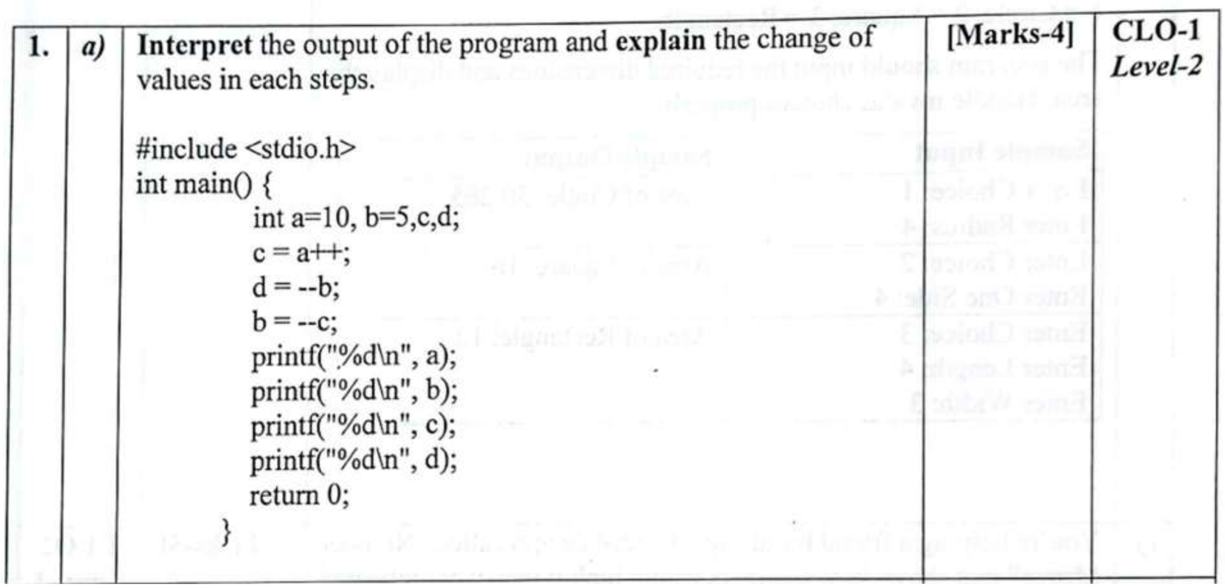
The solutions to the previous two exams of SP

Summer-25(batch 44)



1. *a)* **Interpret** the output of the program and **explain** the change of values in each steps. **[Marks-4]** **CLO-1** ***Level-2***

```
#include <stdio.h>
int main() {
            int a=10, b=5,c,d;
            c = a++;
            d = --b;
            b = --c;
            printf("%d\n", a);
            printf("%d\n", b);
            printf("%d\n", c);
            printf("%d\n", d);
            return 0;
      }
```

Output:

```
11

9

9

4
```

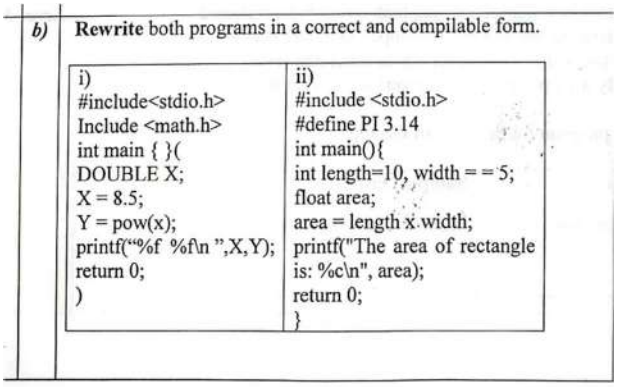
Explanation:

I. **a** starts at 10, and after the post-increment (a++), it becomes 11.

II. **b** starts at 5, and after the pre-decrement (--b), it becomes 4. Then, after the pre-decrement of c, b becomes 9.

III. **c** starts uninitialized, but it gets the value of a++ (which is 10), and then it is decremented to 9.

IV. **d** gets the value of --b (which is 4).



b) **Rewrite** both programs in a correct and compilable form.

i)
#include<stdio.h>
Include <math.h>
int main { }(
DOUBLE X;
X = 8.5;
Y = pow(x);
printf("%f %f\n ",X,Y);
return 0;
)
	ii)
	#include <stdio.h>
	#define PI 3.14
	int main(){
	int length=10, width = = 5;
	float area;
	area = length x width;
	printf("The area of rectangle is: %c\n", area);
	return 0;
	}

