



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

CERTIFICATE

This is to certify that

Mr. /Ms BANCHAL YASH ANIL

With enrollment no 24040304000247 from

Semester 1st Div: B has Successfully

completed his/her laboratory experiments in the **Structure**

Programming (4040243102) from the department of

BACHELOR OF COMPUTER SCIENCE & IT during the

academic year 2024 - 25

Date of Submission: 28/11/2024

Staff In charge: (Signature)

Head of Department



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 1

(A) Write simple programs and observe the differences in their execution using a compiler and an interpreter. Compare factors like speed, memory usage, and portability.

```
// Simple_Program.C
#include <stdio.h>

int factorial (int n) {
    if (n == 0) {
        return 1;
    } else {
        return n * factorial (n - 1);
    }
}

int main () {
    int num = 5; // Change this number to test with
    different input
    printf ("The factorial of %d is %d\n", num, factorial
    (num));
    return 0;
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

```
# simple_program.py
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
if __name__ == "__main__":
    num = 5 # Change this number to test with
    different inputs print("The factorial
    of {num} is {factorial(num)}")
```

Conclusion:

C Execution (Compiler)

1) Save the code as simple_program.c.

2) Compile it using a compiler (like gcc).

gcc simple_program.c -o simple_program



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

(B) Write programs in assembly language to perform basic tasks like arithmetic operations, conditional branching, and memory manipulation. Translate these programs into machine code and execute them.

Section .data

```
num1 dw 5      ; First number
num2 dw 10     ; Second number
result dw 0      ; To store the result of addition
threshold dw 7    ; Threshold to check against
message db "The result is : ", 0 ; Null-terminated String
```

Section .bss

```
temp resw 1      ; Temporary storage
```

Section .text

```
global _start
```

```
_start
```

; 1. Addition of two numbers

```
    mov ax, [num1] ; Load num1 into AX
```

```
    add ax, [num2] ; Add num2 to AX
```

```
    mov [result], ax ; Store result
```

; 2. Conditional branch

```
    cmp ax, [threshold] ; Compare result with threshold
```

```
    jg above_threshold ; If greater, jump to above_threshold
```

; 3. If not above threshold

```
    mov [temp], 0 ; Set temp to 0 (not above threshold)
```

```
    jmp end_procedure
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

above_threshold
nov[Temp], 1

; Set temp to 2 (above threshold)

end-procedure;

; Exit the program

mov eax, 2

`xor ebx, ebx`

~~int 0x80~~

; syscall : exit

; exit code 0

```
#include <stdio.h>
#include<stdio.h>
int main(){
    printf ("Running assembly program ...\\n");
    // Runs the Assembly Program
    int return_code = system("./example");
    if (return_code == 1) {
        perror ("Error running assembly program");
        return 1;
    }
    return 0;
}
```

Conclusion:

This example demonstrate basic assembly language concepts and how to integrate assembly with C for execution. Adjust the program's complexity or functionality based on learning objectives or tasks at hand



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Q. Declare and initialize variables of different data types (int, float, char) and print them

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

```
int main() {
```

```
    // Declare and initialize variable
```

```
    int integerVar = 42; // Integer variable
```

```
    float floatVar = 3.14f; // Float variable
```

```
    char charVar = 'A'; // Character variable
```

```
    // Print the values of the variables
```

```
    printf("Integer value : %d\n", integerVar); // %d format specifier for integer
```

```
    printf("float value %.2f\n", floatVar); // %.2f for floating-point values with 2 decimal precision
```

```
    printf("Character Value %c\n", charVar); // %c for character values
```

```
    return 0;
```

```
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Expected Output

Integer value : 42

Float value : 3.14

Character value : A

Conclusion:

The C program successfully declares and initializes variables of different data types int, float and char. It utilizes the printf function to display their values using appropriate format specifies for each data types. The output conform that the variables have been correctly assigned and printed, demonstrating basic variable handling in C.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 2

(A) Perform addition, subtraction, multiplication, and division on two numbers.

