Date: 07/06/2021

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

1. Days of the Week

Aim:

To get any integer from 1 to 7 as input from the user and to print the corresponding day of the week.

Algorithm:

Step 1: Start
Step 2: Read an integer (range: 1-7) as input from the user.
Step 3: Using the switch conditional, compare the integer with each of the numbers from 1 to 7.
Step 4: If the numbers match, execute the associated statement to print the corresponding day of the week and go to Step 6. If not, move on to the next case (number).

The cases are as follows:

- For integer=1, print *Monday*.
- For integer=2, print *Tuesday*.
- … and so on, till…
- For integer=7, print *Sunday*.

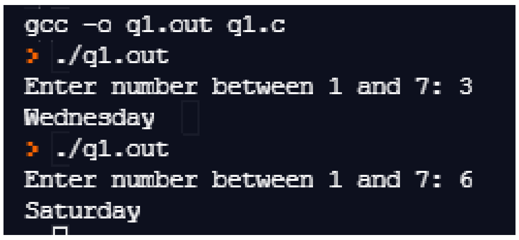
Step 5: If none of the cases match, then state that the input is invalid and go to Step 6.
Step 6: End

Code:

```c
//to get a number (1-7) as input and print the corresponding day of the week
#include <stdio.h>

int main(void) {
  int num;
  printf("Enter number between 1 and 7: ");
  scanf("%d",&num);
  switch(num){
    case 1: printf("Monday"); break;
    case 2: printf("Tuesday"); break;
    case 3: printf("Wednesday"); break;
    case 4: printf("Thursday"); break;
    case 5: printf("Friday"); break;
    case 6: printf("Saturday"); break;
    case 7: printf("Sunday"); break;
    default: printf("Invalid input!");
  }
  printf("\n");

  return 0;
}
```

Exercise 3

Output:



```
gcc -o q1.out q1.c
> ./q1.out
Enter number between 1 and 7: 3
Wednesday
> ./q1.out
Enter number between 1 and 7: 6
Saturday
```

Result:

A program for displaying the name of the nth day of the week is written and executed.

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

2. Number of Days in a Month

Aim:

To get any integer from 1 to 12 as input from the user and to print the number of days in the month corresponding to that number.

Algorithm:

Step 1: Start
Step 2: Read an integer (range: 1-12) as input from the user.
Step 3: Using the switch conditional, compare the integer with each of the numbers from 1 to 12.
Step 4: If the numbers match, execute the associated statement to print the number of days in the corresponding month and go to Step 6. If not, move on to the next case (number).

The cases are as follows:

- For integer=1,4,6,9,11, print *30*.
- For integer=2, print *28 (non-leap years) or 29 (leap years)*.
- For integer=3,5,7,8,10,12, print *31*.

Step 5: If none of the cases match, then state that the input is invalid and go to Step 6.
Step 6: End

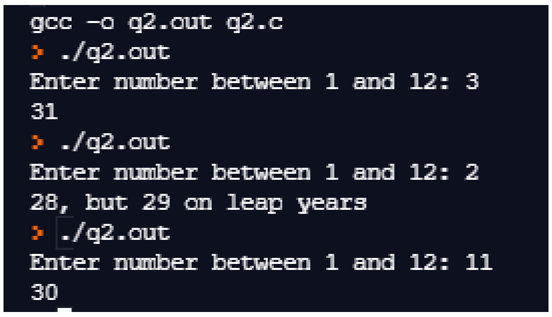
Code:

```c
//to get a number (1-12) as input and print the number of days in the corresponding month
#include <stdio.h>

int main(void) {
  int num;
  printf("Enter number between 1 and 12: ");
  scanf("%d",&num);
  switch(num){
    case 1: case 4: case 6: case 9: case 11: printf("30"); break;
    case 2: printf("28, but 29 on leap years"); break;
    case 3: case 5: case 7: case 8: case 10: case 12: printf("31"); break;
    default: printf("Invalid input!");
  }
  printf("\n");

  return 0;
}
```

<u>Exercise 3</u>

Name: *Krithika Swaminathan*
Roll No.: *205001057*

Output:



```
gcc -o q2.out q2.c
> ./q2.out
Enter number between 1 and 12: 3
31
> ./q2.out
Enter number between 1 and 12: 2
28, but 29 on leap years
> ./q2.out
Enter number between 1 and 12: 11
30
```

Result:

A program for displaying the number of days in a given month is written and executed.

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

3. Odd or Even (Switch-case)

Aim:

To get an integer as input from the user and to classify the integer as odd or even using the switch-case conditional statement.

Algorithm:

Step 1: Start
Step 2: Read an integer as input from the user.
Step 3: Using the switch conditional, compare the remainder obtained when the integer is divided by 2 with the number in each case.
Step 4: If the numbers match, execute the associated statement to print the conclusion and go to Step 5. If not, move on to the next case.

