A PROJECT FILE FOR

“BASIC C PROGRAMMING THEORY & LAB”

OF

“COMPUTER PROGRAMMING”

Submitted To: GURPREET KAUR

Submitted By: mukul dev yagyavalkya

University Roll No: 2315001397

Roll No & Section: 39, AU1

WEEK 1

Q. 1 Write a program to accept height and base of triangle and calculate area of Triangle

Note: area =(h*b)/2

```
#include <stdio.h> int
main()
{
   int height, width, area;     printf("Please enter
height of the triangle: \n");     scanf(" %d",
&height);     printf("Please enter width of the
triangle: \n");     scanf(" %d", &width);     area =
(height * width) / 2;     printf("The area of the
triangle is: %d", area);
}
```

Q. 2 Write a program to accept radius of circle and calculate area of circle Note: area =pi * r^2

```
#include <stdio.h> int
main()
{
   int radius, area;     printf("Please enter radius
of the circle: \n");     scanf(" %d", &radius);
area = 3.14 * radius * radius;     printf("The
area of the circle is: %d", area);
}
```

Q. 3 Write a program to find the lowest marks of three students using conditional operator.

```
#include <stdio.h> int
main()
{
   int marks1, marks2, marks3; printf("Please
enter marks of student 1: \n"); scanf(" %d",
&marks1);     printf("Please enter marks of
student 2: \n");     scanf(" %d", &marks2);
printf("Please enter marks of student 3: \n");
scanf(" %d", &marks3);     if (marks1 < marks2
&& marks1 < marks3)
   {
       printf("Student 1 has the lowest marks");
   }
   else if (marks2 < marks1 && marks2 < marks3)
   {
       printf("Student 2 has the lowest marks");
   }
else
   {
       printf("Student 3 has the lowest marks");
   }
}
```

Q. 4 Write a program to Calculate Compound Interest.

```
#include <stdio.h>
#include <math.h> int
main()
{
   float principal, rate, time, compoundInterest;     printf("Enter
principal amount: ");     scanf("%f", &principal);     printf("Enter
rate of interest (in percentage): ");     scanf("%f", &rate);
printf("Enter time period (in years): ");     scanf("%f", &time);
rate = rate / 100;     compoundInterest = principal * pow(1 +
rate, time) - principal;     printf("Compound Interest = %.2f\n",
compoundInterest);
   return 0;
}
```

Q. 5 Write a program to Calculate Cube of a Number.

```
#include <stdio.h> int
main()
{
   int number = 2;     int cube =
number ^ 3;     int cube1 =
pow(number, 3);
printf("\nCube is %d", cube1);
}
```

WEEK 2

Q .1 Interchange two values using Assignment Operator:

```
#include <stdio.h>
   int main ()
{    int a, b;     printf("Enter two values:\n");
scanf("%d %d", &a, &b);     a = a + b;     b = a -
b;     a = a - b;     printf("Interchanged values:
%d %d\n", a, b);
   return 0;
}
```

Q .2 Interchange two values using Arithmetic Operator:

```
#include <stdio.h>
   int main ()
{     int a, b;     printf("Enter two
values:\n");     scanf("%d %d",
&a, &b);
    a = a * b; b = a / b; a = a / b;
    printf("Interchanged values: %d %d\n", a,
    b);
   return 0;
}
```

Q .3 Interchange two values using Bitwise Operator:

```
#include <stdio.h>
   int main ()
{     int a, b;     printf("Enter two values:\n");
scanf("%d %d", &a, &b);     a = a ^ b;     b = a ^
b;     a = a ^ b;     printf("Interchanged values:
%d %d\n", a, b);
   return 0;
}
```

Q .4 Find the size of all data types :

```
#include <stdio.h>
   int main()
{
   printf("Size of int: %lu bytes\n", sizeof(int));     printf("Size of
float: %lu bytes\n", sizeof(float));     printf("Size of char: %lu
bytes\n", sizeof(char));     printf("Size of double: %lu bytes\n",
sizeof(double));     printf("Size of long double: %lu bytes\n",
sizeof(long double));     printf("Size of short int: %lu bytes\n",
sizeof(short int));     return 0;
}
```

Q .5 Check whether input number is even or odd without using arithmetic operators :

```
#include <stdio.h>
    int main()
{
    int num; printf("Enter a number:
");     scanf("%d", &num);     if (num
& 1){         printf("The number is
odd.\n");
      }
    else{         printf("The number is
even.\n");
       }
    return 0;
}
```

WEEK 3

Q .1 Check whether a given number is even or odd :

```
#include <stdio.h>
    int main()
{
    int num;     printf("Enter a
number: ");     scanf("%d", &num);
if (num % 2 == 0){         printf("The
number is even.\n");
    }     else{         printf("The number
is odd.\n");
    }
```

```
   return 0;
}
```

Q .2 Check whether a given number is positive or negative :

```
#include <stdio.h>
   int main()
{
   int num; printf("Enter a number: ");
scanf("%d", &num); if (num > 0){
printf("The number is positive.\n");
   }
   else if (num < 0){         printf("The
number is negative.\n");
   }
else
   {
      printf("The number is zero.\n");
   }
   return 0;
}
```

Q .3 Check whether a given year is a leap year or not :

```
#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;
}
```

Q .4 Find the largest of three numbers :

```
#include <stdio.h>
   int main()
```

```
{
   int num1, num2, num3;
printf("Enter three numbers: ");
scanf("%d %d %d", &num1, &num2,
&num3);     if (num1 >= num2 &&
num1 >= num3){         printf("The
largest number is %d.\n", num1);
   }
   else if (num2 >= num1 && num2 >= num3){
printf("The largest number is %d.\n", num2);
   }     else{         printf("The largest number is
%d.\n", num3);
   }
   return 0;
}
```