```
#include <iostream.h>
#include <conio.h>

void main(){
    clrscr(); //clears the screen
    float num1, num2
    float sum, difference, product, quotient;
    // Input two numbers
    cout << "Enter first number:";
    cin >> num1
    cout << "Enter Second number:";
    cin >> num2;

    // Perform operations
    sum = num1 + num2
    difference = num1 - num2
    product = num1 * num2
    // Check for division by zero
    if (num2 != 0) {
        quotient = num1 / num2;
        cout << "Quotient : " << quotient << endl;
    } else {
        cout << "Cannot divide by zero" <<
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

```
end;  
}  
// Display result  
cout << "Sum" << sum << endl;  
Count << "Difference" << difference << endl;  
getch(); // Wait for a key press  
before closing  
}
```

Conclusion:

Output would be

Sum : 15
Difference : 5
Product : 50
Quotient : 2



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

(B) Input marks for multiple students in different subjects.

Calculate total marks and average marks for each student.

Assign grades based on the following criteria

A: 90% and above

B: 80% - 89%

C: 70% - 79%

D: 60% - 69%

E: Below 60%

Display the results including student details, total marks, average marks, and grade.

```
#include <iostream.h>
#include <Stdio.h>
#define Max_Student 100
#define Max_Student_Subject 5

void calculate_Grades (float averages);

int main()
{
    int num_Students, num_subjects;
    float marks [MAX_STUDENTS]
    [MAX_SUBJECTS];
    float total [MAX_STUDENTS];
    float average [MAX_STUDENTS];

    printf ("Enter the number of subjects: ");
    scanf ("%d", &num_subjects);
    //input marks for each students
    for (int i = 0; i < num_Students; i++) {
        total[i] = 0; //Initialize total marks
        printf ("Enter marks for Student %d", i + 1);
        for (int j = 0; j < num_subjects; j++) {
            printf ("Enter marks for Subject %d", j + 1);
            scanf ("%f", &marks[i][j]);
            total[i] += marks[i][j];
        }
        average[i] = total[i] / num_subjects;
        if (average[i] >= 90)
            printf ("Grade A\n");
        else if (average[i] >= 80)
            printf ("Grade B\n");
        else if (average[i] >= 70)
            printf ("Grade C\n");
        else if (average[i] >= 60)
            printf ("Grade D\n");
        else
            printf ("Grade E\n");
    }
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

\n", i+1);

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

 printf ("Subject %d: ", i+1);

 scanf ("%f", &marks[i][j]);

 total[i] += marks[i][j]; // calculate

 total

// Calculate average

 average[i] = total[i] / numSubjects;

}

// Display the result

 printf ("\n Student Results :\n");

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

 printf ("\n Student %d: \n", i+1);

 printf ("Total Marks : %2f\n", total[i]);

 printf ("Average Marks : %2f\n", average[i]);



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

printf("Grade:");

Calculate Grades (average[i]);

}

return 0;

}

void calculate Grades (float average) {

if (average >= 90) {

printf ("A\n");

} elseif (average >= 80) {

printf ("B\n");

} else if (average >= 70) {

printf ("C\n");

} elseif (average >= 60) {

printf ("D\n");

} else {

printf ("F\n");

}



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Conclusion:

The program display the total marks average marks and corresponding grade for each student.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 3

a) Write a program to convert a floating-point number to an integer and observe the behavior.

```
#include <stdio.h>

int main () {
    float floatNumber;
    int intNumber;
    // Input a floating - point number
    printf ("Enter a floating - point number: ");
    scanf ("%f", &floatNumber);

    // Convert the floating - point numbers to an
    // integer
    int gerIntNumber = (int) floatNumber;
    // Cast the float to int
    // Display the result
    printf ("%f\n", gerIntNumber);
    PointF ("Converting integers numbers: %d\n", gerIntNumber);
    return 0;
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Output

Enter a floating-point number : 12.35

Floating-point number : 12.35

Converted Integer number is

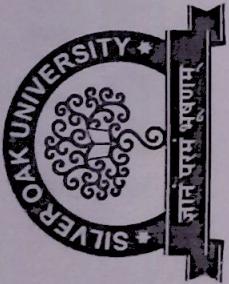
Enter a floating-point number : 3.99

Floating-point number : 3.99

Converted integer number is 3

Conclusion:

The fraction part of the floating-point number is truncated and not rounded. For example, 12.35 becomes 12 because 12 and .35 becomes 12.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

B) Experiment with different arithmetic operations involving operands of different data types to observe implicit type conversion.

