Section 2 Autumn-2024

PDS Lab **Lab-3** **20.08.2024**

Instructions:

- This lab is based on the topics: Condition checking and branching.
- You should save each program with the file name as specified against each problem as <Lab#>-<Assignment#>-<Roll#>.c. For example, **03-01-24NA10006.c** to save Program to 1st assignment in Lab 3 with Roll Number 24NA10006
- You should upload each program to the Moodle system. Also, copy + paste your programs to the text window on the test page.
- A few test cases against each problem is given for your references and but not limited to.
- There are three problems and the maximum time allowed is 120 minutes.
- **Do not use loop, and arrays, etc. in this lab.**

1. **Write a C program that takes three integers as input: day, month, and year. The program should check whether the input represents a valid date in the Gregorian calendar. If the date is valid, print "Valid date", otherwise print "Invalid date". Input will be in the format DD MM YYYY (with spaces)**
   For example:
   20 8 2024
   as a date 20th August in the year 2024 in the Gregorian calendar.

   Hint: Consider leap years and the number of days in each month.

   Test cases:

#	INPUT	OUTPUT
1	29 2 2020	Valid date
2	29 2 2019	Invalid date
3	31 4 2021	Invalid date
4	8 20 2024	Invalid date

```c
// Program to check if the given date is valid or not
// Code creator: Nishkal Prakash
// Roll No: 19CS91R05

#include <stdio.h>

int main(){
    int day, month, year;
    // Taking input from the user
    scanf("%d%d%d", &day, &month, &year);

    // Checking if the given date is valid or not
    if (year < 0){
        // If the year is negative, then it is invalid
        printf("Invalid date");
        return 0;
    }

    if (month < 1 || month > 12){
        // If the month is not in the range of 1 to 12, then it is invalid
        printf("Invalid date");
        return 0;
    }

    if (day < 1){
        // If the day is less than 1, then it is invalid
        printf("Invalid date");
        return 0;
    }

    if (month == 2){
        // If the month is February, then we need to check if it is a leap year
or not
        if (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0)){
            // If it is a leap year, then the year is divisible by 4 and not
divisible by 100 or divisible by 400
            if (day > 29){
                // If the day is greater than 29, then it is invalid
                printf("Invalid date");
                return 0;
            }
        } else {
            // If it is not a leap year, then the maximum days in February is 28
            if (day > 28){
                printf("Invalid date");
                return 0;
            }
        }
    } else if (month == 4 || month == 6 || month == 9 || month == 11){
        // If the month is April, June, September or November, then the maximum
days is 30
        if (day > 30){
            printf("Invalid date");
            return 0;
        }
    } else {
        // For all other months, the maximum days is 31
        if (day > 31){
            printf("Invalid date");
            return 0;
        }
    }
    // If the date is valid, then print "Valid date"
    printf("Valid date");
    return 0;
}
```

2. **Write a C program to create a menu for a bank management system. The menu should include four options as given below.**