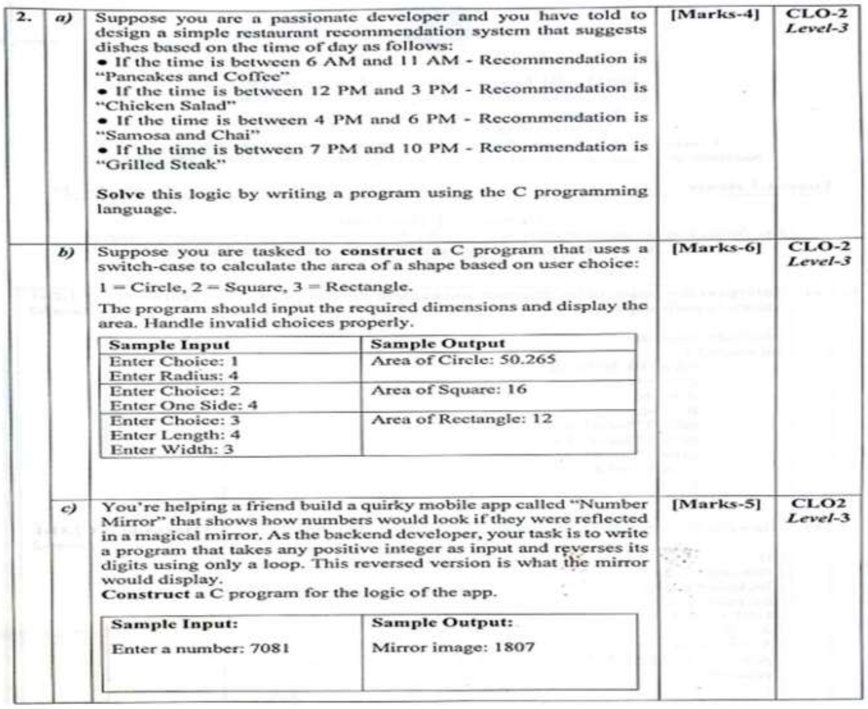
1st problem:

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

int main() {
  double x;
  double y;
  x = 8.5;
  y = pow(x, 2);
  printf("%f %f\n", x, y);
  return 0;
}
```

2nd problem:

```
#include <stdio.h>
#define PI 3.14

int main() {
  int length = 10, width = 5;
  float area;
```

```
        area = length * width;  // Area calculation for a rectangle (length * width)


        printf("The area of the rectangle is: %.2f\n", area);

        return 0;

    }
```



<table>
<tr><td>2.</td><td><i>a)</i></td><td>Suppose you are a passionate developer and you have told to design a simple restaurant recommendation system that suggests dishes based on the time of day as follows:
• If the time is between 6 AM and 11 AM - Recommendation is "Pancakes and Coffee"
• If the time is between 12 PM and 3 PM - Recommendation is "Chicken Salad"
• If the time is between 4 PM and 6 PM - Recommendation is "Samosa and Chai"
• If the time is between 7 PM and 10 PM - Recommendation is "Grilled Steak"

<b>Solve</b> this logic by writing a program using the C programming language.</td><td>[Marks-4]</td><td>CLO-2
<i>Level-3</i></td></tr>
<tr><td></td><td><i>b)</i></td><td>Suppose you are tasked to <b>construct</b> a C program that uses a switch-case to calculate the area of a shape based on user choice:
1 = Circle, 2 = Square, 3 = Rectangle.
The program should input the required dimensions and display the area. Handle invalid choices properly.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>Enter Choice: 1
Enter Radius: 4</td><td>Area of Circle: 50.265</td></tr>
<tr><td>Enter Choice: 2
Enter One Side: 4</td><td>Area of Square: 16</td></tr>
<tr><td>Enter Choice: 3
Enter Length: 4
Enter Width: 3</td><td>Area of Rectangle: 12</td></tr>
</table>
</td><td>[Marks-6]</td><td>CLO-2
<i>Level-3</i></td></tr>
<tr><td></td><td><i>c)</i></td><td>You're helping a friend build a quirky mobile app called "Number Mirror" that shows how numbers would look if they were reflected in a magical mirror. As the backend developer, your task is to write a program that takes any positive integer as input and reverses its digits using only a loop. This reversed version is what the mirror would display.
<b>Construct</b> a C program for the logic of the app.
<table>
<tr><th>Sample Input:</th><th>Sample Output:</th></tr>
<tr><td>Enter a number: 7081</td><td>Mirror image: 1807</td></tr>
</table>
</td><td>[Marks-5]</td><td>CLO2
<i>Level-3</i></td></tr>
</table>

2.a:

