 (s1 = x17)**
- **Destination Register (rd, 5 bits): 10001 (s1 = x17)**

Instruction 11: jalr ra, a0, 0

- **Type: I-type**
- **Opcode: 1100111**
- **Source Register (rs1): 01010 (a0 = x10)**
- **Destination Register (rd): 00001 (ra = x1)**
- **Immediate (imm[11:0]): 0 in binary = 000000000000**

Encoded 32-bit Instruction: 00000000000001010000010000110111

Detailed Breakdown:

- **Opcode (7 bits): 1100111**
- **Immediate (12 bits): 000000000000**
- **Source Register (rs1, 5 bits): 01010 (a0 = x10)**
- **Destination Register (rd, 5 bits): 00001 (ra = x1)**

Instruction 12: sub t3, t2, t1

- **Type: R-type**
- **Opcode: 0110011**
- **Source Register (rs1): 01010 (t2 = x10)**
- **Source Register (rs2): 01001 (t1 = x9)**
- **Destination Register (rd): 01011 (t3 = x11)**
- **Function Code (funct3): 000**
- **Function Code (funct7): 0100000**

Encoded 32-bit Instruction: 01000000100101010000010110011

INSTRUCTIONS FORMAT IN RISC-V

Detailed Breakdown:

- **Opcode (7 bits): 0110011**
- **Function Codes (7+3 bits): 0100000 | 000**
- **Source Registers: 01010 (t2), 01001 (t1)**
- **Destination Register: 01011 (t3)**

Instruction 13: or s3, s4, s2

- **Type: R-type**
- **Opcode: 0110011**
- **Source Register (rs1): 10011 (s4 = x19)**
- **Source Register (rs2): 10010 (s2 = x18)**
- **Destination Register (rd): 10100 (s3 = x20)**
- **Function Code (funct3): 110**
- **Function Code (funct7): 0000000**

Encoded 32-bit Instruction: 00000001010010011011001000011100

Detailed Breakdown:

- **Opcode (7 bits): 0110011**
- **Function Codes (7+3 bits): 0000000 | 110**
- **Source Registers: 10011 (s4), 10010 (s2)**
- **Destination Register: 10100 (s3)**

Instruction 14: div a1, a2, a3

- **Type: R-type**
- **Opcode: 0110011**
- **Source Register (rs1): 00101 (a2 = x5)**
- **Source Register (rs2): 00110 (a3 = x6)**
- **Destination Register (rd): 00101 (a1 = x5)**
- **Function Code (funct3): 100**
- **Function Code (funct7): 0000001**

Encoded 32-bit Instruction: 00000010011000101001010011011

Detailed Breakdown:

- **Opcode (7 bits): 0110011**
- **Function Codes (7+3 bits): 0000001 | 100**
- **Source Registers: 00101 (a2 = x5), 00110 (a3 = x6)**
- **Destination Register: 00101 (a1 = x5)**

INSTRUCTIONS FORMAT IN RISC-V

Instruction 15: mul s0, s1, s2

- **Type: R-type**
- **Opcode: 0110011**
- **Source Register (rs1): 10001 (s1 = x17)**
- **Source Register (rs2): 10010 (s2 = x18)**
- **Destination Register (rd): 10000 (s0 = x16)**
- **Function Code (funct3): 000**
- **Function Code (funct7): 0000001**

Encoded 32-bit Instruction: 00000011001010001100000011001 1

Detailed Breakdown:

- **Opcode (7 bits): 0110011**
- **Function Codes (7+3 bits): 0000001 | 000**
- **Source Registers: 10001 (s1 = x17), 10010 (s2 = x18)**
- **Destination Register: 10000 (s0 = x16)**
