

COMPUTER LABSHEET

**MySQL
C program**

**PREPARED BY :
RAYUSH SAH
12 P2
05**



1) Write a program to check whether a given number is a prime number using a function.

```
#include <stdio.h>
int isprime (int number) {
    int i;
    if (number <= 1)
        return 0;

    for (i=2; i<=number/2; i++) {
        if (number % i == 0)
            return 0;
    }
    return 1;
}

int main(){
    int num;
    printf ("enter a number");
    scanf ("%d", &num);

    if (isprime(num))
        printf ("%d is a prime number", num);
    else
        printf ("%d is not a prime number", num);
    return 0;
}
```

```
Enter a number: 13
13 is a Prime number.

...Program finished with exit code 0
Press ENTER to exit console.
```

2) Write a program to calculate the factorial of a number using iteration and a function.

```
#include <stdio.h>

int factorial (int n) {
    int fact = 1, i;
    for (i=1; i<=n; i++) {
        fact = fact * i;
    }
    return fact;
}

int main() {
    int num;
    printf ("Enter a number: ");
    scanf ("%d", &num);
    printf ("Factorial of %d is %d", num, factorial (num));
    return 0;
}
```

```
Enter a number: 6
Factorial of 6 is 720
...Program finished with exit code 0
Press ENTER to exit console.
```

3) Write a program to calculate the factorial of a number using iteration and a function.

```
#include <stdio.h>

int isPalindrome (int n) {
    int rev = 0, temp = n;
    while (temp != 0) {
        rev = rev * 10 + temp % 10;
        temp = temp / 10;
    }
    if (rev == n)
        return 1;
    else
        return 0;
}

int main () {
    int num;
    printf ("enter a number:");
    scanf ("%d", &num);
    if (isPalindrome (num))
        printf ("%d is a palindrome number ", num);
    else
        printf ("%d is not a palindrome number ", num);
    return 0;
}
```

```
Enter a number: 121
121 is a Palindrome number.
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

3) Write a program to calculate the factorial of a number using iteration and a function.

```
#include <stdio.h>

Void armstrongUpToN (int n) {
    int i, num, digit, sum;
    for (i = 1; i<=n; i++) {
        num = i;
        sum = 0;
        while (num != 0) {
            digit = num % 10;
            sum = sum + digit * digit * digit;
            num = num / 10;
        }
        if (sum == i)
            printf ("%d", i);
    }
}

int main () {
    int n;
    printf ("Enter the limit N:");
    scanf ("%d", &n);
    printf ("Armstrong numbers up to %d are: \n", n);
    armstrongUpToN (n);
    return 0;
}
```

```
Enter the limit N: 500000
Armstrong numbers up to 500000 are:
1 153 370 371 407
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

Table Schema:

Students

Column Name	Data type	Constraints
student_id	INT	PRIMARY KEY, AUTO-INCREMENT
first_name	VARCHAR(50)	NOT NULL
last_name	VARCHAR(50)	NOT NULL
gender	ENUM('Male', 'Female', 'Other')	NOT NULL
dob	DATE	NOT NULL
department	VARCHAR(100)	NOT NULL
admission_year	INT	NOT NULL
marks	INT	CHECK (marks BETWEEN 0 AND 100)

Solution

Table Setup

* Create Database and Table

```

CREATE DATABASE IF NOT EXISTS school-db;
USE school-db;

CREATE TABLE students (
    student_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    gender ENUM('Male', 'Female', 'Other') NOT NULL,
    dob DATE NOT NULL,
    department VARCHAR(100) NOT NULL,
    admission_year INT NOT NULL,
    marks INT CHECK (marks >= 0 AND marks <= 100)
);

```

Query OK, 1 row affected (0.01 sec)

Database changed

Query OK, 0 rows affected (0.03 sec)

* Insert Sample Data

```
INSERT INTO students (first_name, last_name, gender, dob, department, admission_year, marks)
VALUES
```

```
('Prakash', 'Shrestha', 'Male', '2002-03-15', 'Computer Science', 2020, 80),
('Sita', 'Shrestha', 'Female', '2001-07-22', 'Business', 2019, 90),
('Prakash', 'Thapa', 'Male', '2003-01-10', 'Engineering', 2021, 60),
('Maya', 'Lama', 'Female', '2000-11-05', 'Computer Science', 2018, 96),
('Aarav', 'KC', 'Male', '2002-05-28', 'Arts', 2020, 80),
('Nisha', 'Magar', 'Female', '2001-09-12', 'Engineering', 2019, 89);
```

```
Query OK, 6 rows affected (0.02 sec)
```

```
Records: 6 Duplicates: 0 Warnings: 0
```

* Select all records

```
SELECT *
```

```
FROM students;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| student_id | first_name | last_name | gender | dob       | department      | admission_year | marks |
+-----+-----+-----+-----+-----+-----+-----+
| 1          | Prakash   | Shrestha  | Male   | 2002-03-15 | Computer Science | 2020           | 80   |
| 2          | Sita       | Shrestha  | Female | 2001-07-22 | Business        | 2019           | 90   |
| 3          | Prakash   | Thapa     | Male   | 2003-01-10 | Engineering     | 2021           | 60   |
| 4          | Maya       | Lama      | Female | 2000-11-05 | Computer Science | 2018           | 96   |
| 5          | Aarav      | KC         | Male   | 2002-05-28 | Arts            | 2020           | 80   |
| 6          | Nisha      | Magar     | Female | 2001-09-12 | Engineering     | 2019           | 89   |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

* find total number of student