```
#include <stdio.h>

int main() {

   int hour;

   printf("Enter the current hour (0-23): ");

   scanf("%d", &hour);

   if (hour >= 6 && hour < 11) {

      printf("Recommendation: Pancakes and Coffee\n");

   } else if (hour >= 12 && hour < 15) {
        printf("Recommendation: Chicken Salad\n");
    } else if (hour >= 16 && hour < 18) {
        printf("Recommendation: Samosa and Chai\n");
    } else if (hour >= 19 && hour < 22) {
        printf("Recommendation: Grilled Steak\n");
    } else {
        printf("No recommendations available at this hour.\n");
    }

    return 0;
}
```

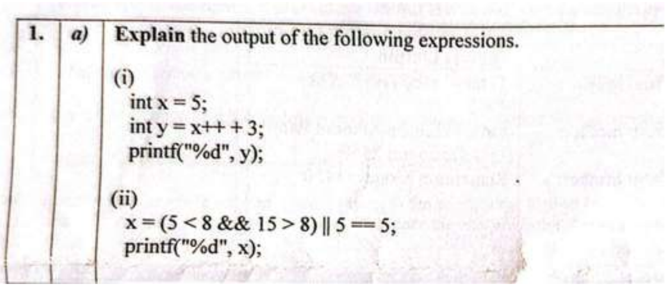
2.b:-

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

int main() {
    int choice;
    float area;

    printf("Enter Choice: ");
    scanf("%d", &choice);

    switch(choice) {
        case 1:  // Circle
            {
                float radius;
                printf("Enter Radius: ");
                scanf("%f", &radius);
                area = M_PI * radius * radius;  // Area of circle = pi * r^2
                printf("Area of Circle: %.3f\n", area);
                break;
            }
        case 2:  // Square
            {
                float side;
                printf("Enter One Side: ");
                scanf("%f", &side);
                area = side * side;  // Area of square = side^2
                printf("Area of Square: %.2f\n", area);
                break;
            }
        case 3:  // Rectangle
            {
                float length, width;
                printf("Enter Length: ");
                scanf("%f", &length);
                printf("Enter Width: ");
                scanf("%f", &width);
                area = length * width;  // Area of rectangle = length * width
                printf("Area of Rectangle: %.2f\n", area);
                break;
            }
        default:
        printf("Invalid choice!\n");
    }
    return 0;
}
```

2.c:-

```
#include <stdio.h>
int main() {
    int num, reversedNum = 0, remainder;
    printf("Enter a number: ");
    scanf("%d", &num);
    while (num != 0) {
        remainder = num % 10;  // Get the last digit
        reversedNum = reversedNum * 10 + remainder;  // Build the reversed number
        num /= 10;  // Remove the last digit
    }
    printf("Mirror image: %d\n", reversedNum);
        return 0;
}
```

Spring-25(batch 43)



1. *a)* **Explain** the output of the following expressions.

(i)

```
int x = 5;
int y = x++ + 3;
printf("%d", y);
```

(ii)

```
x = (5 < 8 && 15 > 8) || 5 == 5;
printf("%d", x);
```

Output_(i):-

8

Explanation

Post-increment uses the old value of x before increasing it.

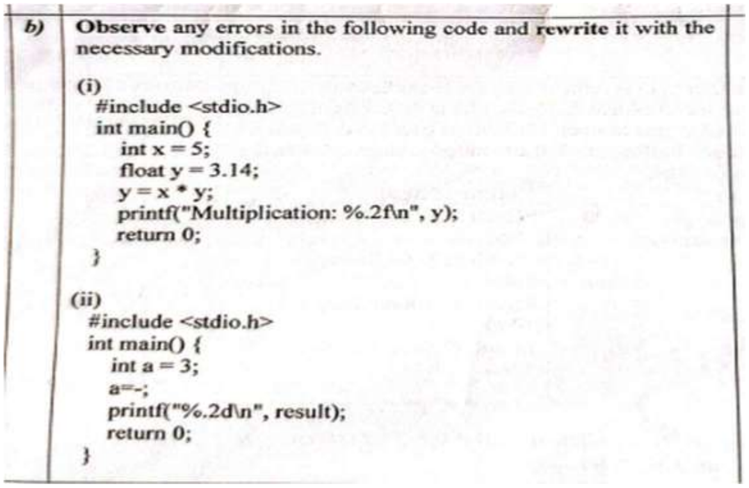
Output_(ii):-

1

Explanation:

1. 5 < 8 → true → 1
2. 15 > 8 → true → 1
   → (5 < 8 && 15 > 8) = 1 && 1 → true (1)
3. 5 == 5 → true (1)
4. So the full expression: (1) || (1) → true (1)

Therefore, x = 1.



b) **Observe** any errors in the following code and **rewrite** it with the necessary modifications.