Q .5 Read temperature and display a message :

```
#include <stdio.h>
   int main()
{
   float temperature;     printf("Enter the
temperature in centigrade: ");     scanf("%f",
&temperature);     if (temperature < 0){
printf("Freezing weather\n");
   }
   else if (temperature >= 0 && temperature <= 10){
printf("Very Cold weather\n");
   }
   else if (temperature > 10 && temperature <= 20){
printf("Cold weather\n");
   }
   else if (temperature > 20 && temperature <= 30){
printf("Normal in Temp\n");
   }
   else if (temperature > 30 && temperature <= 40){
       printf("Its Hot\n");
   }     else{         printf("It's
Very Hot\n");
   }
   return 0;
}
```

Q .6 Display a digit in words :

```
#include <stdio.h>
    int main()
{     int digit;     printf("Enter a
digit (0-9): ");     scanf("%d",
&digit);     switch (digit)
    {
    case 0:
printf("Zero\n");
break;     case 1:
printf("One\n");
break;     case 2:
printf("Two\n");
break;     case 3:
printf("Three\n");
break;     case 4:
printf("Four\n");
break;     case 5:
printf("Five\n");
break;     case 6:
printf("Six\n");
break;     case 7:
printf("Seven\n");
break;     case 8:
printf("Eight\n");
break;     case 9:
printf("Nine\n");
break;     default:
        printf("Invalid digit\n");
    }
```

```
   return 0;
}
```

Q .7 Simple Calculator using switch case:

```
#include <stdio.h> int
main()
{
   char operator;     double num1,
num2, result;     printf("Enter an
operator (+, -, *, /): ");     scanf(" %c",
&operator);     printf("Enter two
numbers: ");     scanf("%lf %lf", &num1,
&num2);     switch (operator)
   {
   case '+':        result =
num1 + num2;        break;
case '-':
      result = num1 - num2;
break;     case '*':
      result = num1 * num2;
break;     case '/':
      result = num1 / num2;
break;     default:
      printf("Invalid operator\n");
return 1; // Exit with an error code
   }
   printf("Result: %.2lf\n", result);
return 0;
}
```

Q .8 Calculate the Area of Rectangle / Circle / Triangle :

```
#include <stdio.h>
   int main()
{
   char choice;     double area;     printf("Choose a shape (R for
Rectangle, C for Circle, T for Triangle): ");     scanf(" %c", &choice);
switch (choice)
   {
   case 'R':
case 'r':
   {
      double length, width;         printf("Enter length
and width of the rectangle: ");         scanf("%lf %lf",
&length, &width);         area = length * width;
break;
   }
   case 'C':
case 'c':
   {
      double radius;         printf("Enter the
radius of the circle: ");         scanf("%lf",
&radius);         area = 3.14159 * radius *
radius;         break;
   }
   case 'T':
case 't':
   {
      double base, height;         printf("Enter the base
and height of the triangle: "); scanf("%lf %lf", &base,
&height);

      area = 0.5 * base * height;
break;
   }
   default:
      printf("Invalid choice\n");
return 1; // Exit with an error code
   }
   printf("Area: %.2lf\n", area);
return 0;
}
```

Q .9 Calculate the sum and average of positive numbers :

```
#include <stdio.h>
   int main()
{
   int num, sum = 0, count = 0;
while (1)
   {
       printf("Enter a number (enter a negative number to stop): ");
scanf("%d", &num);        if (num < 0)
       {
break;
       }
       sum += num;
count++;
   }
   if (count > 0)
   {
       double average = (double)sum / count;
printf("Sum: %d\n", sum);        printf("Average:
%.2lf\n", average);
   }
else
   {
       printf("No positive numbers entered.\n");
   }
   return 0;
}
```

Q .10 Design a digital clock :

```
#include <stdio.h>
   int main()
{
   int hours, minutes, seconds;     printf("Enter time in HH:MM:SS format: ");     scanf("%d:%d:%d", &hours,
&minutes, &seconds);     if (hours >= 0 && hours <= 23 && minutes >= 0 && minutes <= 59 && seconds
>= 0 && seconds <= 59)
   {
       printf("Time entered: %02d:%02d:%02d\n", hours, minutes, seconds);
   }
else
   {
       printf("Invalid time format.\n");
return 1; // Exit with an error code
   }
   // Update time every second for a simulated clock
   while (1)
   {
       printf("%02d:%02d:%02d\n", hours, minutes, seconds);         sleep(1); // Sleep for 1 second
(Note: sleep() function may vary depending on the system)         seconds++;         if (seconds
== 60)
       {
           seconds = 0;
minutes++;             if
(minutes == 60)
           {
               minutes = 0;
hours++;                 if
(hours == 24)
               {
hours = 0;
               }
           }
       }
   }
   return 0;
}
```

Q .11 Find the sum of digits of a number until a single digit is occurred :

```
#include <stdio.h>
    int main()
{
    int num, sum = 0;
printf("Enter a number: ");
scanf("%d", &num);     while
(num > 0 || sum > 9)
```

```
   {
       if (num == 0)
       {
           num = sum;
sum = 0;
       }
       sum += num % 10;
num /= 10;
   }
   printf("Sum of digits until a single digit is occurred: %d\n", sum);
return 0;
}
```

WEEK 4

Q .1 Print multiplication table of a number :

```
#include <stdio.h>
   int main()
{
   int num;     printf("Enter a number: ");
scanf("%d", &num);     printf("Multiplication
table of %d:\n", num);     for (int i = 1; i <= 10;
++i)
   {
       printf("%d x %d = %d\n", num, i, num * i);
   }
   return 0;
}
```

Q .2 Calculate factorial of a number :

```
#include <stdio.h>
   int main()
```

```
{
   int num;     long long
factorial = 1;     printf("Enter
a number: ");     scanf("%d",
&num);     if (num < 0)
   {
       printf("Factorial is not defined for negative numbers.\n");
   }
else
   {
       for (int i = 1; i <= num; ++i)
       {
factorial *= i;
       }
       printf("Factorial of %d = %lld\n", num, factorial);
   }
   return 0;
}
```