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

```
int main()
```

```
{  
    int iVar = 10;  
    float fVar = 5.5;  
    double dVar = 12.34;  
    char cVar = 'A'; // ASCII value of 'A' is 65  
    // Addition  
    printf("int+float : %f\n", iVar + floatVar);  
    printf("int+double : %lf\n", iVar + doubleVar);  
    printf("char + int : %d\n", charVar + iVar);  
    // Subtraction  
    printf("double - float : %f\n", doubleVar -  
          floatVar);  
    printf("int - char : %d\n", iVar - charVar);  
    // Multiplication  
    printf("int * float : %f\n", iVar * floatVar);  
    printf("double * double : %f\n", doubleVar * charVar);  
    // Division  
    printf("float/int : %f\n", floatVar / iVar);  
    printf("double/int : %lf\n", doubleVar / iVar);  
    // Modulus (remainder)  
    printf("int % char : %d\n", iVar % charVar);  
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

return();
}

Conclusion:

- Addition → Shows how an integer is implicitly converted to a float during the operation
- Subtraction : Demonstrates conversion when subtracting a char from an integer
- Multiplication and Division : Covers implicit type conversion between integers, float and double
- modulus : Operates between integers and char values, demonstrating scaling implicit conversion of the char.



SILVER OAK UNIVERSITY

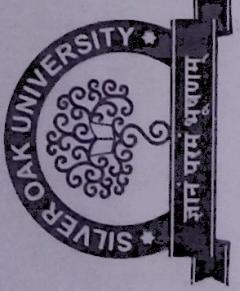
EDUCATION TO INNOVATION

c) Explore the effects of type conversion on the precision and range of values.

```
#include <Stdio.h>
#include <limits.h>
#include <float.h>

void displayRanges() {
    printf ("Range of datatypes:\n");
    printf ("int : %d to %d\n", INT_MIN, INT_MAX);
    printf ("short : %d to %d\n", SHRT_MIN, SHRT_MAX);
    printf ("long : %ld to %ld\n", LONG_MIN, LONG_MAX);
    printf ("float : %e to %e\n", FLT_MIN, FLT_MAX);
    printf ("double : %e to %e\n", DBL_MIN, DBL_MAX);
    printf ("\n");
}

int main () {
    displayRanges ();
    int intVar= 123456789;
    Short ShortVar;
    long longVar;
    float floatVar;
    double doubleVar;
    char charVar = 'B';
    float Var = intVar;
    double Var = floatVar;
    short Var = intVar;
    short Var = floatVar;
    long Var = intVar;
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

```
Pointe ("Original Int value : %d\n", intVar);
Pointe ("Implicit conversion to float : %f\n", floatVar);
Pointe ("Implicit conversion to double : %lf\n", doubleVar);
Pointe ("Implicit Conversion to short : %d\n", shortVar);
Pointe ("Implicit Conversion to long : %ld\n", longVar);

PrintF ("\nCombining different types in arithmetic operation\n");
PrintF ("%int + double : %f\n", intVar + preciseDouble);
PrintF ("float + char : %f\n", PreciseFloat + CharVar);

PointF ("double + char : %f\n", preciseDouble + charVar);
PrintF
Return 0;
}
```

Conclusion:

Range Display - Display Range
Converting int to float
Convert int to double
Convert int to short
Converting intro long



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

d) Write a program that performs explicit type conversion between different data types and verifies the results.

```
if (floatVar != doubleVar) {
    printf ("Conversion from double to float resulted in precision
loss .\n");
} else {
    printf ("No precision loss in conversion from double
to float.\n");
}

if ((double)intVar != doubleVar) {
    printf ("Conversion from double to int resulted in precision
loss due to truncation.\n");
} else {
    printf ("No precision loss in conversion from int to
char.\n");
}

return 0;
}
```

Conclusion:

This code explanation:

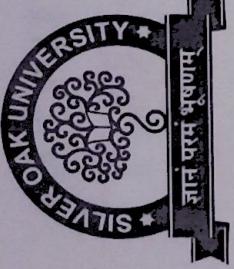
- 1) Explicit Conversion:
 - The doubleVar is explicitly converted, int and char using casting.
 - Each conversion is then printed to verify the results.

2 Verification :-

→ The program checks if the value of float var differs from double var to detect any precision loss.

It converts int var back to double
to verify if truncation occurred
during the conversion

Similarly, it checks if char var
matches int var to verify the precision



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 4

- a) write a program to determine if a number is positive, negative, or zero using if, else if, and else statements.