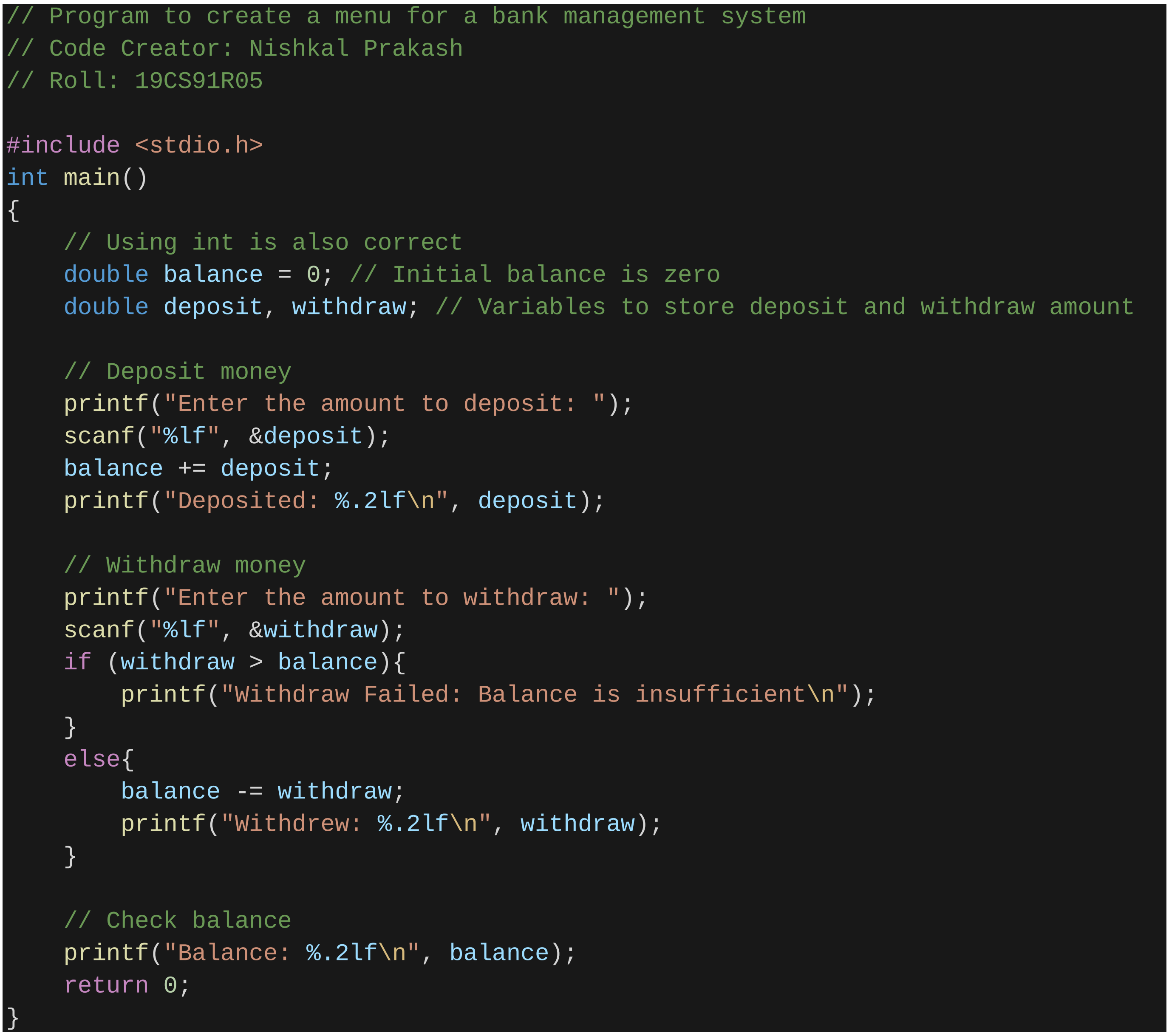
    a. **Deposit money**
    b. **Withdraw money**
    c. **Check balance**
    d. **Exit**

The following constraints must be met by the C program:

- **The initial balance is zero.**
- **If the user tries to withdraw more money than is available in her account, print an error message “Balance is insufficient”.**
- **The program should execute all the options in the following order:**
  1. **Deposit money**
  2. **Withdraw money**
  3. **Check balance**

Test cases:

#	INPUT	OUTPUT
1	Deposit: 1000 Withdraw: 500	Deposited: 1000 Withdrew: 500 Balance: 500
2	Deposit: 500 Withdraw: 600	Deposited: 500 Withdraw Failed: Balance is insufficient Balance 500
3	Deposit: 500 Withdraw: 500	Deposited: 500 Withdrew: 500 Balance: 0