Q .3 Check whether a number is palindrome or not :

```
#include <stdio.h>
   int main()
{
   int num, originalNum, reversedNum = 0, remainder;
printf("Enter a number: ");     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);
return 0;
}
```

Q .4 Count frequency of digits in each number :

```
#include <stdio.h>
   int main()
{
   int num, digit;     int frequency[10] = {0}; // Array to store the
frequency of each digit     printf("Enter a number: ");     scanf("%d",
&num);     while (num != 0)
   {
      digit  =  num  %  10;
frequency[digit]++;
num /= 10;
   }
   printf("Digit Frequency:\n");
for (int i = 0; i < 10; ++i)
   {
      if (frequency[i] > 0)
      {
         printf("%d: %d times\n", i, frequency[i]);
      }
   }
   return 0;
}
```

Q .5 Find HCF(GCD) and LCM of two numbers :

```
#include <stdio.h>
   int main()
{
   int num1, num2, temp, gcd, lcm;
printf("Enter two numbers: ");     scanf("%d
%d", &num1, &num2);
   // Find GCD
while (num2 != 0)
   {
        temp = num2;
num2 = num1 % num2;
num1 = temp;
    }
    gcd = num1;     // Find LCM
lcm = (num1 * num2) / gcd;
printf("GCD: %d\n", gcd);
printf("LCM: %d\n", lcm);
return 0;
}
```

Q .6 Print all prime numbers between 1 to n :

```
#include <stdio.h>
int isPrime(int num)
{
    if (num <= 1)
    {
        return 0; // Not prime
    }
    for (int i = 2; i * i <= num; ++i)
    {
        if (num % i == 0)
        {
            return 0; // Not prime
        }
    }
    return 1; // Prime
}

int main()
{     int n;     printf("Enter the value of n: ");
scanf("%d", &n);     printf("Prime numbers between
1 and %d:\n", n);
    for (int i = 2; i <= n; ++i)
    {        if
(isPrime(i))
```

```
        {
printf("%d ", i);
        }
    }
    return 0;
}
```

Q .7 Print Fibonacci series up to n terms :

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

Q .8 Print Armstrong numbers from 1 to n and check a given number is Armstrong or not :

```
#include <stdio.h> #include
<math.h>
    int isArmstrong(int num)
{
    int originalNum, digit, n = 0, result = 0;
originalNum = num;     while
(originalNum != 0)
    {
        originalNum /= 10;
        ++n;
    }
    originalNum = num;
while (originalNum != 0)
    {
        digit = originalNum % 10;
result += pow(digit, n);
originalNum /= 10;
    }
    return (result == num);
}

int main()
{     int n;     printf("Enter the value of n: ");     scanf("%d",
&n);     printf("Armstrong numbers between 1 and
%d:\n", n);
    for (int i = 1; i <= n; ++i)
    {
        if (isArmstrong(i))
        {
printf("%d ", i);
        }
    }
    printf("\nEnter a number to check if it's Armstrong: ");
scanf("%d", &n);     if (isArmstrong(n))
    {
        printf("%d is an Armstrong number.\n", n);
    }
else
    {
        printf("%d is not an Armstrong number.\n", n);
    }
    return 0;
}
```

Q .9 Print all Perfect numbers between 1 to n and check a given number is Perfect or not :

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

```
    int
    isPerfect(int num)
{
    int sum = 0;     for (int i = 1; i
<= num / 2; ++i)
    {
        if (num % i == 0)
        {
sum += i;
        }
    }
    return (sum == num);
}

int main()
{     int n;     printf("Enter the value of n: ");
scanf("%d", &n);     printf("Perfect numbers between
1 and %d:\n", n);
    for (int i = 1; i <= n; ++i)
    {        if
(isPerfect(i))
        {
printf("%d ", i);
        }
    }
    printf("\nEnter a number to check if it's Perfect: ");
scanf("%d", &n);     if (isPerfect(n))
    {
        printf("%d is a Perfect number.\n", n);
    }
else
    {
        printf("%d is not a Perfect number.\n", n);
    }
    return 0;
}
```

Q .10 Print all Strong Numbers between 1 to n :

```
#include <stdio.h>
int factorial(int num)
{
  if (num == 0 || num == 1)
  {
return 1;
  }
else
  {
    return num * factorial(num - 1);
  }
}

int isStrong(int num)
{
  int originalNum, digit, sum = 0;
originalNum = num;    while
(originalNum != 0)
  {
    digit = originalNum % 10;
sum += factorial(digit);
originalNum /= 10;
  }
  return (sum == num);
}

int main()
{    int n;    printf("Enter the value of n: ");
scanf("%d", &n);    printf("Strong numbers between
1 and %d:\n", n);
  for (int i = 1; i <= n; ++i)
  {       if
(isStrong(i))
    {
printf("%d ", i);
```

```
        }
    }
    return 0;
}
```

WEEK 5

Q. Write a program to print the following patterns :

A.

```
#include <stdio.h>
    int main()
{     for (int j = 0; j < 4;
j++)
    {
        // no. of columns
for (int i = 0; i < 5; i++)
        {
            // no. of rows
printf("*");
        }
        // new line character
printf("\n");
    }
}
```

B.

```
#include <stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
    {        for (int i = 0; i < 5;
i++)
```

```
        {            printf("%d ",
i + 1);
        }
printf("\n");
    }
}
```

```
C.        #include
<stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
    {        for (int i = 0; i < j + 1;
i++)
        {            printf("%d ",
i + 1);
        }
printf("\n");
    }
}
```

```
D.        #include
<stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
    {
        // no. of columns
for (int i = 0; i < j + 1; i++)
        {
            // no. of rows
printf("%d ", j + 1);
        }
        // new line character
printf("\n");
    }
}
```

