

♦ **Q1. Print Message (Problem–1)**

Q1: EMU8086 কী?

A: EMU8086 হলো 8086 microprocessor–এর emulator।

Q2: `print` macro কী করে?

A: Screen–এ string print করে।

Q3: `INT 21H` কেন ব্যবহার করা হয়?

A: DOS service call করার জন্য।

♦ **Q2. Input / Output (Problem–2)**

Q4: Keyboard input নেওয়ার instruction কোনটা?

A: `mov ah, 01h + int 21h`

Q5: Output দেখানোর জন্য কোন register লাগে?

A: `DL` register।

Q6: ASCII থেকে number বানাতে 48 কেন বিয়োগ করা হয়?

A: ASCII '0' = 48, তাই numeric value পেতে 48 বাদ দিতে হয়।

♦ **Q3–6. Arithmetic Operations (Addition, Subtraction, Multiplication, Division)**

Q7: `ADD` instruction কী করে?

A: দুইটা value যোগ করে।

Q8: `SUB` instruction কী করে?

A: বিয়োগ করে।

Q9: `MUL` এর result কোথায় থাকে?

A: `AX` register–এ।

Q10: **DIV** এর quotient আর remainder কোথায় থাকে?

A: Quotient → AL, Remainder → AH।

Q11: **AH** zero করা কেন দরকার?

A: Division/MUL ঠিকভাবে কাজ করার জন্য।

♦ **Q7. Temperature Conversion (C ↔ F)**

Q12: Celsius → Fahrenheit formula কী?

A: $F = (C \times 9) / 5 + 32$

Q13: Fahrenheit → Celsius formula কী?

A: $C = (F - 32) \times 5 / 9$

Q14: Integer division করলে fraction কেন আসে না?

A: Assembly-তে default integer arithmetic হয়।

♦ **Q8. Leap Year Check**

Q15: Leap year চেক করার rule কী?

A: বছর 4 দিয়ে ভাগ গেলে leap year।

Q16: remainder কোথায় পাওয়া যায়?

A: **AH** register-এ।

♦ **Q9. Positive / Negative Check**

Q17: Positive বা Negative কীভাবে চেক করো?

A: Comparison (**CMP**) এবং jump instruction দিয়ে।

♦ **Q10. Even / Odd Check**

Q18: Even number কীভাবে চেক করো?

A: Number কে 2 দিয়ে ভাগ করে remainder দেখি।

♦ **Q11. Greatest Among Three**

Q19: Greatest number বের করতে কী লাগে?

A: CMP + conditional jump (JGE, JL)।

Q20: কেন result save করতে হয়?

A: print macro register change করে দেয়।

♦ **Q12. Prime Number**

Q21: Prime number কী?

A: যে সংখ্যা 1 আর নিজে ছাড়া আর কারও দ্বারা বিভাজ্য না।

Q22: Loop কেন দরকার prime check-এ?

A: বারবার division চেক করার জন্য।

♦ **Q13. Sum of First N Natural Numbers**

Q23: Loop counter কোন register?

A: CX register।

Q24: LOOP instruction কী করে?

A: CX কমায়, zero না হলে jump করে।

♦ **Q14. Factorial**

Q25: Factorial কী?

A: $n! = 1 \times 2 \times 3 \times \dots \times n$

Q26: Factorial—এ 16-bit multiply কেন ব্যবহার করো?

A: Result বড় হয়, overflow এড়াতে।

♦ **Q15–16. Array Operation**

Q27: Array কী?

A: একই type—এর multiple data।

Q28: Array access করতে কোন register লাগে?

A: SI / DI register।

Q29: Largest/Smallest বের করতে কী লাগে?

A: Loop + compare।

♦ **Q17. String Reverse**

Q30: String end বোঝাতে কোন symbol?

A: \$ sign।

Q31: Stack ছাড়া reverse কীভাবে করো?

A: End থেকে start পর্যন্ত print করে।

♦ **Q18–19. Palindrome**

Q32: Palindrome কী?

A: উল্টো করলেও একই থাকে।

Q33: String palindrome চেক করতে কয়টা compare লাগে?

A: Length-এর অর্ধেক।

♦ **Q20. Grade System**

Q34: Marks check—এ কোন instruction লাগে?

A: **CMP** + **JAE** / **JL**।

Q35: কেন always top grade আগে চেক করা হয়?

A: Condition overlap এড়ানোর জন্য।

♦ **Q21. Procedure (CALL & RET)**

Q36: Procedure কী?

A: Reusable code block।

Q37: **CALL** কী করে?

A: Procedure—এ control transfer করে।

Q38: **RET** কী করে?

A: Stack থেকে return address নিয়ে ফেরত যায়।

Q39: Stack কেন দরকার?

A: Return address সংরক্ষণ করার জন্য।

● **MOST IMPORTANT VIVA ONE-LINERS**

- **print** macro register modify করে
- **AX** হলো accumulator register
- **CX** loop counter
- **SI/DI** array & string pointer
- **INT 21H** = DOS service
- **CALL + RET** = function call
- Result print করার আগে save করা দরকার