Lab 0

Operating Systems

Pranav Thakur

Name: Pranav Thakur

Class: BTECH CSE 4B

Roll no: 2K19CSUN01101

Laboratory Objective: To revisit and implement C Programming Concepts

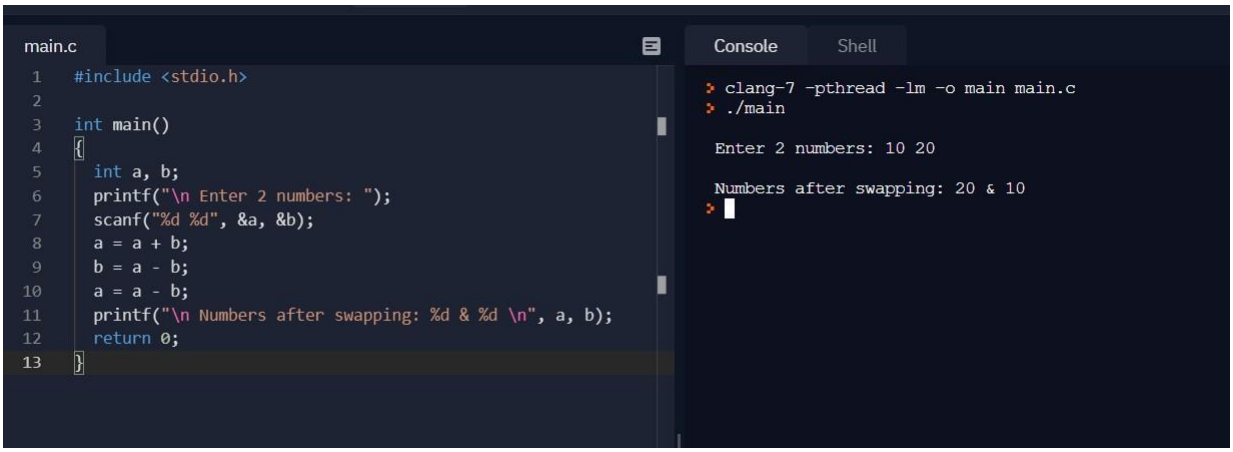
Learning Outcome: To gain familiarity with the concepts of C.

Course Outcome: CO1

Blooms Taxonomy: BT1, BT2, BT3, BT4

Write programs using the C/C++ Language

Basics



1. To swap two numbers without using a temporary variable.

```c
#include <stdio.h>

int main()
{
  int quantity = 0;
  float rate, discount, tax, total;
  rate = discount = tax = total = 0.0;

  printf("\n Quantity sold: ");
  scanf("%d", &quantity);
  printf("\n Rate: ");
  scanf("%f", &rate);
  printf("\n Discount: ");
  scanf("%f", &discount);
  printf("\n Tax: ");
  scanf("%f", &tax);

  int totalRate = quantity * rate;

  total = ((totalRate * discount) / 100) + ((totalRate * tax)
  printf("\n Total: %f \n", total);
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

Quantity sold: 5

Rate: 100

Discount: 5

Tax: 12

Total: 85.000000
```