E.

```
#include <stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
    {         for (int i = 0; i < j + 1;
i++)
        {
printf("*");
        }
printf("\n");
    }
}
```

F.

```
#include <stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
    {         for (int k = 4; k > j;
k--)
        {
printf(" ");
printf(" ");
        }         for (int i = 0; i < j +
1; i++)
        {             printf("%c ", i
+ 65);
        }
printf("\n");     }

}
```

```
G. #include
<stdio.h> int
main()
```

```
{     for (int j = 0; j < 4;
j++)
   {         for (int i = 0; i < j + 1;
i++)
      {             printf("%d ", i +
1 + j);
      }
printf("\n");
   }
}
```

H.

```
#include <stdio.h> int
main()
{     for (int j = 0; j < 5;
j++)
   {         for (int i = 0; i < j + 1;
i++)
      {             if (i %
2 == 0)
         {
printf("1 ");             }
else             {
printf("0 ");
         }
      }
printf("\n");
   }
}
```

I.

```
#include <stdio.h> int
main()
{     for (int j = 0; j < 4;
j++)
   {         for (int k = 4; k > j;
k--)
```

```
        {
printf(" ");
        }         for (int i = 0; i < j +
1; i++)
        {
printf(" *");
        }
printf("\n");
    }     for (int j = 0; j < 4;
j++)
    {         for (int k = 4; k > j;
k--)
        {
printf(" ");
        }         for (int i = j + 1; i
> 0; i--)
        {
printf(" *");
        }
printf("\n");
    }
}
```

WEEK 6

Q .1 Menu driven program to insert and delete elements of kth position in an array :

```
#include <stdio.h>      void
displayArray(int arr[], int size)
{
    printf("Array elements: ");
for (int i = 0; i < size; ++i)

    {         printf("%d ",
arr[i]);
```

```
    }
    printf("\n");
}

void insertElement(int arr[], int *size, int position, int element)
{     if (*size >=
position)
    {         for (int i = *size; i >=
position; --i)
        {             arr[i] =
arr[i - 1];
        }
        arr[position - 1] = element;
        (*size)++;         printf("Element inserted
successfully.\n");
    }
else
    {
        printf("Invalid position.\n");
    }
}

void deleteElement(int arr[], int *size, int position)
{
    if (*size > 0 && position > 0 && position <= *size)
    {
        for (int i = position - 1; i < *size - 1; ++i)
        {             arr[i] =
arr[i + 1];
        }
        (*size)--;
        printf("Element deleted successfully.\n");
    }
else
    {
        printf("Invalid position.\n");
    }
}
```

```
int main()
{
   int arr[100], size, choice, position, element;
printf("Enter the size of the array: ");     scanf("%d",
&size);

   printf("Enter %d elements: ", size);
for (int i = 0; i < size; ++i)
   {
      scanf("%d", &arr[i]);
   }
   do
   {
      printf("\nMenu:\n");
printf("1. Display Array\n");
printf("2. Insert Element\n");
printf("3. Delete Element\n");
printf("4. Exit\n");         printf("Enter
your choice: ");         scanf("%d",
&choice);         switch (choice)
      {
      case 1:
displayArray(arr, size);
break;         case 2:
         printf("Enter position and element to insert: ");
scanf("%d %d", &position, &element);
insertElement(arr, &size, position, element);
break;         case 3:
         printf("Enter position to delete: ");
scanf("%d", &position);
deleteElement(arr, &size, position);
break;         case 4:
         printf("Exiting the program.\n");
         break;
default:
```

```
            printf("Invalid choice. Please enter a valid option.\n");
        }
    } while (choice != 4);
return 0;
}
```

Q .2 Print the biggest and smallest element in an array :

```
#include <stdio.h>
    int main()
{     int arr[100], size, i;     printf("Enter
the size of the array: ");     scanf("%d",
&size);     printf("Enter %d elements: ",
size);     for (i = 0; i < size; ++i)
    {
        scanf("%d", &arr[i]);
    }
    // Assume the first element is both the smallest and largest
int smallest = arr[0], largest = arr[0];     for (i = 1; i < size; ++i)
    {        if (arr[i] <
smallest)
        {            smallest
= arr[i];
        }        if (arr[i] >
largest)
        {            largest
= arr[i];
        }
    }
    printf("Smallest Element: %d\n", smallest);
printf("Largest   Element:   %d\n",   largest);
return 0;
}
```

Q .3 Program to print the sum and average of an array :

```
#include <stdio.h>
    int main()
```

```
{     int arr[100], size, i, sum = 0;
printf("Enter the size of the array: ");
scanf("%d", &size);     printf("Enter %d
elements: ", size);     for (i = 0; i < size;
++i)
    {
        scanf("%d", &arr[i]);
sum += arr[i];
    }
    double average = (double)sum / size;     printf("Sum
of Array Elements: %d\n", sum);     printf("Average of
Array Elements: %.2lf\n", average);
    return 0;
}
```

Q .4 Program to sort an array using bubble sort :

```
#include <stdio.h>
    void     bubbleSort(int arr[],
int size)
{     int temp, i, j;     for (i =
0; i < size - 1; ++i)
    {         for (j = 0; j < size - i - 1;
++j)
        {             if (arr[j] >
arr[j + 1])
            {
                // Swap the elements if they are in the wrong order
                temp = arr[j];
arr[j] = arr[j + 1];
arr[j + 1] = temp;
            }
        }
    }
}
```

```
int main()
{     int arr[100], size, i;     printf("Enter
the size of the array: ");     scanf("%d",
&size);

    printf("Enter %d elements: ", size);
for (i = 0; i < size; ++i)
    {
        scanf("%d", &arr[i]);
    }
    bubbleSort(arr, size);
printf("Sorted Array: ");     for
(i = 0; i < size; ++i)
    {         printf("%d ",
arr[i]);
    }
    return 0;
}
```