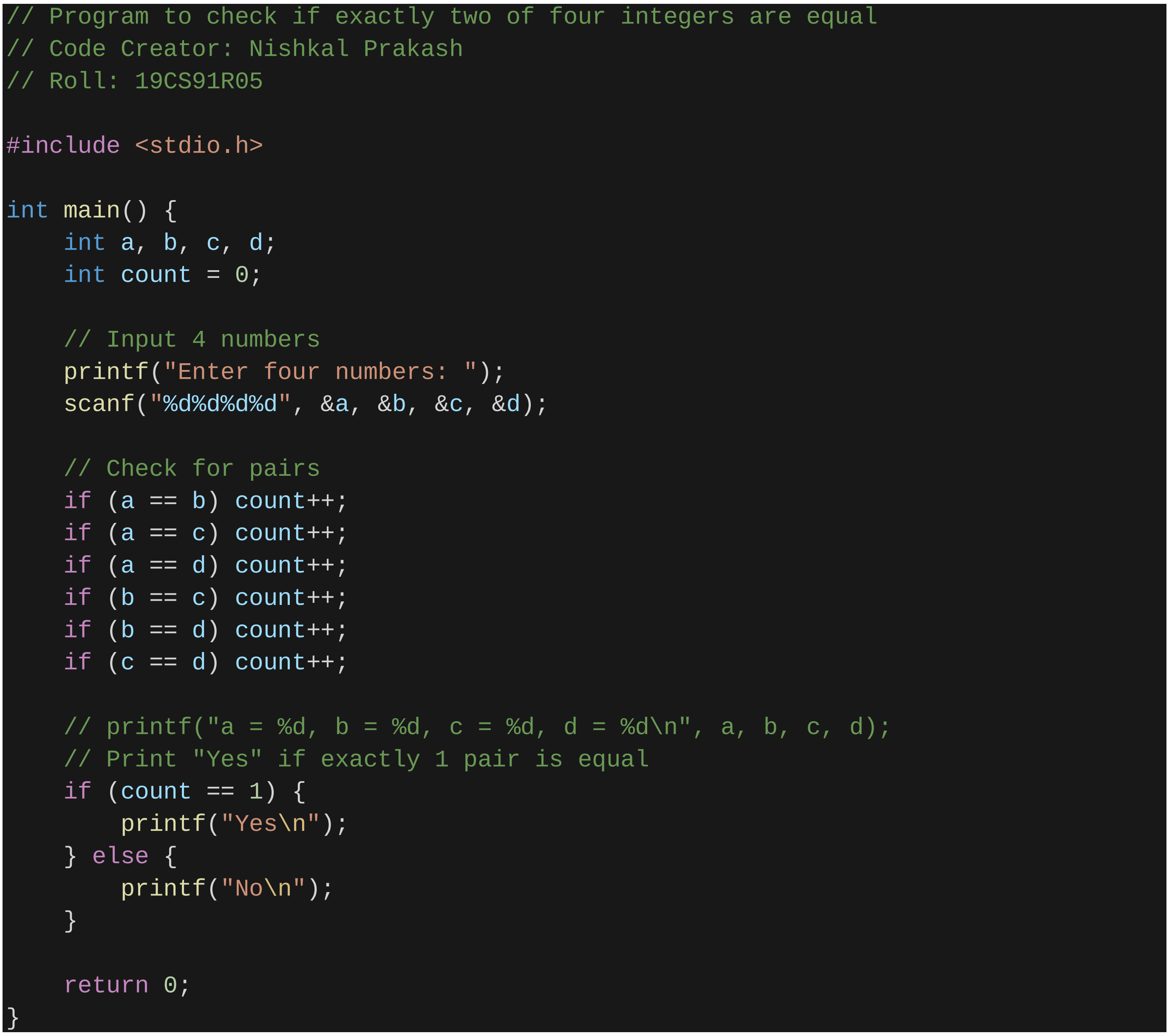
```c
// Program to create a menu for a bank management system
// Code Creator: Nishkal Prakash
// Roll: 19CS91R05

#include <stdio.h>
int main()
{
    // Using int is also correct
    double balance = 0; // Initial balance is zero
    double deposit, withdraw; // Variables to store deposit and withdraw amount

    // Deposit money
    printf("Enter the amount to deposit: ");
    scanf("%lf", &deposit);
    balance += deposit;
    printf("Deposited: %.2lf\n", deposit);

    // Withdraw money
    printf("Enter the amount to withdraw: ");
    scanf("%lf", &withdraw);
    if (withdraw > balance){
        printf("Withdraw Failed: Balance is insufficient\n");
    }
    else{
        balance -= withdraw;
        printf("Withdrew: %.2lf\n", withdraw);
    }

    // Check balance
    printf("Balance: %.2lf\n", balance);
    return 0;
}
```

3. **Write a C program that takes four integers as input and prints "YES" if exactly two of them are equal, "NO" otherwise.**

   Note: The program should not use any loop or array.

Test cases:

#	INPUT	OUTPUT
1	a = 3, b = 3, c = 5, d = 7	YES
2	a = 3, b = 4, c = 5, d = 6	NO
3	a = 2, b = 2, c = 2, d = 3	NO
4	a = 1, b = 2, c = 3, d = 1	YES



```c
// Program to check if exactly two of four integers are equal
// Code Creator: Nishkal Prakash
// Roll: 19CS91R05

#include <stdio.h>

int main() {
    int a, b, c, d;
    int count = 0;

    // Input 4 numbers
    printf("Enter four numbers: ");
    scanf("%d%d%d%d", &a, &b, &c, &d);

    // Check for pairs
    if (a == b) count++;
    if (a == c) count++;
    if (a == d) count++;
    if (b == c) count++;
    if (b == d) count++;
    if (c == d) count++;

    // printf("a = %d, b = %d, c = %d, d = %d\n", a, b, c, d);
    // Print "Yes" if exactly 1 pair is equal
    if (count == 1) {
        printf("Yes\n");
    } else {
        printf("No\n");
    }

    return 0;
}
```

Marks distribution: 1. [25]   2. [35]. 3. [40]

---*---