ASSIGNMENT 10

Q. Create a branch in your calculator program to do hexadecimal calculation. Write code and develop two branches. Merge the two to have a decimal/hex calculator. Demonstrate git branch, merge capability.

→

Step 1: Initialize Git

```
sus@MrugankshaK MINGW64 ~/Documents/sem4/dtl
 git init
nitialized empty Git repository in C:/Users/Asus/Documents/sem4/dtl/.git/
```

Step 2 : Create and Switch to a new Branch

Create a branch for **hexadecimal calculations** and switch to it

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git branch hex_calculator

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git checkout hex_calculator
Switched to branch 'hex_calculator'
```

Step 3: Modify Code for Hexadecimal

- **Modified calculator.c**

```
#include <stdio.h>
#include <math.h>
void calculate_decimal(double a, char op, double b) {
 double result;
 switch (op) {
    case '+': result = a + b; break;
    case '-': result = a - b; break;
    case '*': result = a * b; break;
    case '/':
      if (b != 0)
        result = a / b;
      else {
        printf("Error: Division by zero\n");
        return;
      }
```

```
            break;
        case '^': result = pow(a, b); break;
        default:
            printf("Error: Invalid operator\n");
            return;
    }
   printf("Decimal Result: %.2lf\n", result);
}
void calculate_hexadecimal() {
    int num1, num2, result;
    char operator;
    printf("Enter a hexadecimal expression (e.g., A + B): ");
    scanf("%x %c %x", &num1, &operator, &num2);
    switch (operator) {
        case '+': result = num1 + num2; break;
        case '-': result = num1 - num2; break;
        case '*': result = num1 * num2; break;
        case '/':
            if (num2 != 0)
                result = num1 / num2;
            else {
                printf("Error: Division by zero\n");
                return;
            }
            break;
        default:
            printf("Error: Invalid operator\n");
            return;
    }
    printf("Hexadecimal Result: %X\n", result);
}
int main() {
    int choice;
```

```
    printf("Choose Calculation Mode:\n1. Decimal\n2. Hexadecimal\nEnter choice: ");
    scanf("%d", &choice);
    if (choice == 1) {
        double num1, num2;
        char operator;
        printf("Enter a decimal expression (e.g., 2 + 3): ");
        scanf("%lf %c %lf", &num1, &operator, &num2);
        calculate_decimal(num1, operator, num2);
    } else if (choice == 2) {
        calculate_hexadecimal();
    } else {
        printf("Invalid choice.\n");
    }
    return 0;
}
```

Step 4: Commit changes in hex_calculator branch

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (hex_calculator)
$ git add calculator.c

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (hex_calculator)
$ git commit -m "added hexadecimal calculator"
[hex_calculator db73744] added hexadecimal calculator
 1 file changed, 41 insertions(+), 7 deletions(-)
```

Step 5: Switch to main Branch and Merge

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (hex_calculator)
$ git checkout master
Switched to branch 'master'

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git merge hex_calculator
Updating a64f7bf..db73744
Fast-forward
 calculator.c | 48 ++++++++++++++++++++++++++++++++++++++++++++-------
 1 file changed, 41 insertions(+), 7 deletions(-)
```

Step 6 : Verify Merge And Resolve It

If there are conflicts, Git will notify you. Open calculator.c, fix the issues manually, and then:

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git add calculator.c

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git commit -m "resolved conflict issue"
[master 4e13863] resolved conflict issue
 1 file changed, 2 insertions(+), 2 deletions(-)
```

Step 7: Check branch and Log

View commit history

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git log --oneline --graph --all
* 4e13863 (HEAD -> master) resolved conflict issue
* db73744 (hex_calculator) added hexadecimal calculator
* a64f7bf (origin/master) added 8th assignment

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git branch
  hex_calculator
* master
```

Step 8: Delete the Merged Branch

```
Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git branch -d hex_calculator
Deleted branch hex_calculator (was db73744).

Asus@MrugankshaK MINGW64 ~/Documents/sem4/dtl (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.04 KiB | 1.04 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Mruganksha/DTL-Lab.git
   a64f7bf..4e13863  master -> master
```

Conclusion:

We successfully implemented hexadecimal calculations by creating and merging a feature branch using Git. This demonstrated Git's branching capability, enabling structured development, feature isolation, and integration. Through git branch, checkout, merge, and conflict resolution, we ensured a smooth workflow for collaborative software development.