Q .5 Program to search an element using linear search as well as binary search :

```
#include <stdio.h>
    int linearSearch(int arr[], int size, int key)
{     for (int i = 0; i < size;
++i)
    {         if (arr[i] ==
key)
        {
return i;
        }
    }
    return -1; // Element not found
}
int binarySearch(int arr[], int size, int key)
{
    int low = 0, high = size - 1, mid;
while (low <= high)
    {
        mid = (low + high) / 2;
if (arr[mid] == key)
        {
            return mid;
        }
        else if (arr[mid] < key)
        {
            low = mid + 1;
        }
else
        {
            high = mid - 1;
        }
    }
    return -1; // Element not found
}

int main()
{     int arr[100], size, key, result;     printf("Enter the size of the array: ");
scanf("%d", &size);     printf("Enter %d elements (in sorted order for
binary search): ", size);
    for (int i = 0; i < size; ++i)
    {
        scanf("%d", &arr[i]);
    }
    printf("Enter the element to search: ");
scanf("%d", &key);     result =
linearSearch(arr, size, key);     if (result !=
-1)
    {
        printf("Linear Search: Element found at index %d.\n", result);
    }
else
    {
        printf("Linear Search: Element not found.\n");
    }
```

```
   result = binarySearch(arr, size, key);
if (result != -1)
   {
      printf("Binary Search: Element found at index %d.\n", result);
   }
else
   {
      printf("Binary Search: Element not found.\n");
   }
   return 0;
}
```

Q .6 Program to analyze an array of 20 integer inputs :

```
#include <stdio.h>
   int main()
{
   int arr[20], positiveCount = 0, negativeCount = 0, oddCount = 0, evenCount = 0, zeroCount = 0;
printf("Enter 20 integer inputs: ");     for (int i = 0; i < 20; ++i)
   {
      scanf("%d", &arr[i]);
       if (arr[i] >
0)
      {
         positiveCount++;
      }        else if
(arr[i] < 0)
      {
         negativeCount++;
      }
else
      {
         zeroCount++;
      }
      if (arr[i] % 2 == 0)
      {
         evenCount++;
      }
else
```

```
        {
            oddCount++;
        }
    }
    printf("Number of positive numbers: %d\n", positiveCount);
printf("Number of negative numbers: %d\n", negativeCount);
printf("Number of odd numbers: %d\n", oddCount);     printf("Number
of even numbers: %d\n", evenCount);     printf("Number of zeros:
%d\n", zeroCount);
    return 0;
}
```

Q .7 Split an array into middle and store in two different arrays :

```
#include <stdio.h>
    int main()
{
    int arr[10], size = 10, middle = size / 2;     int
firstArray[middle], secondArray[size - middle];
printf("Enter 10 elements: ");     for (int i = 0; i < size;
++i)
    {
        scanf("%d", &arr[i]);
    }
    // Split the array into two
for (int i = 0; i < middle; ++i)
    {        firstArray[i] =
arr[i];
    }
    for (int i = middle; i < size; ++i)
    {
        secondArray[i - middle] = arr[i];
    }
    // Display the initial array
printf("Initial array: ");     for
(int i = 0; i < size; ++i)
```

```
   {        printf("%d ",
arr[i]);
   }
   printf("\n");
   // Display the split arrays
printf("After splitting:\n");
printf("First Array: ");     for (int
i = 0; i < middle; ++i)
   {        printf("%d ",
firstArray[i]);
   }
   printf("\n");     printf("Second
Array: ");     for (int i = 0; i < size -
middle; ++i)
   {
       printf("%d ", secondArray[i]);
   }
   printf("\n");
return 0;
}
```

Q .8 Program to count frequency of each element in an array :

```
#include <stdio.h>
   int main()
{    int arr[20], size, freq[20] = {0};
printf("Enter the size of the array: ");
scanf("%d", &size);     printf("Enter %d
elements: ", size);     for (int i = 0; i <
size; ++i)
   {
       scanf("%d", &arr[i]);
freq[arr[i]]++;
   }
   printf("Element Frequency:\n");
   for (int i = 0; i < 20; ++i)
   {        if (freq[i]
> 0)        {
           printf("%d: %d times\n", i, freq[i]);
```

```
        }
    }
    return 0;
}
```

WEEK 7

Q .1 Program to print row major and column major matrix :

```
#include <stdio.h>
    int main()
{     int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};
    // Row Major Order
printf("Row Major Order:\n");
    for (int i = 0; i < 3; ++i)
    {         for (int j = 0; j < 3;
++j)
        {             printf("%d ",
matrix[i][j]);
        }
printf("\n");
    }
    // Column Major Order
printf("Column Major Order:\n");
    for (int j = 0; j < 3; ++j)
    {         for (int i = 0; i < 3;
++i)
        {             printf("%d ",
matrix[i][j]);
        }
printf("\n");
    }
    return 0;
}
```

Q .2 Program to print the sum of a whole matrix :

```
#include <stdio.h>
   int main()
{     int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};     int sum = 0;

   // Calculate the sum of the matrix
   for (int i = 0; i < 3; ++i)
   {         for (int j = 0; j < 3;
++j)
       {
           sum += matrix[i][j];
       }
   }
   printf("Sum of the whole matrix: %d\n", sum);
return 0;
}
```

Q .3 Program to add and multiply two 3x3 matrices :

```
#include <stdio.h>
   int main()
{     int matrix1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};     int matrix2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3,
2, 1}};     int resultSum[3][3], resultMultiply[3][3];
   // Addition     printf("Matrix
Addition:\n");     for (int i = 0; i
< 3; ++i)
   {         for (int j = 0; j < 3;
++j)
       {
           resultSum[i][j] = matrix1[i][j] + matrix2[i][j];
printf("%d ", resultSum[i][j]);
       }
printf("\n");
```

```
   }
   // Multiplication     printf("Matrix
Multiplication:\n");     for (int i = 0; i
< 3; ++i)
   {        for (int j = 0; j < 3;
++j)
      {            resultMultiply[i][j]
= 0;            for (int k = 0; k < 3;
++k)
         {
            resultMultiply[i][j] += matrix1[i][k] * matrix2[k][j];
         }
         printf("%d ", resultMultiply[i][j]);
      }
printf("\n");
   }
   return 0;
}
```