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

```
int main () {
```

```
    int number;
```

```
    char choice;
```

```
    do {
```

```
        // Ask the user to enter a number
```

```
        printf ("Please enter an integer: ");
```

```
        Pointf
```

```
        // Input validation
```

```
        while (scanf ("%d", &number) != 1) {
```

```
            printf ("Invalid input. Please enter an integer: ");
```

```
            // Clear the invalid input
```

```
            while (getchar () != '\n');
```

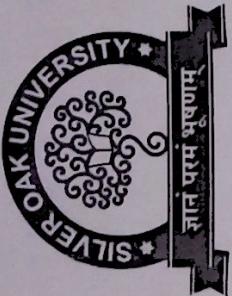
```
        }
```

```
    } // Determine if the number is positive, negative, or zero
```

```
    if (number > 0) {
```

```
        printf ("The number %d is positive.\n", number);
```

```
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

```
else {
    Pointf ("The number is zero.\n");
}

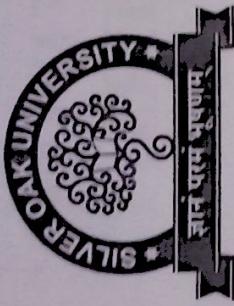
// Ask the user if they want to check another
// number
Pointf ("Would you like to check another number?
        (y/n): ");
scanf ("%c", &choice)

// Input validation for choice
while (choice != 'y' & & choice != 'n') f
    Pointf ("Invalid input. Please enter 'y' for yes or 'n' for
            no");
    scanf ("%c", &choice);
}

while (choice == 'y') {
    Pointf ("Thankyou for using the number checker
Program. Goodbye\n");
    return 0;
}
```

Conclusion:

The first if checks if the number is greater than zero
The else if checks if the number is less than zero
The else handles the case where the number is zero



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

B) Determine if a number is even or odd using if-else statements

```
#include <stdio.h>

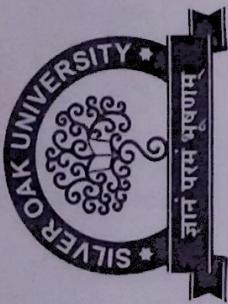
int main() {
    int number;
    printf ("Enter an integer:");
    scanf ("%d", &number);

    if (number % 2 == 0) {
        printf ("%d is even.\n", number);
    } else {
        printf ("%d is odd.\n", number);
    }
    return 0;
}
```

Conclusion:

Example 2:

If the user enters 10, the output will be:-
10 is even



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

c) Create a program that calculates the maximum of three numbers entered by the user using conditional statements.

```
#include <stdio.h>

int main () {
    int num1, num2, num3, max;

    Pointf("Enter three integers:");
    Scanf ("%d %d %d", &num1, &num2, &num3);

    if (num1 > num2) {
        if (num1 >= num3) {
            max = num1;
        } else {
            max = num3;
        }
    } else {
        if (num2 >= num3) {
            max = num2;
        } else {
            max = num3;
        }
    }
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

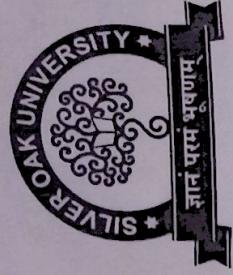
```
maxc = num3;  
}
```

```
}  
Pointf("The maximum number is : %d\n",  
maxc);  
scanf("%d");  
}
```

Conclusion:

Example

Enter three integers : 10 25 7
The maximum number is : 25



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

d) Determine the largest of three numbers using nested if-else.

#include <stdio.h>

```
#include <stdio.h>
int num1, num2, num3, longest;
printf ("Enter three numbers : ");
scanf ("%d %d %d", &num1, &num2, &num3)
if (num1 >= num2) {
    if (num1 >= num3) {
        longest = num1
    } else {
        longest = num2;
    }
} else {
    longest = num3;
}
printf ("The largest number is : %d\n", longest);
```

Point between 0;
Conclusion:

Output:
Enter three number : 25 18 32
The largest number is: 32



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

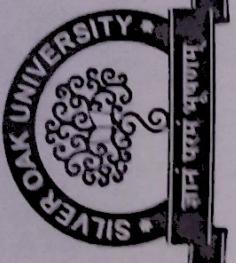
e) Implement a simple login system where the user enters a username and password, and the program validates them using conditional statements.