The cases are as follows:

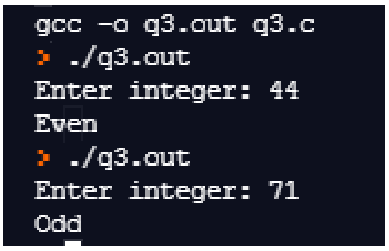
- For remainder=0, print *Even*.
- For remainder=1, print *Odd*.

Step 5: End

Code:

```c
//to check if a number is even or odd using switch-case
#include <stdio.h>

int main(void) {
  int num;
  printf("Enter integer: ");
  scanf("%d",&num);
  switch(num%2){
    case 0: printf("Even"); break;
    case 1: printf("Odd"); break;
  }
  printf("\n");

  return 0;
}
```

Output:



```
gcc -o q3.out q3.c
> ./q3.out
Enter integer: 44
Even
> ./q3.out
Enter integer: 71
Odd
```

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

Result:

A program for identifying a given integer as odd or even using switch-case is written and executed.

Date: 07/06/2021

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

4. Basic Calculator

Aim:

To create a menu-driven calculator that performs basic functions, namely, addition, subtraction, multiplication and division.

Algorithm:

Step 1: Start
Step 2: Read two numbers and an associated binary operator in the format of an expression, i.e., ‘*num1*’ ’*op*’ ’*num2*’.
Step 3: Using the switch conditional, switch *op* with each case and check for a match.
Step 4: If the operation matches, then execute the accompanying statement and go to Step 6. If not, then move on to the next case.

The cases are as follows:

- For *op*=’+’, add the two numbers and store the result.
- For *op*=’-’, subtract the second number from the first and store the result.
- For *op*=’*’, multiply the two numbers and store the result.
- For *op*=’/’, divide the first number by the second and store the result.

Step 5: If none of the cases match, then state that the operation entered is invalid and go to Step 7.
Step 6: Print the result.
Step 7: End

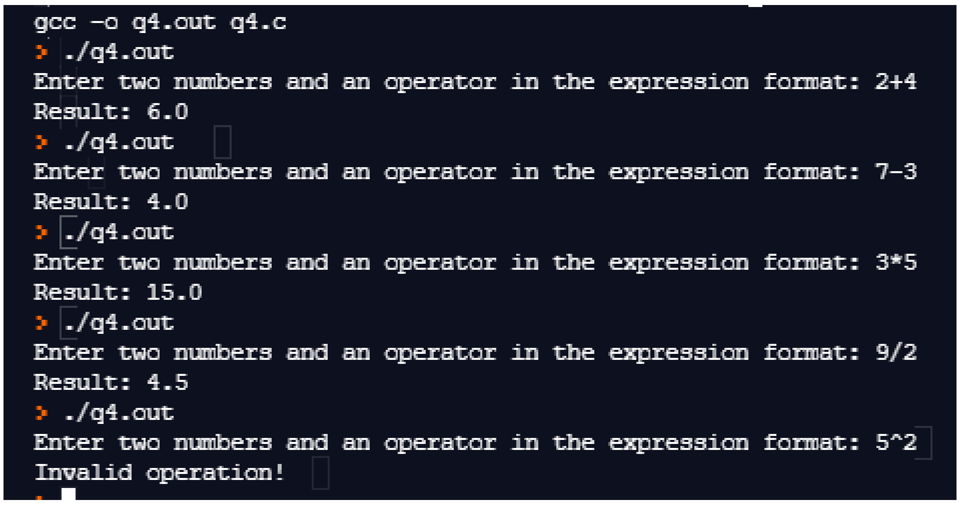
Code:

```c
//to create a menu-driven calculator that performs basic functions
#include <stdio.h>

int main(void) {
  float a,b,result;
  char op;
  printf("Enter two numbers and an operator in the expression format: ");
  scanf("%f%c%f",&a,&op,&b);
  switch(op){
    case '+': result=a+b; break;
    case '-': result=a-b; break;
    case '*': result=a*b; break;
    case '/': result=a/b; break;
    default: printf("Invalid operation!\n"); return 0;
  }
  printf("Result: %.1f\n",result);

  return 0;
}
```

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

Output:



```
gcc -o q4.out q4.c
> ./q4.out
Enter two numbers and an operator in the expression format: 2+4
Result: 6.0
> ./q4.out
Enter two numbers and an operator in the expression format: 7-3
Result: 4.0
> ./q4.out
Enter two numbers and an operator in the expression format: 3*5
Result: 15.0
> ./q4.out
Enter two numbers and an operator in the expression format: 9/2
Result: 4.5
> ./q4.out
Enter two numbers and an operator in the expression format: 5^2
Invalid operation!
```

Result:

A program to create a menu-driven calculator to perform basic operations is written and executed.