(i)

```
#include <stdio.h>
int main() {
    int x = 5;
    float y = 3.14;
    y = x * y;
    printf("Multiplication: %.2f\n", y);
    return 0;
}
```

(ii)

```
#include <stdio.h>
int main() {
    int a = 3;
    a=-;
    printf("%.2d\n", result);
    return 0;
}
```

b-(i):-

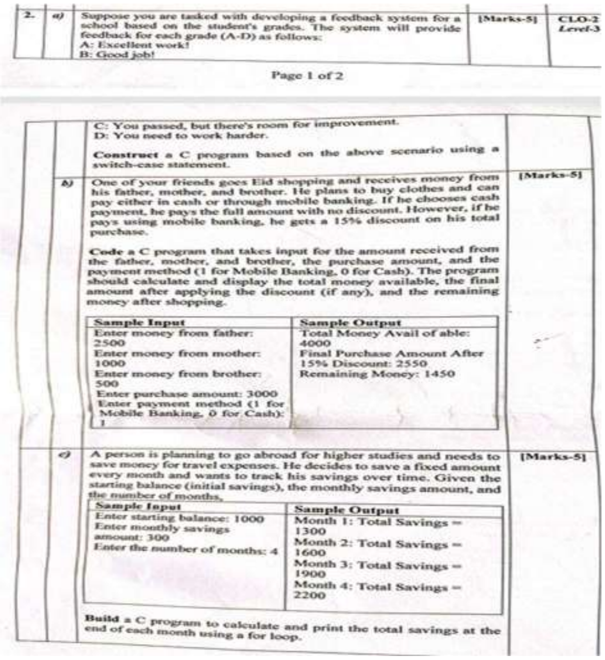
#include <stdio.h>

int main() {

```
    int x = 5;

    float y = 3.14;

    float result;

    result = x * y;

    printf("Multiplication: %.2f\n", result);

    return 0;

}
```

b-(ii):-

```
#include <stdio.h>

int main() {

    int a = 3;

    int result;

    result = a - 1;

    printf("%d\n", result);

    return 0;

}
```



2. a) Suppose you are tasked with developing a feedback system for a school based on the student's grades. The system will provide feedback for each grade (A-D) as follows: [Marks-5] CLO-2 *Level-3*

A: Excellent work!
B: Good job!

Page 1 of 2

C: You passed, but there's room for improvement.
D: You need to work harder.

Construct a C program based on the above scenario using a switch-case statement.

b) One of your friends goes Eid shopping and receives money from his father, mother, and brother. He plans to buy clothes and can pay either in cash or through mobile banking. If he chooses cash payment, he pays the full amount with no discount. However, if he pays using mobile banking, he gets a 15% discount on his total purchase. [Marks-5]

Code a C program that takes input for the amount received from the father, mother, and brother, the purchase amount, and the payment method (1 for Mobile Banking, 0 for Cash). The program should calculate and display the total money available, the final amount after applying the discount (if any), and the remaining money after shopping.

Sample Input	Sample Output
Enter money from father: 2500 Enter money from mother: 1000 Enter money from brother: 500 Enter purchase amount: 3000 Enter payment method (1 for Mobile Banking, 0 for Cash): 1	Total Money Avail of able: 4000 Final Purchase Amount After 15% Discount: 2550 Remaining Money: 1450

c) A person is planning to go abroad for higher studies and needs to save money for travel expenses. He decides to save a fixed amount every month and wants to track his savings over time. Given the starting balance (initial savings), the monthly savings amount, and the number of months, [Marks-5]

Sample Input	Sample Output
Enter starting balance: 1000 Enter monthly savings amount: 300 Enter the number of months: 4	Month 1: Total Savings = 1300 Month 2: Total Savings = 1600 Month 3: Total Savings = 1900 Month 4: Total Savings = 2200

Build a C program to calculate and print the total savings at the end of each month using a for loop.