```
SELECT COUNT(*) as total_students
FROM students;
```

```
+-----+
| total_students |
+-----+
| 6             |
+-----+
1 row in set (0.00 sec)
```

* Select specific columns

```
SELECT first_name, last_name, department  
FROM students;
```

first_name	last_name	department
Prakash	Shrestha	Computer Science
Sita	Shrestha	Business
Prakash	Thapa	Engineering
Maya	Lama	Computer Science
Aarav	KC	Arts
Nisha	Magar	Engineering

6 rows in set (0.01 sec)

* Filter by department

```
SELECT *  
FROM students  
WHERE department = 'Engineering';
```

student_id	first_name	last_name	gender	dob	department	admission_year	marks
3	Prakash	Thapa	Male	2003-01-10	Engineering	2021	60
6	Nisha	Magar	Female	2001-09-12	Engineering	2019	89

2 rows in set (0.01 sec)

* Order by Marks (highest first)

```
SELECT *  
FROM students  
ORDER BY marks DESC;
```

student_id	first_name	last_name	gender	dob	department	admission_year	marks
4	Maya	Lama	Female	2000-11-05	Computer Science	2018	96
2	Sita	Shrestha	Female	2001-07-22	Business	2019	90
6	Nisha	Magar	Female	2001-09-12	Engineering	2019	89
1	Prakash	Shrestha	Male	2002-03-15	Computer Science	2020	80
5	Aarav	KC	Male	2002-05-28	Arts	2020	80
3	Prakash	Thapa	Male	2003-01-10	Engineering	2021	60

6 rows in set (0.01 sec)

* Show full names using CONCAT

```
SELECT CONCAT(first-name, ' ', last-name) AS full-name  
FROM students;
```

full_name
Prakash Shrestha
Sita Shrestha
Prakash Thapa
Maya Lama
Aarav KC
Nisha Magar

6 rows in set (0.01 sec)

SQL wildcards and Special symbols

Symbol	Use in MySQL
%	Matches any sequence of characters
-	Matches exactly one character
[]	Matches a single character from a set /range
[^]	Matches a single character not in set /range
\	Escape character (with ESCAPE)

Search Students

* by first name

```
SELECT *  
FROM students  
WHERE first-name LIKE '%Prakash%';
```

student_id	first_name	last_name	gender	dob	department	admission_year
1	Prakash	Shrestha	Male	2002-03-15	Computer Science	2020
3	Prakash	Thapa	Male	2003-01-10	Engineering	2021

2 rows in set (0.01 sec)

* by full name

```
SELECT *  
FROM students
```

```
WHERE CONCAT(first_name, ' ', last_name) LIKE '%Prakash Shrestha%';
```

student_id	first_name	last_name	gender	dob	department	admission_year
1	Prakash	Shrestha	Male	2002-03-15	Computer Science	2020

1 row in set (0.01 sec)

* all with prepared statement

```
PREPARE stmt FROM
```

```
'SELECT *  
FROM students
```

```
WHERE first_name LIKE ?
```

```
OR last_name LIKE ?
```

```
OR CONCAT(first_name, " ", last_name) LIKE ?';
```

student_id	first_name	last_name	gender	dob	department	admission_year
1	Prakash	Shrestha	Male	2002-03-15	Computer Science	2020
3	Prakash	Thapa	Male	2003-01-10	Engineering	2021

2 rows in set (0.01 sec)

* Count students in each department

```
SELECT department, COUNT(*) AS total_students  
FROM students  
GROUP BY department;
```

department	total_students
Arts	1
Business	1
Computer Science	2
Engineering	2

4 rows in set (0.01 sec)

* Students admitted after 2019

```
SELECT * FROM students  
WHERE admission_year > 2019;
```

student_id	first_name	last_name	gender	dob	department	admission_year
1	Prakash	Shrestha	Male	2002-03-15	Computer Science	2020
3	Prakash	Thapa	Male	2003-01-10	Engineering	2021
5	Aarav	KC	Male	2002-05-28	Arts	2020

3 rows in set (0.01 sec)

* Total marks by gender

```
SELECT gender, SUM(marks) AS total_marks  
FROM students  
GROUP BY gender;
```

gender	total_marks
Female	275
Male	220

2 rows in set (0.01 sec)

* Show only the full name and pass/fail result

```
SELECT
    CONCAT(first-name, ' ', last-name) AS full-name,
    CASE
        WHEN marks >= 40 THEN 'Pass'
        ELSE 'Fail'
    END AS result
FROM students;
```

full_name	result
Prakash Shrestha	Pass
Sita Shrestha	Pass
Prakash Thapa	Pass
Maya Lama	Pass
Aarav KC	Pass
Nisha Magar	Pass

6 rows in set (0.01 sec)

Update Table

* Update Marks for student

```
UPDATE students
SET marks = 95
WHERE student-id = 1;
```

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 0