Section 2.1 — The General-Purpose Registers in ARM (Mazidi)

Chapter 2 · Section 2.1 — Exercises (Mazidi)

> Problems are paraphrased to respect copyright. For theory and examples, see **Mazidi, Ch. 2 §2.1**.

---

1) ARM is a(n) ______-bit microprocessor.

Answer: 32-bit.
Why: Classic ARM (AArch32) uses a 32-bit programming model and word size here.

---

2) The general-purpose registers are ______ bits wide.

Answer: 32 bits.
Why: Registers **R0–R15** are each 32-bit wide in the AArch32 model.

---

3) The value in `MOV R2,#value` is ______ bits wide.

Answer: 8 bits (immediate field as taught in this section).
Why: The introductory encoding uses an **8-bit literal** (later chapters explain rotations/literal pools for larger constants).

---

4) The largest number that an ARM GPR can have is ______ in hex.

Answer: 0xFFFFFFFF.
Why: Unsigned maximum for a 32-bit register.

---

5) What is the result of the following code and where is it kept?

```
MOV   R2,#0x15
MOV   R1,#0x13
ADD   R2,R1,R2
```

Answer: R2 = 0x28 (40 decimal), kept in **R2**.
Why: `ADD Rd,Rn,Op2` → `R2 = R1 + R2 = 0x13 + 0x15 = 0x28`.

---

6) Which of the following is/are illegal?

(a) `MOV R2,#0x50000` (b) `MOV R2,#0x50` (c) `MOV R1,#0x00`
(d) `MOV R1,255` (e) `MOV R17,#25` (f) `MOV R23,#0xF5` (g) `MOV 123,0x50`

Answer: (a), (d), (e), (f), (g) are illegal; (b) and (c) are legal.
Why (brief):

- **(a)** exceeds the simple 8-bit immediate taught here.
- **(b)** legal (8-bit immediate).
- **(c)** legal (zero is allowed).
- **(d)** missing `#` for immediate.
- **(e) R17** does not exist (valid GPRs are **R0–R15**).
- **(f) R23** does not exist.
- **(g)** destination must be a **register**, not an immediate.

---

7) Which of the following is/are illegal?

(a) `ADD R2,#20,R1` (b) `ADD R1,R1,R2` (c) `ADD R5,R16,R3`

Answer: (a) and (c) are illegal; (b) is legal.
Why:

- **(a)** Format is `ADD Rd,Rn,Operand2`; the **immediate** can only be `Operand2`, not `Rn`.
- **(b)** Valid three-operand form `Rd=R1`, `Rn=R1`, `Rm=R2`.
- **(c) R16** is outside the GPR range (only **R0–R15**).

---

8) What is the result of the following code and where is it kept?

```
MOV   R9,#0x25
ADD   R8,R9,#0x1F
```

Answer: R8 = 0x44 (68 decimal), kept in **R8**.
Why: `0x25 + 0x1F = 0x44`.

---

9) What is the result of the following code and where is it kept?

```
MOV   R1,#0x15
ADD   R6,R1,#0xEA
```

Answer: R6 = 0xFF (255 decimal), kept in **R6**.
Why: `0x15 + 0xEA = 0xFF`.

---

10) True or False. We have 32 general-purpose registers in the ARM.

Answer: False.
Why: The classic programmer's model exposes **16 architected registers** (R0–R15); some are special-purpose (`SP=R13`, `LR=R14`, `PC=R15`). Some modes bank a subset, but there are **not 32 GPRs**.

Notes for learners

- Remember the **three-operand** form: `ADD Rd,Rn,Operand2`.
- **GPR range** is **R0–R15**; higher numbers like `R16`, `R23` are invalid.
- The immediate in `MOV` is introduced as **8-bit** here; later you'll learn techniques for forming larger constants.