2-(a):-

```
#include <stdio.h>
int main() {
    char grade;
    printf("Enter your grade (A-D): ");
    scanf(" %c", &grade);

    switch (grade) {
        case 'A':
            printf("Excellent work!\n");
            break;
        case 'B':
            printf("Good job!\n");
            break;
        case 'C':
            printf("You passed, but there's room for improvement.\n");
            break;
        case 'D':
            printf("You need to work harder.\n");
            break;
        default:
            printf("Invalid grade entered.\n");
    }
    return 0;
}
```

2-(b):-

```
#include <stdio.h>
int main() {
    int father, mother, brother, purchase, payment;
    float total, finalAmount, remaining;

    printf("Enter money from father: ");
    scanf("%d", &father);

    printf("Enter money from mother: ");
    scanf("%d", &mother);

    printf("Enter money from brother: ");
    scanf("%d", &brother);

    printf("Enter purchase amount: ");
    scanf("%d", &purchase);

    printf("Enter payment method (1 for Mobile Banking, 0 for Cash): ");
    scanf("%d", &payment);

    total = father + mother + brother;
    if (payment == 1)
        finalAmount = purchase * 0.85;
    else
        finalAmount = purchase;
    remaining = total - finalAmount;
    printf("\nTotal Money Available: %.0f", total);
    printf("\nFinal Purchase Amount After Discount: %.0f", finalAmount);
    printf("\nRemaining Money: %.0f\n", remaining);
    return 0;
}
```

2-(c):-

```
#include <stdio.h>

int main() {
    int i, months;
    float savings, monthly;

    printf("Enter starting balance: ");
    scanf("%f", &savings);

    printf("Enter monthly savings amount: ");
    scanf("%f", &monthly);

    printf("Enter number of months: ");
    scanf("%d", &months);

    for (i = 1; i <= months; i++) {
        savings += monthly;
        printf("Month %d: Total Savings = %.0f\n", i, savings);
    }
```

```
   return 0;
}
```

Daffodil International University
Department of Software Engineering
Faculty of Science & Information Technology
Midterm Examination, Fall 2024
Course Code: SE 121; Course Title: Structured Programming
Sections & Teachers: ALL

Time: 1 Hour 30 Mins **Marks: 25**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1. ***a)*** **Define** header files in C programming. **Describe** the rules for valid and invalid identifiers in C with examples. [Marks-5] CLO-1 *Level-1*

b) Consider the following C program, [Marks-5] CLO-1 *Level-2*

```
#include <stdio.h>
int main() {
   int a = 15, b = 7, c;
   c = (a - b) / 2 + b;
   printf("The value of c: %d\n", c);

   a *= 3;
   printf("The value of a: %d\n", a);

   a++;
   c = a % (b + 5) / b + b;
   printf("The updated value of c: %d\n", c);
   return 0;
}
```

Interpret the output of the program and **explain** how the arithmetic and assignment operators are applied in the calculations.

2. ***a)*** **Examine** the following code for any errors and rewrite the code with the necessary corrections. [Marks-5] CLO-2 *Level-3*

```
#include <stdio.h>
int main() {
   int number sum = 0;
   printf("Enter a number: ");
   scanf("%d", number);

   while (number =! 0)
      sum += number % 10;
```

```
        number /= 10;
    }

    printf("Sum of digits is: %d/n", sum);
    return 0
}
```

b) A school is interested in helping students learn about even and odd numbers. The objective is to calculate the sum of all even numbers and the sum of all odd numbers within a given range defined by two positive integers. Refer to the sample input and output below for a clearer understanding. [Marks-5]

Sample Input:	**Sample Output:**
5 9	The sum of Even Numbers: 14 The sum of Odd Numbers: 21

Articulate the steps required to build the program and solve this logic using the C programming language.

c) **Construct** a C program based on the following scenario, [Marks-5]

In rural Bangladesh, people often calculate how much money they need to save depending on the number of animals they own, like cows or chickens. Different types of animals require different amounts of savings for their care.

Your task is to create a simple system that calculates the total savings needed based on the number of animals and their respective care costs.

- If the user owns **1 cow** - Add 5000 taka to the savings.
- If the user owns **2 cows** - Add 10,000 taka to the savings.
- If the user owns **3 or more cows** - Add 15,000 taka to the savings.
- If the user owns **1 chicken** - Add 100 taka to the savings.
- If the user owns **2 chickens** - Add 200 taka to the savings.
- If the user owns **3 or more chickens** - Add 300 taka to the savings.

If the input is invalid, such as a negative number of animals, display: "Please enter a valid number."

Sample Input:	**Sample Output:**
2 5	Total savings: 10300

Explanation: For the first sample, the first input is 2 which is the number of cows and the second input is 5 which is the number of chickens.
2 cows = 10,000 taka
5 chickens = 300 taka
So total savings are = (10000 + 300) = 10300

Solution of Batch 42 Fall-2024 Mid Exam

1.(a) **Header files** in C contain declarations of functions, variables and data types used in a program. They allow code reuse and modular programming, and serves as an interface.