Q .4 Program to print the sum of all diagonal elements, upper triangular matrix, and lower triangular matrix :

```
#include <stdio.h>
   int main()
{    int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};
   int sumDiagonal = 0, sumUpper = 0, sumLower = 0;
   // Calculate the sum of diagonal, upper triangular, and lower triangular elements
   for (int i = 0; i < 3; ++i)
   {        for (int j = 0; j < 3;
++j)
      {           if
(i == j)
         {
            sumDiagonal += matrix[i][j];
         }
if (i < j)
```

```
            {
                sumUpper += matrix[i][j];
            }
if (i > j)
{
                sumLower += matrix[i][j];
            }
        }
    }
    printf("Sum of Diagonal Elements: %d\n", sumDiagonal);
printf("Sum of Upper Triangular Elements: %d\n", sumUpper);
printf("Sum of Lower Triangular Elements: %d\n", sumLower);
    return 0;
}
```

Q .5 Program to find the frequency of odd and even elements in a matrix :

```
#include <stdio.h>
    int main()
{     int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};     int oddCount = 0, evenCount = 0;
    // Count the frequency of odd and even elements
    for (int i = 0; i < 3; ++i)
    {        for (int j = 0; j < 3;
++j)
        {
            if (matrix[i][j] % 2 == 0)
            {
                evenCount++;
            }
else
            {
                oddCount++;
            }
        }
    }
    printf("Frequency of Odd Elements: %d\n", oddCount);
printf("Frequency of Even Elements: %d\n", evenCount);
    return 0;
}
```

Q .6 Program to find the sum of each row and sum of each column of a matrix : #include <stdio.h>

```
   int main()
{     int matrix[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8,
9}};     int rowSum[3] = {0}, colSum[3] = {0};
   // Calculate the sum of each row and each column
   for (int i = 0; i < 3; ++i)
   {         for (int j = 0; j < 3;
++j)
       {
           rowSum[i] += matrix[i][j];
colSum[j] += matrix[i][j];
       }
   }
   printf("Sum of Each Row:\n");
   for (int i = 0; i < 3; ++i)
   {
      printf("Row %d: %d\n", i + 1, rowSum[i]);
   }
   printf("Sum of Each Column:\n");
   for (int j = 0; j < 3; ++j)
   {
      printf("Column %d: %d\n", j + 1, colSum[j]);
   }
   return 0;
}
```

Q .7 Initialize a 2D array of a 3x3 matrix :

```
#include <stdio.h>
   int main()
{     int matrix[3][3] = {{1, 2, 3}, {2, 3, 4}, {3, 4,
5}};     printf("2D Array of a 3x3 Matrix:\n");
for (int i = 0; i < 3; ++i)
   {         for (int j = 0; j < 3;
++j)
```

```
    {
      printf("%d ", matrix[i][j]);
    }
printf("\n");
  }
  return 0;
}
```

Q .8 Program to check whether the matrix is a diagonal, upper triangular, or lower triangular matrix :

```
#include <stdio.h>
  int main()
{    int matrix[3][3] = {{1, 0, 0}, {0, 2, 0}, {0, 0, 3}};     int
diagonalFlag = 1, upperFlag = 1, lowerFlag = 1;
  // Check if the matrix is diagonal, upper triangular, or lower triangular
  for (int i = 0; i < 3; ++i)
  {        for (int j = 0; j < 3;
++j)
    {            if (i != j &&
matrix[i][j] != 0)
      {
        diagonalFlag = 0;
      }            if (i > j &&
matrix[i][j] != 0)
      {
        upperFlag = 0;
      }            if (i < j &&
matrix[i][j] != 0)
      {
        lowerFlag = 0;
      }
    }
  }
  if (diagonalFlag)
  {
    printf("The matrix is a diagonal matrix.\n");
  }
  else if (upperFlag)
```

```
    {
        printf("The matrix is an upper triangular matrix.\n");
    }
    else if (lowerFlag)
    {
        printf("The matrix is a lower triangular matrix.\n");
    }
else
    {
        printf("The matrix is neither diagonal, upper triangular, nor lower triangular.\n");
    }
    return 0;
}
```

Q .9 Program to check whether the matrix is a sparse matrix or not :

```
#include <stdio.h>
    int main()
{     int matrix[3][3] = {{1, 0, 0}, {0, 0, 0}, {0, 0,
5}};     int zeroCount = 0, nonZeroCount = 0;
    // Count the number of zero and non-zero elements
    for (int i = 0; i < 3; ++i)
    {         for (int j = 0; j < 3;
++j)
        {             if
(matrix[i][j] == 0)
            {
                zeroCount++;
            }
else
            {
                nonZeroCount++;
            }
        }
    }
    // Check if the matrix is sparse
if (zeroCount > nonZeroCount)
```

```
    {
        printf("The matrix is a sparse matrix.\n");
    }
    else
    {
        printf("The matrix is not a sparse matrix.\n");
    }
    return 0;
}
```

WEEK 8

Q .1 Program to create, initialize, and use pointers :

```
#include <stdio.h>
    int main()
{
    int num = 10;
    int *ptr;
    // Create and initialize a pointer
ptr = &num;
    // Use the pointer to access the value
printf("Value   of   num:   %d\n",   *ptr);
return 0;
}
```