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

int main () {
    char username [20], password [20];
    int max_attempts = 3;
    int attempts = 0;

    printf ("Welcome to the Login System\n");
    while (attempts < max_attempts) {
        printf ("\nEnter your name : ");
        scanf ("%s", username);
        printf ("\nEnter your password : ");
        scanf ("%s", password);

        if (strcmp (username, "user") == 0 && strcmp
            (password, "password") == 0) {
            printf ("Login successful\n");
            return 0;
        }
    }
    if (attempts < max_attempts) {
        printf ("Invalid username or password\n");
    }
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Point F ("You have %d attempts remaining.\n",
max_attempts

{}

}

}

Point F ("maximum login attempts reached.

Please try again later.\n");

return L;

{}

Conclusion:

Welcome to the Login System
Enter Your Username: wrong_user
Enter your Password: wrong_password
If you did not enter or password
you have 2 attempt remaining
Enter your Username: user
Enter your Password: wrong_password
If you did not enter or password
you have 1 attempt remaining
Enter your Username: user
Enter your Password: password
If you did not enter or password

by error



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Q) Write a program that determines whether a year entered by the user is a leap year or not using conditional statements.

```
#include <stdio.h>
int main()
{
    int year;
    printf("Enter a year:");
    scanf("%d", &year);
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        printf("%d is a leap year.\n", year);
    else
        printf("%d is not a leap year.\n", year);
    return 0;
}
```

Conclusion:

Enter a year : 2024
2024 is a leap year

Enter a year : 2023
2023 is not a leap
year



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

EXPERIMENT NO: 5

a) Write a program that prints numbers from 1 to 10 using each type of loop.

```
#include <stdio.h>
int main() {
    for(int i = 1; i <= 10; i++) {
        printf("%d", i);
    }
    printf("\n")
    int j = 1;
    while (j <= 10) {
        printf("%d", j);
        j++;
    }
    printf("\n");
    int k = 1;
    do {printf("%d", k);
        k++;
    } while (k <= 10);
    printf("\n");
    return 0;
}
```

1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

b) Create a program that calculates the factorial of a number using a for loop.

```
#include <stdio.h>
int main() {
    int num, i, factorial = 1;
    printf ("Enter a non-negative integer:");
    scanf ("%d", &num); // Factorial is not defined for integer
    scanf ("%d", &num);
    if(num < 0) {
        printf ("Factorial is not defined for negative
                number.\n");
    } else {
        for (i=1; i<=num; ++i) {
            factorial *= i;
        }
        printf ("Factorial of
                %d = %d\n", num, factorial);
    }
    return 0;
}
```

Conclusion:

Enter a non-negative integer: 5
Factorial of 5 = 120



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

c) Implement a program to find the sum of digits of a number using a while loop.

```
#include <stdio.h>

int main () {
    int num, sum = 0, digit;
    printf ("Enter a positive integer:");
    scanf ("%d", &num);
    while (num > 0) {
        digit = num % 10; // Extract the last digit
        sum += digit; // Add the digit to the sum
        num /= 10; // Remove the last digit from the number
    }
    printf ("Sum of digits = %d\n", sum);
    return 0;
}
```

Conclusion:

Enter a positive integer: 123456

Sum of digits = 15



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

d) Write a program that prompts the user to enter integers until a negative number is entered, and then calculates the average of the entered numbers using a do-while loop.

```
#include <Stdio.h>
int main(){
    int num, count = 0, sum = 0;
    float average
    do {
        printf("Enter an integer (negative to quit): ");
        scanf("%d", &num);
        if (num >= 0) {
            sum += num;
            count++;
        }
    } while (num >= 0);
    if (count > 0) {
        average = (float)sum / count;
        printf("Average of the entered numbers: %.2f\n", average);
    } else {
        printf("No positive numbers were entered.\n");
    }
}
```

Conclusion:

```
Enter an integer (negative to quit): 10
Enter an integer (negative to quit): 20
Enter an integer (negative to quit): 30
Enter an integer (negative to quit): -1
Average of the entered numbers: 20.00
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

c) Generate Fibonacci series up to N terms.

```
#include <stdio.h>
int main(){
    int n, i, t1 = 0, t2 = 1, nextTerm;
    printf ("Enter the number of terms:");
    scanf ("%d", &n);
    printf ("Fibonacci Series:");
    for (i = 1; i <= n; ++i){
        printf ("%d", t1);
        nextTerm
        = t1 + t2;
        t2 = nextTerm;
    }
    return 0;
}
```

Conclusion:

Entered the number of terms : 10
Fibonacci Series: 0, 1, 2, 3, 5, 8, 13, 21, 34



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

i) Check if a number is a palindrome.