Example:

#include <stdio.h>   // standard input-output functions

#include <math.h>   // mathematical functions

#include<string.h> //string functions

Rules for Identifiers in C:

1. Must begin with a letter or an underscore (_).
2. Can contain letters, digits, and underscores.
3. No spaces or special symbols (e.g., @, #, $) allowed.
4. Case-sensitive (Sum ≠ sum).
5. Cannot be a keyword (e.g., int, return).

Examples:
Valid: student_name, a1, _totalMarks

Invalid:
1total (starts with digit)
total marks (space not allowed)
int (keyword)

1.(b) Step-by-Step Execution:

Initial values: a = 15, b = 7

Step 1:
c = (a - b) / 2 + b
= (15 - 7) / 2 + 7
= 8 / 2 + 7
= 4 + 7
➞ c = 11

Output:
The value of c: 11

Step 2:
a *= 3 → a = 15 * 3 = 45
Output:
The value of a: 45

Step 3:
a++ → a = 46

c = a % (b + 5) / 4 + b
= 46 % 12 / 4 + 7
= 10 / 4 + 7 (since 46 % 12 = 10)
= 2 + 7 (integer division)
➞ c = 9

Output:
The updated value of c: 9

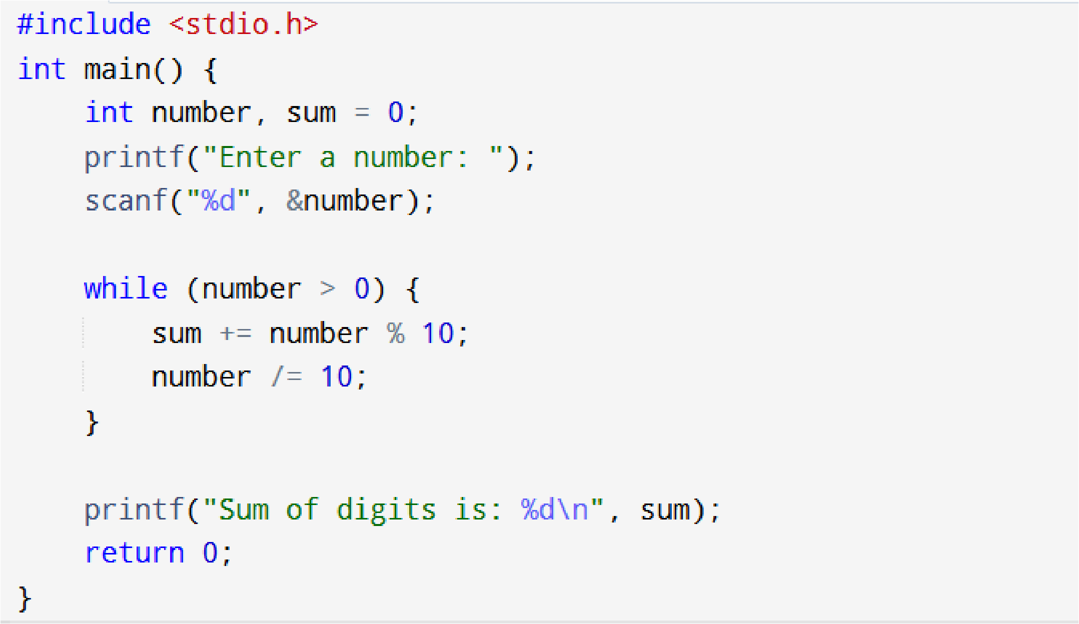
Final output:

The value of c: 11

The value of a: 45

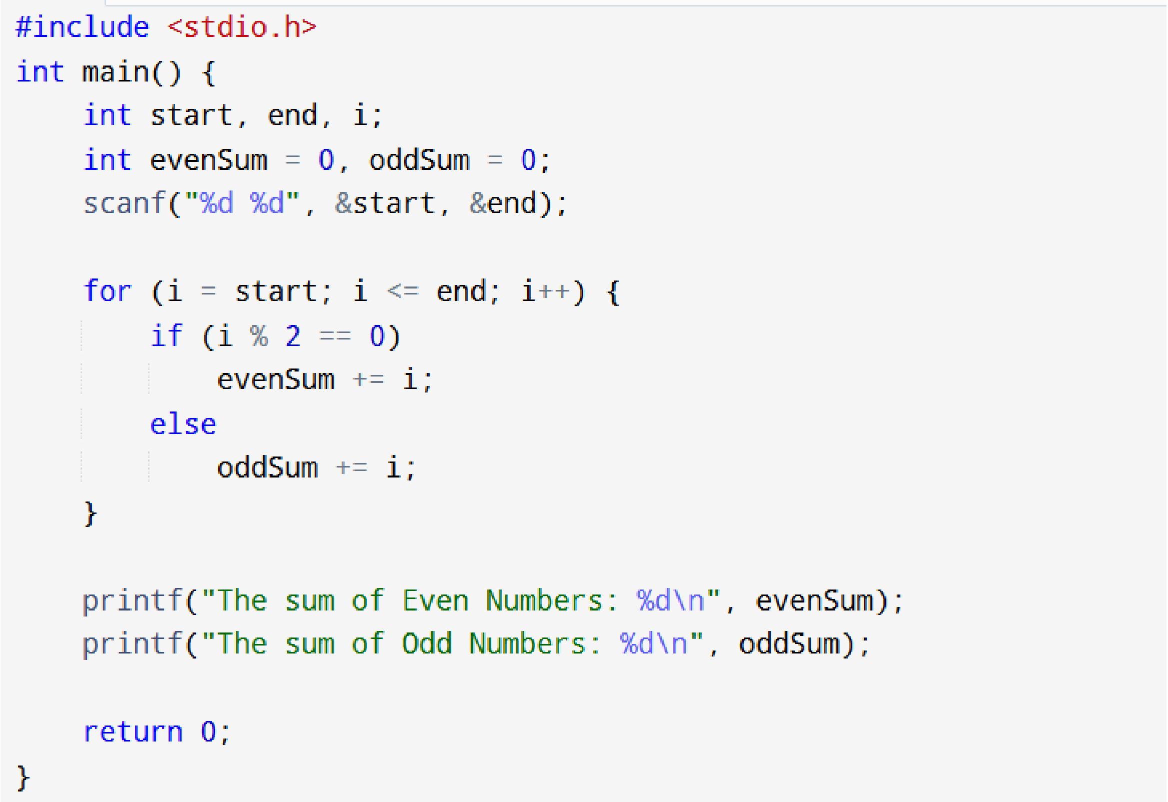
The updated value of c: 9

2.(a) Coreected code:

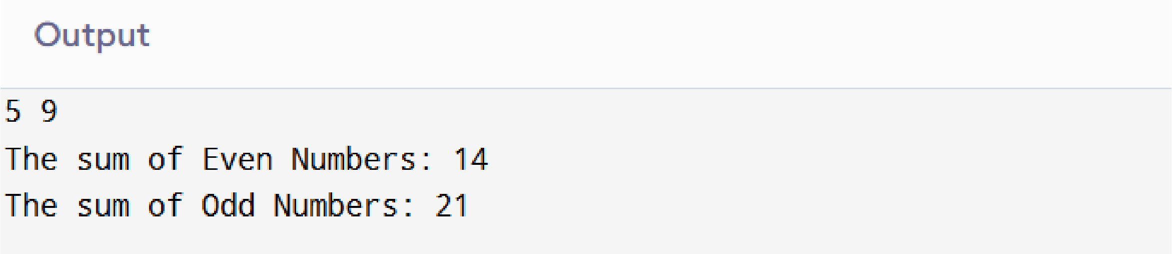


```
#include <stdio.h>
int main() {
    int number, sum = 0;
    printf("Enter a number: ");
    scanf("%d", &number);

    while (number > 0) {
        sum += number % 10;
        number /= 10;
    }

    printf("Sum of digits is: %d\n", sum);
    return 0;
}
```

2.(b)