Q .2 Program to add two numbers using pointers :

```
#include <stdio.h>
    int main()
{
    int num1, num2, sum;
    int *ptr1, *ptr2;
    // Input two numbers     printf("Enter
two numbers: ");     scanf("%d %d",
&num1, &num2);     // Create pointers
and initialize them     ptr1 = &num1;
ptr2 = &num2;
    // Add the numbers using pointers
    sum = *ptr1 + *ptr2; //
    Display the result
    printf("Sum: %d\n",
    sum);
   return 0;
}
```

Q .3 Program to swap two numbers using pointers :

```
#include <stdio.h>     void
swap(int *a, int *b)
{
   int temp = *a;
*a = *b;
   *b = temp;
}

int main()
{
   int num1, num2;     // Input two numbers
printf("Enter two numbers: ");     scanf("%d %d",
&num1, &num2);     // Display original numbers
printf("Original numbers: %d, %d\n", num1, num2);
   // Swap numbers using pointers     swap(&num1,
&num2);     // Display swapped numbers
printf("Swapped numbers: %d, %d\n", num1, num2);
   return 0;
}
```

Q .4 Program to input and print array elements using pointers :

```
#include <stdio.h>
   int main()
```

```
{     int
arr[5];     int
*ptr;
    // Input elements using pointers
    printf("Enter 5 array elements: ");
    for (ptr = arr; ptr < arr + 5; ++ptr)
    {         scanf("%d",
ptr);
    }
    // Display elements using pointers
printf("Array elements: ");     for (ptr
= arr; ptr < arr + 5; ++ptr)
    {         printf("%d ",
*ptr);
    }
    printf("\n");
return 0;
}
```

Q .5 Program to copy one array to another using pointers :

```
#include <stdio.h>     void copyArray(int *source, int
*destination, int size)
{     for (int i = 0; i < size;
++i)
    {
        *(destination + i) = *(source + i);
    }
}

int main()
{
    int sourceArray[5] = {1, 2, 3, 4, 5};
int destinationArray[5];     int size =
5;
    // Copy array using pointers
copyArray(sourceArray, destinationArray, size);
// Display the original and copied arrays
printf("Source Array: ");     for (int i = 0; i < size; ++i)
    {
        printf("%d ", sourceArray[i]);
    }
    printf("\n");
    printf("Copied Array: ");
    for (int i = 0; i < size; ++i)
    {
        printf("%d ", destinationArray[i]);
    }
    printf("\n");
return 0;
}
```

Q .6 Program to swap two arrays using pointers :

```
#include <stdio.h>     void swapArrays(int
*arr1, int *arr2, int size)
{     for (int i = 0; i < size;
++i)
    {
        int temp = *(arr1 + i);
        *(arr1 + i) = *(arr2 + i);
        *(arr2 + i) = temp;
    }
}

int main()
{     int arr1[5] = {1, 2, 3, 4,
5};     int arr2[5] = {6, 7, 8, 9,
10};     int size = 5;
    // Swap arrays using pointers
swapArrays(arr1, arr2, size);     //
Display the swapped arrays
printf("Swapped Array 1: ");
```

```
  for (int i = 0; i < size; ++i)
  {       printf("%d ",
arr1[i]);
  }
  printf("\n");
printf("Swapped Array 2: ");
for (int i = 0; i < size; ++i)
  {       printf("%d ",
arr2[i]);
  }
  printf("\n");
return 0;
}
```

Q .7 Program to reverse an array using pointers :

```
#include <stdio.h>    void
reverseArray(int *arr, int size)
{    int *start = arr;    int
*end = arr + size - 1;
while (start < end)
  {
      int temp = *start;
*start = *end;
      *end = temp;
      ++start;
      --end;
  }
}

int main()
{    int arr[5] = {1, 2, 3, 4,
5};    int size = 5;
  // Reverse array using pointers
reverseArray(arr, size);    //
Display the reversed array
printf("Reversed Array: ");    for
(int i = 0; i < size; ++i)
```

```
   {        printf("%d ",
arr[i]);
   }
   printf("\n");
return 0;
}
```

Q .8 Program to add two matrices using pointers :

```
#include <stdio.h>     void addMatrices(int *mat1, int *mat2, int
*result, int rows, int cols)
{     for (int i = 0; i < rows;
++i)
   {        for (int j = 0; j < cols;
++j)
       {
           *((result + i * cols) + j) = *((mat1 + i * cols) + j) + *((mat2 + i * cols) + j);
       }
   }
}

int main()
{     int mat1[2][2] = {{1, 2}, {3,
4}};     int mat2[2][2] = {{5, 6},
{7, 8}};     int result[2][2];     int
rows = 2, cols = 2;
   // Add matrices using pointers     addMatrices(&mat1[0][0],
&mat2[0][0], &result[0][0], rows, cols);
   // Display the result matrix
printf("Result Matrix:\n");     for
(int i = 0; i < rows; ++i)
   {        for (int j = 0; j < cols;
++j)
       {           printf("%d ",
result[i][j]);
```

```
        }
printf("\n");
    }
    return 0;
}
```

WEEK 9

Q .1 Program to search a string :

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[50], key[20];     // Input
string and key to search
printf("Enter a string: ");     gets(str);
printf("Enter the search key: ");
gets(key);
    // Search for the key in the string
char *result = strstr(str, key);     if
(result != NULL)
    {
        printf("Key found at index: %d\n", result - str);
    }
else
    {
        printf("Key not found.\n");
    }
    return 0;
}
```

Q .2 Program to reverse words in a string :

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str[100];     //
Input string
printf("Enter a string: ");
gets(str);

   // Reverse words in the string
char *token = strtok(str, " ");     while
(token != NULL)
   {
       strrev(token);
printf("%s ", token);       token
= strtok(NULL, " ");
   }
   return 0;
}
```