```
#include <stdio.h>
int main() {
    int num, reversedNum = 0, originalNum,
        remainder;
    printf("Enter an integer: ");
    scanf("%d", &num);
    originalNum = num;
    while (num != 0) {
        remainder = num % 10;
        reversedNum = reversedNum * 10 + remainder;
        num /= 10;
    }
    if (originalNum == reversedNum)
        printf("%d is a palindrome.\n", originalNum);
    else
        printf("%d is not a palindrome.\n", originalNum);
}
```

Conclusion:

Enter an integer: 121

121 is a palindrome

Enter an integer: 123

123 is not a palindrome



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

g) Calculate factorial of N numbers.

```
#include <stdio.h>
int main() {
    int n, i, factorial = 1;
    printf ("Enter a non-negative integer : ");
    scanf ("%d", &n);
    if (n < 0) {
        printf ("Factorial is not defined for negative numbers.\n");
    } else {
        for (i=1; i<=n; ++i) {
            factorial *= i;
        }
        printf ("Factorial of %d = %d\n", n, factorial);
    }
}
```

Conclusion:

Enter a non-negative integer : 5
Factorial of 5 = 120



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

EXPERIMENT NO: 6

a) Using break statement: It prints numbers from 1 to 4 and exits the loop when the loop variable i equals 5.

```
#include <stdio.h>
int main () {
    int i;
    for (i=1;i<=10;i++){
        if (i==5){
            break;
        }
        printf ("%d",i);
    }
    printf ("\n");
    return 0
}
```

Conclusion

1 2 3 4



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

b) Using continue statement: It prints only odd numbers from 1 to 10 by skipping even numbers using the continue statement.

```
#include <stdio.h>
int main () {
    for (int i=1 ; i<=10 ; i++) {
        if (i % 2 == 0) {
            continue;
        }
        printf ("%d", i);
    }
    printf ("\n");
    return 0;
}
```

Conclusion:

1 3 5 7 9



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 7

a) Menu-Driven Program with Error Handling:

```
#include <stdio.h>

int main() {
    int mainchoice, num1, num2, num3 result;
    do {
        printf ("\nMenu:\n");
        printf ("1. Addition\n");
        printf ("2. Subtraction\n");
        printf ("3. Multiplication\n");
        printf ("4. Division\n");
        printf ("5. Exit\n");
        switch (choice) {
            Case 1:
                printf ("Enter two numbers: ");
                scanf ("%d %d", &num1, &num2);
                result = num1 + num2;
                printf ("Sum = %d\n", result);
                break;
            Case 2:
                printf ("Enter two numbers: ");
                scanf ("%d %d", &num1, &num2);
                result = num2 - num1;
                printf ("Difference = %d\n", result);
                break;
            Case 3:
                printf ("Enter two numbers: ");
                . . .
                scanf ("%d %d", &num1, &num2);
                if (num2 == 0) {
                    Conclusion: printf ("Error : Division by zero\n");
                } else {
                    result = num1 / num2;
                }
        }
    } while (choice != 5);
    return 0;
}
```

Menu

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

Enter choice: 2

```
printf ("Quotient = %d\n", result)
} break;
case 5:
    printf ("Exiting ... \n");
}
} while (choice != 5);

return 0;
}
```

Enter two numbers: 10 20
Sum = 30



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

b) Create a menu-driven program that allows users to perform various operations (e.g., calculate factorial, find prime numbers). Implement error handling using a goto statement to redirect to an error message label if invalid input is provided.

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

