

University of Mila
Faculty of Science and Technology
Department of Process Engineering

Practical Work 02: Solutions

Courses: Introduction to Programming

Level: 1st year ST - ENG & LMD

Semester 02

By:

Dr. KECITA Farouk

Academic Year: 2025/2026

Exercise 01:

Part A : Write a C program that asks the user to enter two integers and checks:

- a) Whether both numbers are positive
- b) Whether at least one number is even

Display appropriate messages for each condition.

Solution:

```
1 #include <stdio.h>
2 int main() {
3     int num1, num2;
4     printf("Enter first integer: ");
5     scanf("%d", &num1);
6     printf("Enter second integer: ");
7     scanf("%d", &num2);
8     // Check if both numbers are positive
9     if (num1 > 0 && num2 > 0) {
10        printf("Both numbers are positive.\n");
11    } else {
12        printf("Not both numbers are positive.\n");
13    }
14
15    // Check if at least one number is even
16    if (num1 % 2 == 0 || num2 % 2 == 0) {
17        printf("At least one number is even.\n");
18    } else {
19        printf("Both numbers are odd.\n");
20    }
21
22    return 0;
23 }
```

Part A Solution

Part B : Write a C program that asks the user to enter a positive integer N, then calculates and displays the sum of all numbers from 1 to N that are divisible by 3 **OR** divisible by 5, but NOT divisible by 15.

Solution:

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, sum = 0;
5
6     printf("Enter a positive integer N: ");
7     scanf("%d", &N);
8
9     if (N <= 0) {
10        printf("Please enter a positive integer.\n");
11        return 1;
12    }
13    for (i = 1; i <= N; i++) {
```

```
14 // Check if divisible by 3 OR 5 but NOT 15
15 if ((i % 3 == 0 || i % 5 == 0) && i % 15 != 0) {
16     sum += i;
17     printf("%d ", i); // Optional: show which numbers are
18         added
19 }
20 printf("\nSum of numbers divisible by 3 or 5 but not 15:
21     %d\n", sum);
22 return 0;
}
```

Part B Solution

Exercise 02:

Part A : Write a C program that asks for a student's grade (0-100) and determines:

- If the grade is valid (between 0 and 100)
- If the student passes (grade ≥ 33)
- Whether they got distinction (grade ≥ 80)

Solution:

```
1 #include <stdio.h>
2 int main() {
3     float grade;
4     printf("Enter student grade (0-100): ");
5     scanf("%f", &grade);
6     // Check if grade is valid
7     if (grade < 0 || grade > 100) {
8         printf("Invalid grade! Grade must be between 0 and
9             100.\n");
10        return 1;
11    } else {
12        printf("Valid grade.\n");
13    }
14    // Check if student passes
15    if (grade >= 33) {
16        printf("Result: PASS\n");
17
18        // Check for distinction
19        if (grade >= 80) {
20            printf("Congratulations! You got DISTINCTION.\n");
21        }
22    } else {
23        printf("Result: FAIL\n");
24    }
25    return 0;
}
```

Part A Solution

Part B : Write a C program that asks the user to enter grades for 10 students (using a loop), and counts:

- Number of students who passed (grade ≥ 33)
- Number of students who failed (grade < 33)
- The highest grade entered

Solution:

```
1 #include <stdio.h>
2 int main() {
3     float grade, highest = 0;
4     int i, passed = 0, failed = 0;
5     printf("Enter grades for 10 students:\n");
6     for (i = 1; i <= 10; i++) {
7         printf("Enter grade for student %d: ", i);
8         scanf("%f", &grade);
9         // Update highest grade
10        if (grade > highest) {
11            highest = grade;
12        }
13        // Count passed and failed students
14        if (grade >= 33) {
15            passed++;
16        } else {
17            failed++;
18        }
19    }
20    printf("\n--- Results ---\n");
21    printf("Number of students passed: %d\n", passed);
22    printf("Number of students failed: %d\n", failed);
23    printf("Highest grade: %.2f\n", highest);
24    return 0;
25 }
```

Part B Solution

Exercise 03:

Part A: Write a C program that asks the user to enter an integer and checks:

- If the number is positive, negative, or zero
- If the number is within the range 10 to 50 (inclusive)

Solution:

