Q1 (a): Differentiate between low level, high level and assembly programming languages with two examples each.

Low Level Languages:
- Machine Language (Binary 0s and 1s)
- Assembly Language (Close to hardware, symbolic representation)
High Level Languages:
- C, Java, Python (easy to understand, portable across systems)

Q1 (b): How does a program work (compilation to execution)?

1. Source Code written by programmer.
2. Preprocessing: Handles directives (#include, #define).
3. Compilation: Converts to assembly.
4. Assembler: Converts assembly to machine code (object code).
5. Linking: Combines object code with libraries.
6. Loader: Loads executable into memory.
7. Execution: CPU executes instructions.

Q2: C program to calculate sum, difference, product, quotient, remainder

```
#include <stdio.h>

int main() {
    int a, b;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);

    printf("Sum = %d\n", a + b);
    printf("Difference = %d\n", a - b);
    printf("Product = %d\n", a * b);

    if (b != 0) {
        printf("Quotient = %d\n", a / b);
        printf("Remainder = %d\n", a % b);
    } else {
        printf("Division by zero not allowed!\n");
    }
    return 0;
}
```

Q3: Compound Interest Calculation

```
#include <stdio.h>
#include <math.h>

int main() {
    double P, r, CI;
```

```
    int t;
    printf("Enter Principal, Rate and Time: ");
    scanf("%lf %lf %d", &P, &r, &t);

    CI = P * pow((1 + r / 100), t) - P;
    printf("Compound Interest = %.2lf\n", CI);

    return 0;
}
```

Q4: USD to PKR conversion

```
#include <stdio.h>

int main() {
    double usd, pkr;
    printf("Enter amount in USD: ");
    scanf("%lf", &usd);
    pkr = usd * 280;
    printf("Amount in PKR = %.2lf\n", pkr);
    return 0;
}
```

Q5: Check positive/even or negative/odd

```
#include <stdio.h>

int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);

    if (num > 0 && num % 2 == 0) {
        printf("Positive and Even\n");
    } else if (num < 0 && num % 2 != 0) {
        printf("Negative and Odd\n");
    } else {
        printf("Does not match conditions\n");
    }
    return 0;
}
```

Q6: Find larger number using ternary operator

```
#include <stdio.h>

int main() {
    int a, b, max;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    max = (a > b) ? a : b;
    printf("Larger number = %d\n", max);
    return 0;
}
```

Q7: Expression Evaluation

Expression: int result = 5 + 3 * 2 > 10 && 8 - 4 != 2;
Step 1: 3 * 2 = 6
Step 2: 5 + 6 = 11
Step 3: 11 > 10 → true (1)
Step 4: 8 - 4 = 4
Step 5: 4 != 2 → true (1)
Step 6: 1 && 1 = 1
Final result = 1