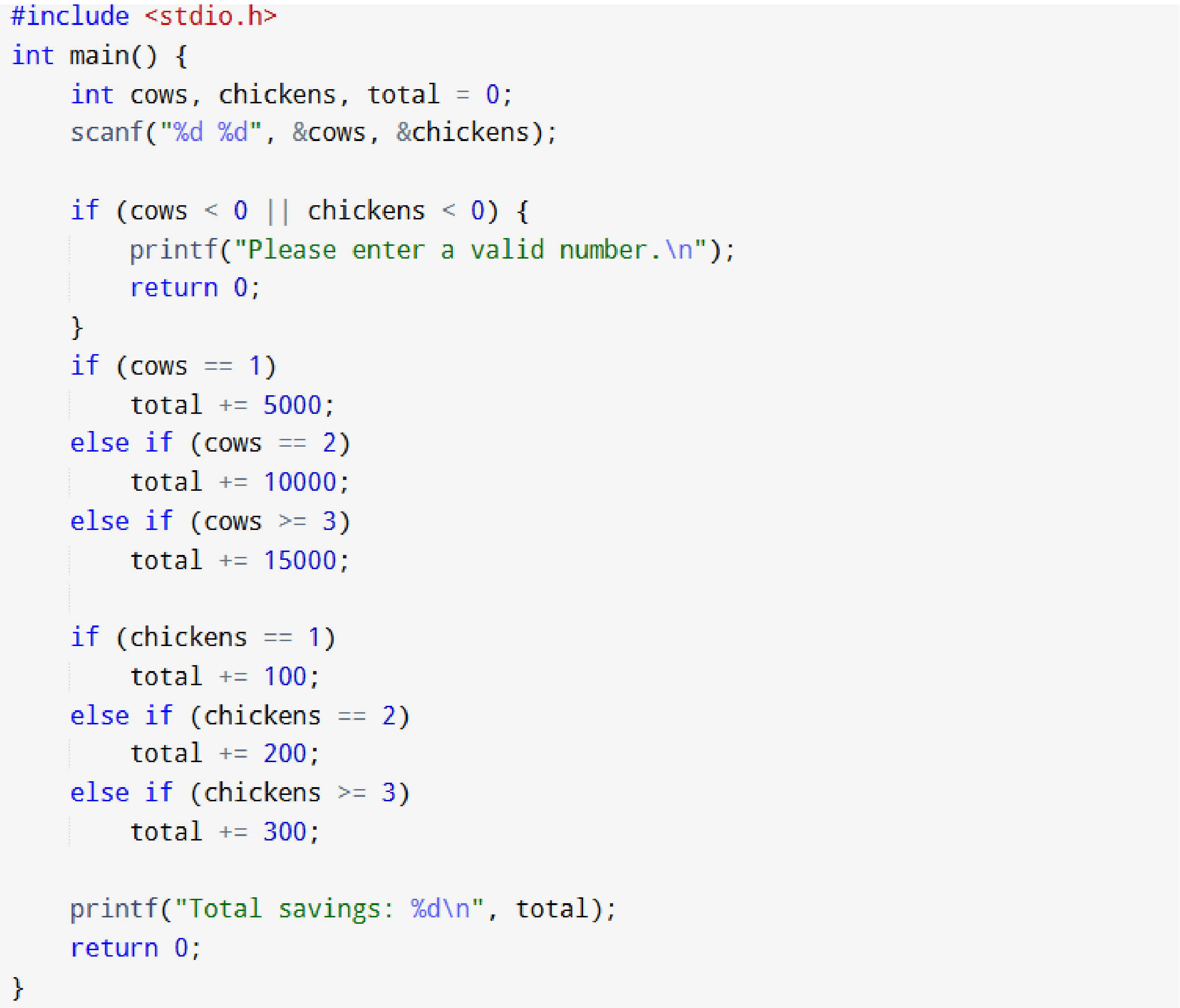
```c
#include <stdio.h>
int main() {
    int start, end, i;
    int evenSum = 0, oddSum = 0;
    scanf("%d %d", &start, &end);

    for (i = start; i <= end; i++) {
        if (i % 2 == 0)
            evenSum += i;
        else
            oddSum += i;
    }

    printf("The sum of Even Numbers: %d\n", evenSum);
    printf("The sum of Odd Numbers: %d\n", oddSum);

    return 0;
}
```



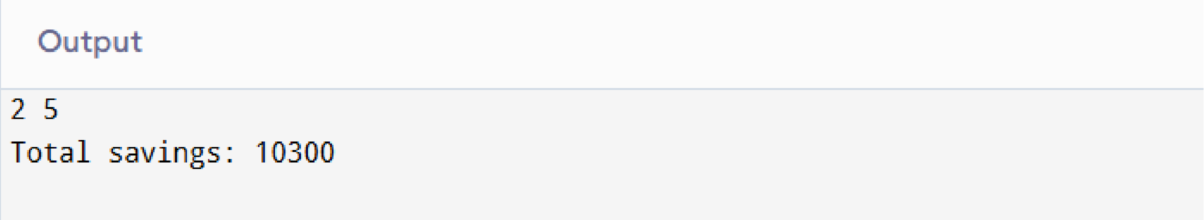
Output

```
5 9
The sum of Even Numbers: 14
The sum of Odd Numbers: 21
```

2.(c) Code:



```c
#include <stdio.h>
int main() {
    int cows, chickens, total = 0;
    scanf("%d %d", &cows, &chickens);

    if (cows < 0 || chickens < 0) {
        printf("Please enter a valid number.\n");
        return 0;
    }
    if (cows == 1)
        total += 5000;
    else if (cows == 2)
        total += 10000;
    else if (cows >= 3)
        total += 15000;

    if (chickens == 1)
        total += 100;
    else if (chickens == 2)
        total += 200;
    else if (chickens >= 3)
        total += 300;

    printf("Total savings: %d\n", total);
    return 0;
}
```



Output

```
2 5
Total savings: 10300
```