Name: *Krithika Swaminathan*
Roll No.: *205001057*

5. Product of Digits of a Number

Aim:

To get a number as input from the user and to find the product of the digits of the number.

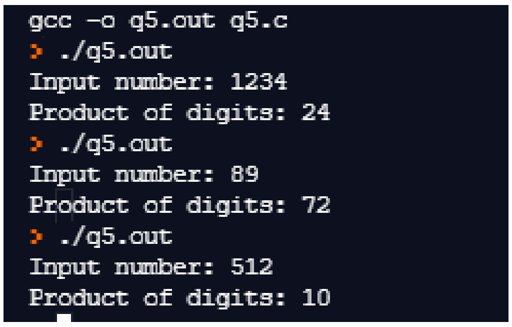
Algorithm:

Step 1: Start
Step 2: Read a number as input from the user.
Step 3: If the number is 0, set the product to be 0. If not, then initialise the product as 1.
Step 4: Run a loop to extract each digit from the number one by one (perform modulo division by 10 in each iteration and divide the number by 10 for the alteration) and multiply it with the existing product.
Step 5: Terminate the loop when all the digits have been multiplied and the altered number is 0.
Step 5: Print the product.
Step 6: End

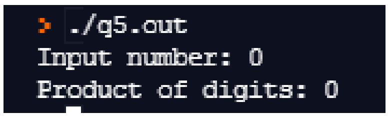
Code:

```c
//to find the product of the digits of a given number
#include <stdio.h>

int main(void) {
  int num,prod;
  printf("Input number: ");
  scanf("%d",&num);
  prod=(num==0)?0:1;
  for (int i=num; i>0; i/=10){
    prod*=(i%10);
  }
  printf("Product of digits: %d\n",prod);

  return 0;
}
```

Output:



```
gcc -o q5.out q5.c
> ./q5.out
Input number: 1234
Product of digits: 24
> ./q5.out
Input number: 89
Product of digits: 72
> ./q5.out
Input number: 512
Product of digits: 10
```



```
> ./q5.out
Input number: 0
Product of digits: 0
```

Exercise 3

Result:

A program for finding the product of the digits of a given number is written and executed.

Exercise 3

Name: *Krithika Swaminathan*
Roll No.: *205001057*

6. Swapping First and Last digits

Aim:

To get a number as input from the user and to swap the first and last digits of the number.

Algorithm:

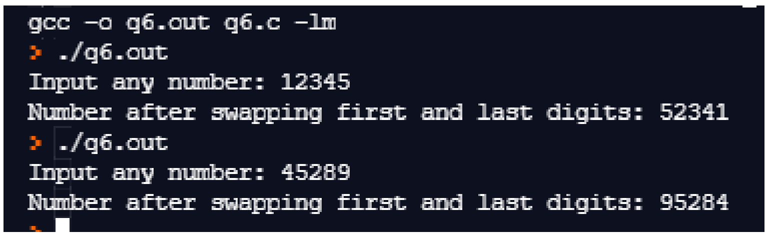
Step 1: Start
Step 2: Read a number as input from the user.
Step 3: Store the original last digit as the modulo division of the number by 10. This will be the first digit of the swapped number.
Step 4: Initialise a counter and set it to 0. Do the same for a variable that will contain the swapped number.
Step 5: Run a loop to extract each digit starting from the tens' digit one by one (modulo division by 10 in every iteration). After every iteration, alter the number by dividing it by 10 to access the next digit.
Step 6: In each iteration, increment the counter. Then, add the product of the digit and 10 raised to the counterth power to the previous value of the variable containing the swapped number.
Step 7: Terminate the loop when the altered number is a single digit and store this original first digit. This will be the last digit of the swapped number.
Step 8: Add the previously mentioned first and last digits to the value of the swapped number in the correct positions, i.e., need to add ((original last digit x $10^{counter+1}$) + original first digit).
Step 9: Print the swapped number.
Step 10: End

Code:

```c
//to swap the first and last digits of a given number
#include <stdio.h>
#include <math.h>

int main(void) {
  int num,swpd=0,i,ld,ctr=0;
  printf("Input any number: ");
  scanf("%d",&num);

  ld=num%10;
  for (i=num/10; i>9; i/=10){
    ctr++;
    swpd+=(i%10)*pow(10,ctr);
  }
  swpd+=(ld*pow(10,ctr+1))+i;

  printf("Number after swapping first and last digits: %d\n",swpd);

  return 0;
}
```

Exercise 3

Output:



```
gcc -o q6.out q6.c -lm
> ./q6.out
Input any number: 12345
Number after swapping first and last digits: 52341
> ./q6.out
Input any number: 45289
Number after swapping first and last digits: 95284
```

Result:

A program to swap the first and last digits of a given number is written and executed.