```
1 #include <stdio.h>
2 int main() {
3     int num;
4     printf("Enter an integer: ");
5     scanf("%d", &num);
```

```
6 // Check if positive, negative, or zero
7 if (num > 0) {
8     printf("The number is POSITIVE.\n");
9 } else if (num < 0) {
10    printf("The number is NEGATIVE.\n");
11 } else {
12    printf("The number is ZERO.\n");
13 }
14 // Check if number is between 10 and 50 (inclusive)
15 if (num >= 10 && num <= 50) {
16    printf("The number is within the range 10 to 50.\n");
17 } else {
18    printf("The number is NOT within the range 10 to 50.\n");
19 }
20
21 return 0;
22 }
```

Part A Solution

Part B : Write a C program that prints the following pattern for N rows (N entered by user), but only prints numbers that are NOT divisible by 3:

```
1
2
4
5
7
8
10
...
```

Solution:

```
1 #include <stdio.h>
2 int main() {
3     int N, i, count = 0, num = 1;
4     printf("Enter number of rows (N): ");
5     scanf("%d", &N);
6     printf("Pattern (numbers not divisible by 3):\n");
7     while (count < N) {
8         if (num % 3 != 0) {
9             printf("%d\n", num);
10            count++; // Only increment count when we print a
11                    number
12            }
13            num++; // Always move to the next number
14        }
15    return 0;
16 }
```

Part B Solution

Alternative Solution using for loop:

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, printed = 0;
5
6     printf("Enter number of rows (N): ");
7     scanf("%d", &N);
8
9     printf("Pattern (numbers not divisible by 3):\n");
10
11    for (i = 1; printed < N; i++) {
12        if (i % 3 != 0) {
13            printf("%d\n", i);
14            printed++;
15        }
16    }
17
18    return 0;
19 }
```

Alternative Solution

Exercise 04:

Write a C program that repeatedly asks the user to enter numbers until they enter 0. For each number entered:

- If the number is positive and even, add it to a sum
- If the number is negative and odd, count it
- Display the final sum of positive-even numbers and the count of negative-odd numbers

Solution:

```
1 #include <stdio.h>
2 #include <math.h> // For fabs function
3
4 int main() {
5     int num, sum = 0, count = 0;
6
7     printf("Enter numbers (enter 0 to stop):\n");
8
9     do {
10        printf("Enter a number: ");
11        scanf("%d", &num);
12
13        if (num > 0 && num % 2 == 0) {
14            sum += num;
15            printf(" Added %d to sum (positive-even)\n", num);
16        }
17    } while (num != 0);
18 }
```

```
17     else if (num < 0 && num % 2 != 0) {
18         count++;
19         printf("   Counted %d (negative-odd)\n", num);
20     }
21
22 } while (num != 0);
23
24 printf("\n--- Final Results ---\n");
25 printf("Sum of positive-even numbers: %d\n", sum);
26 printf("Count of negative-odd numbers: %d\n", count);
27
28 return 0;
29 }
```

Part B Solution

Alternative Solution using while loop:

```
1 #include <stdio.h>
2
3 int main() {
4     int num, sum = 0, count = 0;
5
6     printf("Enter numbers (enter 0 to stop):\n");
7
8     while (1) {
9         printf("Enter a number: ");
10        scanf("%d", &num);
11
12        if (num == 0) {
13            break; // Exit the loop when 0 is entered
14        }
15
16        if (num > 0 && num % 2 == 0) {
17            sum += num;
18        }
19        else if (num < 0 && num % 2 != 0) {
20            count++;
21        }
22    }
23
24    printf("\nSum of positive-even numbers: %d\n", sum);
25    printf("Count of negative-odd numbers: %d\n", count);
26
27    return 0;
28 }
```

Alternative Solution