

ASSIGNMENT DAY 6 – 25/12

STUDENT GRADE SYSTEM:

```
#include <stdio.h>

#define MAX_GRADES 5

static int total_students = 0;
volatile int external_update = 1;

void process_student(int marks) {
    char grade;

    if (marks >= 90) {
        grade = 'A';
    } else if (marks >= 80) {
        grade = 'B';
    } else if (marks >= 70) {
        grade = 'C';
    } else if (marks >= 60) {
        grade = 'D';
    } else {
        grade = 'F';
    }

    switch (grade) {
        case 'A':
            printf("Grade: A\n");
            break;
```

```
    case 'B':

        printf("Grade: B\n");

        break;

    case 'C':

        printf("Grade: C\n");

        break;

    case 'D':

        printf("Grade: D\n");

        break;

    case 'F':

        printf("Grade: F\n");

        break;

}

total_students++;

}

int main() {

    int marks;

    for (int i = 0; i < MAX_GRADES; i++) {

        if (external_update) {

            printf("Enter marks for student %d: ", i + 1);

            scanf("%d", &marks);

            process_student(marks);

        }

    }

    printf("Total students processed: %d\n", total_students);
```

```
    return 0;
}
```

OUTPUT:

Enter marks for student 1: 67

Grade: D

Enter marks for student 2: 43

Grade: F

Enter marks for student 3: 90

Grade: A

Enter marks for student 4: 76

Grade: C

Enter marks for student 5: 66

Grade: D

Total students processed: 5

PRIME NUMBER FINDER:

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

```
const int N = 100;
```

```
static int prime_count = 0;
```

```
int main() {
```

```
    for (int i = 2; i <= N; i++) {
```

```
        int is_prime = 1;
```

```
        for (int j = 2; j * j <= i; j++) {
```

```
            if (i % j == 0) {
```

```

        is_prime = 0;
        break;
    }
}
if (is_prime) {
    printf("%d ", i);
    prime_count++;
}
}
printf("\nTotal primes: %d\n", prime_count);
return 0;
}

```

OUTPUT:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Total primes: 25

DYNAMINC CALCULATOR:

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

```
static int operation_count = 0;
```

```
const char *operations[] = {"Addition", "Subtraction", "Multiplication", "Division"};
```

```
int main() {
```

```
    int choice;
```

```
    float num1, num2, result;
```

```
do {

    printf("\nMenu:\n");

    printf("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5. Exit\n");

    printf("Enter choice: ");

    scanf("%d", &choice);


    if (choice >= 1 && choice <= 4) {

        printf("Enter two numbers: ");

        scanf("%f %f", &num1, &num2);


        switch (choice) {

            case 1: result = num1 + num2; break;

            case 2: result = num1 - num2; break;

            case 3: result = num1 * num2; break;

            case 4:

                if (num2 != 0) result = num1 / num2;

                else {

                    printf("Error! Division by zero.\n");

                    continue;

                }

                break;

        }

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

        operation_count++;

    } else if (choice != 5) {

        printf("Invalid choice. Try again.\n");

    }

} while (choice != 5);


printf("Total operations performed: %d\n", operation_count);
```

```
    return 0;  
}
```

OUTPUT:

Menu:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter choice: 3

Enter two numbers: 8

9

Result: 72.00

Menu:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit

Enter choice: 5

Total operations performed: 1

MATRIX CALCULATION:

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

```
const int MAX_SIZE = 3;

static int result[3][3];

int main() {

    int A[3][3], B[3][3];

    int choice;

    printf("Enter elements for matrix A:\n");
    for (int i = 0; i < MAX_SIZE; i++) {
        for (int j = 0; j < MAX_SIZE; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    printf("Enter elements for matrix B:\n");
    for (int i = 0; i < MAX_SIZE; i++) {
        for (int j = 0; j < MAX_SIZE; j++) {
            scanf("%d", &B[i][j]);
        }
    }

    printf("Choose operation:\n1. Add\n2. Multiply\n");
    scanf("%d", &choice);

    if (choice == 1) {
        for (int i = 0; i < MAX_SIZE; i++) {
            for (int j = 0; j < MAX_SIZE; j++) {
                result[i][j] = A[i][j] + B[i][j];
            }
        }
    }
}
```

```

    printf("Addition result:\n");
} else if (choice == 2) {
    for (int i = 0; i < MAX_SIZE; i++) {
        for (int j = 0; j < MAX_SIZE; j++) {
            result[i][j] = 0;
            for (int k = 0; k < MAX_SIZE; k++) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    printf("Multiplication result:\n");
} else {
    printf("Invalid choice\n");
    return 0;
}

for (int i = 0; i < MAX_SIZE; i++) {
    for (int j = 0; j < MAX_SIZE; j++) {
        printf("%d ", result[i][j]);
    }
    printf("\n");
}

return 0;
}

```

OUTPUT:

Enter elements for matrix A:

3

4

2

6

8

9

4

5

67

Enter elements for matrix B:

4

5

6

8

45

2

3

4

5

Choose operation:

1. Add

2. Multiply

2

Multiplication result:

50 203 36

115 426 97

257 513 369

TEMPERATURE MORNING SYSTEM:

#include <stdio.h>

```
volatile int temperature;

static int max_temp = 0;

int main() {
    while (1) {
        printf("Enter temperature: ");
        scanf("%d", &temperature);

        if (temperature > max_temp) {
            max_temp = temperature;
        }

        if (temperature > 30) {
            printf("Warning: Temperature is too high!\n");
        } else if (temperature < 10) {
            printf("Warning: Temperature is too low!\n");
        } else {
            printf("Temperature is normal.\n");
        }

        printf("Max temperature recorded: %d\n", max_temp);
    }

    return 0;
}
```

OUTPUT:

Enter temperature: 43

Warning: Temperature is too high!

Max temperature recorded: 43

Enter temperature: 12

Temperature is normal.

Max temperature recorded: 43

Enter temperature: 44

Warning: Temperature is too high!

Max temperature recorded: 44

Enter temperature:

PASSWORD VALIDATION SYSTEM:

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

```
#include <string.h>
```

```
const int MAX_ATTEMPTS = 3;
```

```
static int failed_attempts = 0;
```

```
int main() {
```

```
    char password[] = "password123";
```

```
    char input[50];
```

```
    int choice;
```

```
    do {
```

```
        printf("Enter password: ");
```

```
        scanf("%s", input);
```

```
        if (strcmp(input, password) == 0) {
```

```
            printf("Password correct.\n");
```

```
        break;
    } else {
        failed_attempts++;
        printf("Incorrect password.\n");

        if (failed_attempts >= MAX_ATTEMPTS) {
            printf("Maximum attempts reached.\n");
            break;
        }
    }

    printf("Failed attempts: %d\n", failed_attempts);
} while (failed_attempts < MAX_ATTEMPTS);

return 0;
}
```

OUTPUT:

Enter password: ASASDFHT

Incorrect password.

Failed attempts: 1

Enter password: DFXGHDGJGF

Incorrect password.

Failed attempts: 2

Enter password: DFDGFDJYU7

Incorrect password.

Maximum attempts reached.

BANKING SYSTEM:

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

```
const int MAX_WITHDRAWAL = 5000;
```

```
static float balance = 10000;
```

```
int main() {
```

```
    int choice;
```

```
    float amount;
```

```
    do {
```

```
        printf("\nBank Transaction Menu:\n");
```

```
        printf("1. Deposit\n2. Withdraw\n3. Check Balance\n4. Exit\n");
```

```
        printf("Enter your choice: ");
```

```
        scanf("%d", &choice);
```

```
        if (choice == 1) {
```

```
            printf("Enter deposit amount: ");
```

```
            scanf("%f", &amount);
```

```
            balance += amount;
```

```
            printf("Deposited %.2f. New balance: %.2f\n", amount, balance);
```

```
        } else if (choice == 2) {
```

```
            printf("Enter withdrawal amount: ");
```

```
            scanf("%f", &amount);
```

```
            if (amount > MAX_WITHDRAWAL) {
```

```
                printf("Error: Withdrawal exceeds maximum limit of %.2f\n", MAX_WITHDRAWAL);
```

```
            } else if (amount > balance) {
```

```
                printf("Error: Insufficient balance.\n");
```

```
            } else {
```

```
        balance -= amount;

        printf("Withdrawn %.2f. New balance: %.2f\n", amount, balance);
    }
} else if (choice == 3) {
    printf("Current balance: %.2f\n", balance);
} else if (choice != 4) {
    printf("Invalid choice. Try again.\n");
}

} while (choice != 4);

printf("Exiting. Final balance: %.2f\n", balance);
return 0;
}
```

OUTPUT:

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 3

Current balance: 10000.00

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 1

Enter deposit amount: 678

Deposited 678.00. New balance: 10678.00

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: W 2

Enter withdrawal amount: 4000

Withdrawn 4000.00. New balance: 6678.00

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 2

Enter withdrawal amount: 5PP 000

Withdrawn 5000.00. New balance: 1678.00

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 2

Enter withdrawal amount: 679

Withdrawn 679.00. New balance: 999.00

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 50001

Invalid choice. Try again.

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice: 2

Enter withdrawal amount: 3000

Error: Insufficient balance.

Bank Transaction Menu:

1. Deposit
2. Withdraw
3. Check Balance
4. Exit

Enter your choice:

DIGITAL CLOCK SIMULATION:

DOUBT

GAME SCORE TRACKER:

```
#include <stdio.h>

const int WINNING_SCORE = 10;
static int current_score = 0;

int main() {
    int score;

    while (current_score < WINNING_SCORE) {
        printf("Enter score for this round: ");
        scanf("%d", &score);

        current_score += score;

        if (current_score >= WINNING_SCORE) {
            printf("You win! Final score: %d\n", current_score);
        } else {
            printf("Current score: %d\n", current_score);
        }
    }

    return 0;
}
```

OUTPUT:

Enter score for this round: 34

You win! Final score: 34