```
int main() {
    int choice, num, i, fact = 1, isPrime = 1;
    while (1) {
        printf ("In Menu:\n");
        printf ("1. Calculate Factorial\n");
        printf ("2. Check For Prime Number\n");
        printf ("3. Exit\n");
        printf ("Enter your choice: ");
        scanf ("%d", &choice);
        switch choice {
            case 1:
                printf ("Entered a non-negative integer: ");
                scanf ("%d", &num);
                if (num < 0) {
                    goto error;
                } for (i = 1; i <= num; i++) {
                    fact *= i;
                }
                printf ("Factorial of %d = %d\n", num, fact);
                break;
            case 2:
                printf ("Enter a positive integer: ");
                scanf ("%d", &num);
                if (num <= 1) {
                    goto error;
                }
                // Implement prime number check here
                // ...
                isPrime = 1;
                for (i = 2; i < num; i++) {
                    if (num % i == 0) {
                        isPrime = 0;
                        break;
                    }
                }
                if (isPrime) {
                    printf ("%d is a prime number.\n", num);
                } else {
                    printf ("%d is not a prime number.\n", num);
                }
                break;
        }
    }
error:
    // Handle error condition
}
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

```

}
for (i=2; i <= num/2; ++i){
    if (num % i == 0){
        isPrime = 0;
        break;
    }
}
if (isPrime) {
    printf ("%d is a prime number.\n", num);
} else {
    printf ("%d is not a prime number.\n", num);
}
break;
Case 3:
printf ("Exiting the Program.\n");
return 0;
default: goto error;
}
}
error:
printf ("Invalid input, Please enter a valid choice.\n");
}

```

Conclusion:

Menu:

- 1. Calculate Factorial
- 2. Check for Prime Number
- 3. Exit

Enter your choice:-1
Enter a non-negative integer:-5
Factorial of 5 = 120

Menu:

- 1. Calculate Factorial
- 2. Check for Prime Number
- 3. Exit

Enter your choice:-2
Enter a positive integer:-7
7 is a prime number.

Menu:

- 1. Calculate Factorial
- 2. Check for Prime Number
- 3. Exit

Enter your choice:-3
Invalid output.
Please enter valid choice



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

File Processing with Interrupt Handling

```
#include <stdio.h>
int dataBuffer[10];
volatile int dataReady = 0;
void timerISR() {
    for (int i=0; i<10; i++) {
        dataBuffer[i] = i;
    }
    dataReady = 1;
}
int main() {
    FILE *fp;
    while (1) {
        if (dataReady) {
            disableInterrupts();
            fp = fopen ("data.txt", "w");
            if (fp == NULL) {
                perror ("Error opening file");
                exit (1);
            }
            for (int i=0; i<10; i++) {
                fprintf (fp, "%d", dataBuffer[i]);
            }
            fprintf (fp, "\n");
            if (fclose (fp)) {
                dataReady = 0;
                enableInterrupts();
            }
        }
    }
}
```

Conclusion:

Print	0 1 2 3 4 5 6 7 8 9
DE	0 1 2 3 4 5 6 7 8 9
CAS	...
P	
S	
I	



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

d) Develop a program that reads data from a file and performs certain operations on the data.
Implement interrupt handling using a goto statement to jump to an error-handling section if the file cannot be opened or read successfully.

```
#include <stdio.h>
#include <stdlib.h>
int main (){
FILE *fp;
int data;
int sum=0, count=0;
fp = fopen ("data.txt", "r");
if (fp=NULL) {
    goto error;
}
while (fscanf (fp, "%d", &data) != EOF){
    sum+=data;
    Count++;
}
fclose (fp);
printf ("Sum of numbers : %d \n", sum);
printf ("Average : %.2f \n", (float)sum / count);
return 0;
error
printf ("Error opening or reading file.\n");
return 1;
}
```

Conclusion:

10
20
30
40
50

Sum of numbers : 150
Average : 30.00



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

EXPERIMENT NO: 8

a) Write a program to find the sum and average of elements in an array.

```
#include <stdio.h>
int main(){
    int arr[100], n, i, sum = 0;
    float avg;
    printf ("Enter the number of elements : ");
    scanf ("%d", &n);
    printf ("Enter the elements :\n");
    for (i=0; i<n; i++) {
        scanf ("%d", &arr[i]);
    }
    for (i=0; i<n; i++) {
        sum
        + = arr[i];
    }
    avg = (float) sum / n;
    printf ("Sum of the array elements: %d\n", sum);
    printf ("Average of the array elements: %.2f\n", avg);
    return
    0;
}
```

Conclusion:

Enter the number of elements: 5

Enter the element

10

20

30

40

50

Sum of the array elements : 150

Average of the array elements : 30.00



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

b) Reverse the elements of an array.

```
#include <stdio.h>
Void reverseArray (int arr[], int size){
    int Start = 0, end = size - 1;
    while (Start < end) {
        int temp = arr[Start];
        arr[Start] = arr[end];
        arr[end] = temp;
        Start++;
        end--;
    }
}
int main(){
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    printf ("Original array: ");
    for (int i = 0; i < size; i++) {
        printf ("%d", arr[i]);
    }
    reversalArray (arr, size);
    printf ("\n Reversed array: ");
    for (int i = 0; i < size; i++) {
        printf ("%d", arr[i]);
    }
    return 0;
}
```

Conclusion:

Output

Original array: 1, 2, 3, 4, 5
Reversed array: 5, 4, 3, 2, 1



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

c) Implement a program to find the largest and smallest elements in an array.

```
#include <stdio.h>
Void find largest Smallest (int arr[] ; int n) {
    int largest = arr[0];
    int smallest = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] > largest) {
            largest = arr[i];
        }
    }
    printf ("Largest element : %d\n", largest);
    printf ("Smallest element : %d\n", smallest);
}

int main () {
    int arr[]
    = {12, 31, 56, 78, 90, 23, 45, 67, 89, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    find largest Smallest (arr, n);
    return 0;
}
```

Conclusion:

Output

Largest element : 90
Smallest element : 10



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

d) Develop a program to search for a specific element in an array and display its index.

```
#include <stdio.h>
int main() {
    int arr[100], n, i, key, found = 0;
    printf ("Enter the number of elements : ");
    scanf ("%d", &n);
    printf ("Enter the elements : \n");
    for (i=0; i<n; i++) {
        scanf ("%d", &arr[i]);
    }
    printf ("Enter the element to search : ");
    scanf ("%d", &key);
    for (i=0; i<n; i++) {
        if (arr[i] == key) {
            found = 1;
            break;
        }
    }
    if (found == 1) {
        printf ("Element found at index %d\n", i);
    } else {
        printf ("Element not found\n");
    }
    return 0;
}
```

Conclusion:

Example Output

```
Enter the number of elements : 5
Enter the elements
0, 20, 30, 40, 50
Enter the element to search : 30
Enter found at index 2
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

1) Perform addition of two matrices.

```
#include <stdio.h>
int main(){
    int m, n, i, j;
    printf("Enter the number of rows and columns of the matrices:");
    scanf("%d %d", &m, &n);
    int matrix[m][n], matrix2[m][n], sum[m][n];
    printf("Enter the elements of the first matrix:\n");
    for (i=0; i<m; i++){
        for (j=0; j<n; j++){
            scanf("%d", &matrix[i][j]);
        }
    }
    printf("Enter the elements of the second matrix:\n");
    for (i=0; i<m; i++){
        for (j=0; j<n; j++){
            scanf("%d", &matrix2[i][j]);
        }
    }
    for (i=0; i<m; i++){
        for (j=0; j<n; j++){
            sum[i][j] = matrix[i][j] + matrix2[i][j];
        }
    }
    printf("Sum matrix:\n");
    for (i=0; i<m; i++){
        for (j=0; j<n; j++){
            printf("%d ", sum[i][j]);
        }
        printf("\n");
    }
}
```

Conclusion:

Enter the number of rows of the matrices: 2 3

Enter the element of the first matrix:

1 2 3

4 5 6

Enter the element of the second matrix:

7 8 9

10 11 12



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

EXPERIMENT NO: 9

a) Calculate the length of a string without `strlen`

```
#include <stdio.h>
#include <string.h>
int main(){
    char str[100];
    printf ("Enter a string:");
    gets (str);
    int length = strlen (str);
    length
    printf ("Length of the string : %d\n", length);
    return 0;
}
```

Conclusion:

Output

```
Enter a string:Hello, world!
length of the string : 13
```



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

b) Concatenate two strings without using the built-in function.

```
#include <stdio.h>
#include <String.h>
int main(){
    char str1[100], str2[100], result[200];
    int i, j, k;
    printf("Enter the first string:");
    scanf("%s", str1);
    printf("Enter the second string:");
    scanf("%s", str2);
    i = strlen(str1);
    j = strlen(str2);
    for (k = 0; k < j; k++){
        result[k] = str1[k];
    }
    for (k = 0; k < i; k++){
        result[i + k] = str2[k];
    }
    result[i + j] = '\0';
    printf("Concatenated string: %s\n", result);
    return 0;
}
```

Conclusion:

Output

Enter the first string: Hello

Enter the Second string: World

Concatenated string: HelloWorld

(✓)