2. To calculate the bill amount for an item given the quantity sold, rate, discount and tax.

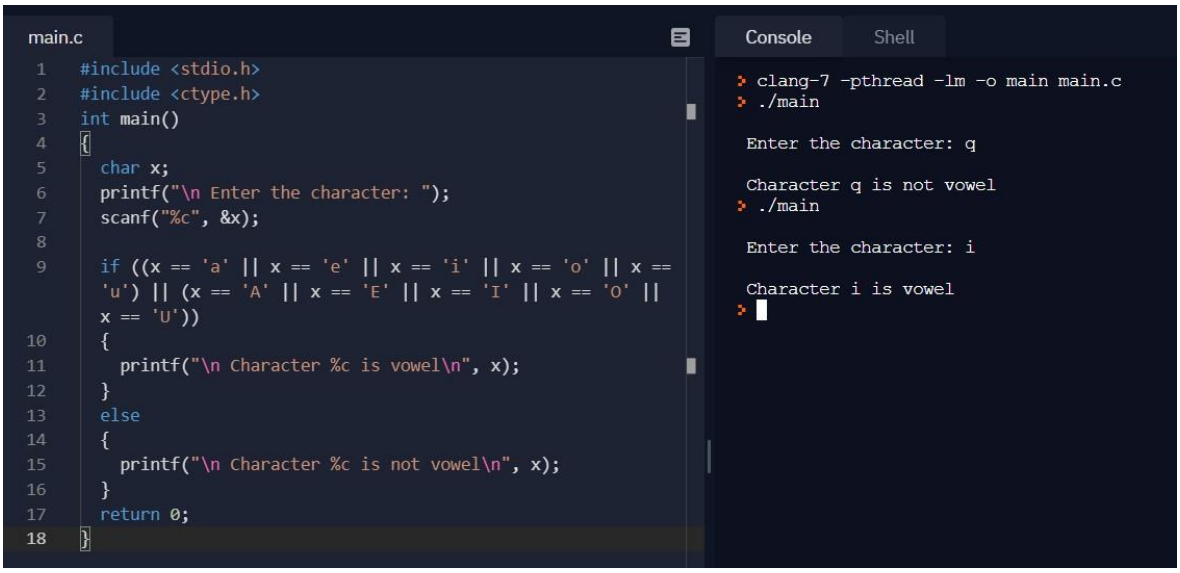
Based on Decision Control

3. Program to enter any character. If the entered character is in lower case then convert it into upper case and if it is lower case then convert it into upper case.

```c
#include <stdio.h>
#include <ctype.h>
int main()
{
  char x;
  printf("\n Enter the character: ");
  scanf("%c", &x);

  if (islower(x))
  {
    x = toupper(x);
    printf("\n Value of character entered: %c \n", x);
  }
  else
  {
    printf("\n Value of character entered: %c \n", x);
  }
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

Enter the character: q

Value of character entered: Q
```

4. To enter a character and determine whether it is vowel or not



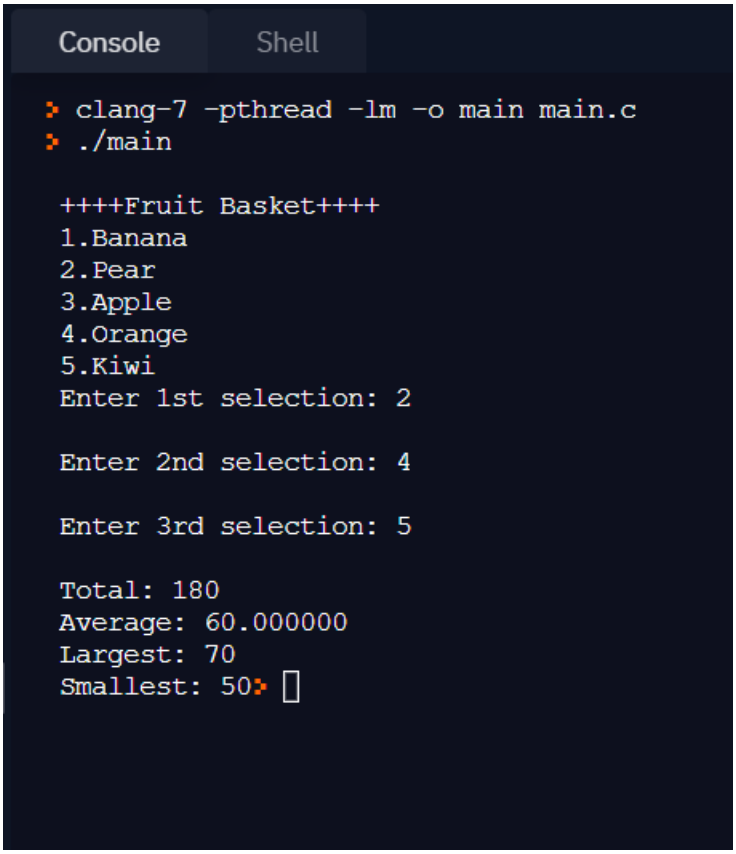
Based on Switch Case:

5. Program to display a menu that offers five options: Read three numbers, calculate total, average, display the smallest and largest value.

main.c

```c
#include <stdio.h>

int main()
{
  int num1, num2, num3, large, small, sum;
  int arr[5] = {40, 50, 30, 60, 70};
  float avg;
  printf("\n ++++Fruit Basket++++");
  printf("\n 1.Banana");
  printf("\n 2.Pear");
  printf("\n 3.Apple");
  printf("\n 4.Orange");
  printf("\n 5.Kiwi");

  printf("\n Enter 1st selection: ");
  scanf("%d", &num1);
  printf("\n Enter 2nd selection: ");
  scanf("%d", &num2);
  printf("\n Enter 3rd selection: ");
  scanf("%d", &num3);

  sum = arr[num1 - 1] + arr[num2 - 1] + arr[num3 - 1];
  avg = sum / 3;

  large = small = arr[num1 - 1];

  if (arr[num2 - 1] > large && arr[num2 - 1] > arr[num3 - 1])
  {
    large = arr[num2 - 1];
  }
  else if (arr[num3 - 1] > large)
  {
    large = arr[num3 - 1];
```

```
  if (arr[num2 - 1] > large && arr[num2 - 1] > arr[num3 - 1])
  {
    large = arr[num2 - 1];
  }
  else if (arr[num3 - 1] > large)
  {
    large = arr[num3 - 1];
  }

  if (arr[num2 - 1] < small && arr[num2 - 1] < arr[num3 - 1])
  {
    small = arr[num2 - 1];
  }
  else if (arr[num3 - 1] < small)
  {
    small = arr[num3 - 1];
  }

  printf("\n Total: %d", sum);
  printf("\n Average: %f", avg);
  printf("\n Largest: %d", large);
  printf("\n Smallest: %d", small);

  return 0;
}
```



Based on Iterative Statements

6. To classify a number as prime or composite.

main.c

```c
#include <stdio.h>

int main()
{
  int num, i;
  int flag = 0;
  printf("\n Enter a number: ");
  scanf("%d", &num);

  for (i = 2; i <= num / 2; i++)
  {
    if (num % i == 0)
    {
      flag = 1;
      break;
    }
  }

  if (num > 0)
  {
    if (num == 1)
    {
      printf("\n %d is neither prime nor composite \n", num);
    }
    else
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Enter a number: 97

 97 is a prime number
> ./main

 Enter a number: 98

 98 is not a prime number
>
```

```c
    if (num == 1)
    {
      printf("\n %d is neither prime nor composite \n", num);
    }
    else
    {
      if (flag == 0)
      {
        printf("\n %d is a prime number \n", num);
      }
      else
      {
        printf("\n %d is not a prime number \n", num);
      }
    }
  }
  else
  {
    printf("\n Number entered is invalid \n");
  }
  return 0;
}
```

Based on Functions

7. To find the Fibonacci using recursive function.

main.c

```c
#include <stdio.h>

void fib(int n)
{
  static int n1 = 0, n2 = 1, n3;
  if (n > 0)
  {
    n3 = n1 + n2;
    n1 = n2;
    n2 = n3;
    printf("%d ", n3);
    fib(n - 1);
  }
}

int main()
{
  int num;
  printf("\n Enter the number of elements: ");
  scanf("%d", &num);
  printf("\n Fibonacci Series: \n");
  printf("%d %d ", 0, 1);
  fib(num - 2);
  printf("\n");
  return 0;
}
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Enter the number of elements: 8

 Fibonacci Series:
0 1 1 2 3 5 8 13
>
```

Based on Arrays

8. To insert a number at a given location in an array.

main.c

```c
#include <stdio.h>

int main()
{
  int i, n, num, pos, arr[20];
  printf("\n Enter the array size: ");
  scanf("%d", &n);
  printf("\n Enter elements in array: \n");
  for (i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
  }
  printf("\n Enter the number to be insert: ");
  scanf("%d", &num);
  printf("\n Enter position to place in array: ");
  scanf("%d", &pos);

  for (i = n; i >= pos; i--) {
    arr[i + 1] = arr[i];
  }
  arr[pos-1] = num;
  n++;

  printf("\n Array after insertion: \n");
  for (i = 0; i < n; i++) {
      printf("%d ", arr[i]);
  }
  return 0;
}
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Enter the array size: 5

 Enter elements in array:
1 2 4 5 6

 Enter the number to be insert: 3

 Enter position to place in array: 3

 Array after insertion:
1 2 3 5 5 6 >
```

9. In a class there are 10 students. Each student is supposed to appear in three tests. Write a program using 2D array to print

(a) The marks obtained by each student in different subjects.

(b) Sort the average obtained by each student.

main.c

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

void swap(int *x, int *y)
{
  int temp = *x;
  *x = *y;
  *y = temp;
}

int compare(const void *a, const void *b)
{
  return (*(int *)a - *(int *)b);
}

int main()
{
  int students[10][3] = {
      {92, 87, 99},
      {90, 74, 49},
      {62, 73, 71},
      {81, 77, 68},
      {100, 99, 98},
      {90, 80, 79},
      {94, 95, 96},
      {90, 98, 80},
      {80, 88, 70},
      {50, 50, 50},
  };
  int avg[10], sum, i, j;

  for (i = 0; i < 10; i++)
  {
    sum = 0;
    for (j = 0; j < 3; j++)
    {
      sum += students[i][j];
    }
    avg[i] = sum / 3;
  }

  qsort(avg, 10, sizeof(int), compare);

  printf("\n Student detail: ");
  for (i = 0; i < 10; i++)
  {
    printf("Student %d: ", i + 1);
    for (j = 0; j < 3; j++)
    {
      printf("%d  ", students[i][j]);
    }
    printf("\n");
  }

  printf("\n Sorted average marks: ");
  for (i = 0; i < 10; i++)
  {
    printf("%d  ", avg[i]);
  }

  return 0;
}
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Student detail: Student 1: 92  87  99
Student 2: 90  74  49
Student 3: 62  73  71
Student 4: 81  77  68
Student 5: 100  99  98
Student 6: 90  80  79
Student 7: 94  95  96
Student 8: 90  98  80
Student 9: 80  88  70
Student 10: 50  50  50

 Sorted average marks: 50  68  71  75  79  83  89  92  95  99
```

10. Accept any two strings from the user. Display whether both the strings are equal or not. (do not use standard functions).

main.c

```c
#include <stdio.h>

int main() {
  int count1 = 0, count2 = 0, flag = 0, i, j;
  char string1[20], string2[20];
  printf("\n Enter 1st string: ");
  fgets(string1, sizeof(string1), stdin);
  printf("\n Enter 2nd string: ");
  fgets(string2, sizeof(string2), stdin);

  for (i = 0; string1[i] != '\0'; i++) {
    if (string1[i] == string2[i]) {
      flag = 1;
    } else {
      flag = 0;
    }
  }
  if (flag == 1) {
    printf("\n Both strings are equal\n");
  }
  else {
    printf("\n Both strings are equal\n");
  }
  return 0;
}
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Enter 1st string: hello

 Enter 2nd string: hello

 Both strings are equal
>
```

Based on Structures

11. Write a program to accept a list of 10 integers in an array, pass the starting address of array in sum function, calculate the sum of the array elements in the function and return the sum calculated in the main function.

main.c

```c
#include <stdio.h>

int sum(int arr[], int n) {
  int sum = 0, i;
  for (i = 0; i < n; i++) {
    sum += arr[i];
  }
  return sum;
}

int main() {
  int arr[10], i, n = 10;
  int result = 0;
  printf("\n Enter array values: ");
  for (i = 0; i < n; i++) {
    scanf("%d", &arr[i]);
  }
  result = sum(arr, n);
  printf("\n Sum of arrays: %d", result);
  return 0;
}
```

Console Shell

```
> clang-7 -pthread -lm -o main main.c
> ./main

 Enter array values: 1 2 3 4 5 6 7 8 9 10

 Sum of arrays: 55>
```