Q .3 Program to count vowels, consonants, etc.in a string :

```
#include <stdio.h> #include
<ctype.h>
   int main()
{
   char str[100];     int vowels = 0, consonants = 0, digits
= 0, spaces = 0;
   // Input string
printf("Enter a string: ");
gets(str);
   // Count vowels, consonants, digits, and spaces
   for (int i = 0; str[i] != '\0'; ++i)
   {
       char ch = tolower(str[i]);         if (ch == 'a' || ch == 'e' || ch
== 'i' || ch == 'o' || ch == 'u')
```

```
        {
            vowels++;
        }
        else if (isalpha(ch))
        {
            consonants++;
        }
        else if (isdigit(ch))
        {
digits++;
        }
        else if (isspace(ch))
        {
            spaces++;
        }
    }
    printf("Vowels: %d\n", vowels);
printf("Consonants: %d\n", consonants);
printf("Digits: %d\n", digits);
printf("Spaces: %d\n", spaces);     return 0;
}
```

Q .4 Program to separate characters in a given string :

```
#include <stdio.h>
#include <string.h>
int main()
{     char str[50];     //
Input string
printf("Enter a string: ");
gets(str);
    // Separate characters in the string
    for (int i = 0; str[i] != '\0'; ++i)
    {        printf("%c ",
str[i]);
    }
    return 0;
}
```

Q .5 Program to concatenate two strings and add a space between them :

```
#include <stdio.h>
#include <string.h>
int main()
{
   char str1[50], str2[50];     //
Input two strings
printf("Enter the first string: ");
gets(str1);
   printf("Enter the second string: ");
gets(str2);
   // Concatenate the strings with a space in between
   strcat(str1, " ");     strcat(str1, str2);     //
Display the result     printf("Concatenated
String: %s\n", str1);     return 0;
}
```

Q .6 Program to toggle the case of characters in a string:

```
#include <stdio.h>
#include <string.h> int
main()
{
   char str[100];     //
Input string
printf("Enter a string: ");
gets(str);
   // Toggle the case of characters in the string
   for (int i = 0; str[i] != '\0'; ++i)
   {        if
(isupper(str[i]))
      {           str[i] =
tolower(str[i]);
      }        else if
(islower(str[i]))
      {           str[i] =
toupper(str[i]);
```

```
        }
    }
    //        Display        the        result
printf("Toggled  Case:  %s\n",  str);
return 0;
}
```

Q .7 Program to check if two strings are identical without using string functions : #include <stdio.h>   int compareStrings(char str1[], char str2[])

```
{     int i = 0;     while (str1[i] != '\0' | |
str2[i] != '\0')
    {        if (str1[i] != str2[i]){
return 0; // Not identical
        }
i++;
    }
    return 1;
}

int main()
{
    char str1[50], str2[50];     // Input
two strings     printf("Enter the first
string: ");     gets(str1);
printf("Enter the second string: ");
gets(str2);
    // Check if the strings are identical
if (compareStrings(str1, str2))
    {
        printf("Identical\n");
    }
else
    {
        printf("Not Identical\n");
    }
    return 0;
}
```

Q .8 Program to sort a list of student names in alphabetical order :

```
#include <stdio.h> #include <string.h>
void sortStrings(char names[][50], int n) {
    char temp[50];
   for (int i = 0; i < n - 1; ++i)
   {        for (int j = i + 1; j < n;
++j)
       {
           if (strcmp(names[i], names[j]) > 0)
           {
               strcpy(temp, names[i]);
strcpy(names[i], names[j]);
strcpy(names[j], temp);
           }
       }
   }
}

int main()
{    int
n;
   // Input number of students
printf("Enter the number of students: ");
scanf("%d", &n);     char
studentNames[n][50];     // Input student
names     printf("Enter the names of
students:\n");
   for (int i = 0; i < n; ++i)
   {
       scanf("%s", studentNames[i]);
   }
   // Sort student names in alphabetical order
sortStrings(studentNames, n);     // Display the
sorted names     printf("Sorted Names:\n");
   for (int i = 0; i < n; ++i)
   {
       printf("%s\n", studentNames[i]);
   }
   return 0;
}
```

WEEK 10

Q .1 Program to find the length of a string using pointers :

```
#include <stdio.h>     int
findLength(char *str)
{
   int length = 0;
while (*str != '\0')
   {
      length++;
str++;
   }
   return length;
}

int main()
{
   char str[100];     //
Input string
printf("Enter a string: ");
gets(str);
   // Find and display the length using pointers
printf("Length of the string: %d\n", findLength(str));
return 0;
}
```

Q .2 Program to copy one string to another using pointers :

```
#include <stdio.h>     void copyString(char
*source, char *destination)
{
   while (*source != '\0')
   {
      *destination = *source;
source++;        destination++;

   }
   *destination = '\0'; // Null-terminate the destination string
}

int main()
{
   char source[50], destination[50];
// Input string to be copied
printf("Enter a string to be copied: ");
gets(source);
   // Copy the string using pointers
copyString(source, destination);    //
Display the copied string    printf("Copied
string: %s\n", destination);    return 0;
}
```

Q .3 Program to concatenate two strings using pointers :

```
#include <stdio.h>    void
concatenateStrings(char *str1, char *str2)
{    while (*str1 !=
'\0')
   {
str1++;
   }
   while (*str2 != '\0')
   {
      *str1 = *str2;
str1++;        str2++;
   }
   *str1 = '\0'; // Null-terminate the concatenated string
}
```

```
int main()
{
   char str1[50], str2[50];
   // Input two strings
   printf("Enter the first
   string: ");     gets(str1);
   printf("Enter the
   second string: ");
   gets(str2);
   // Concatenate the strings using pointers
concatenateStrings(str1, str2);     // Display
the concatenated string
printf("Concatenated string: %s\n", str1);
return 0;
}
```

Q .4 Program to compare two strings using pointers :

```
#include <stdio.h>     int
compareStrings(char *str1, char *str2)
{
   while (*str1 != '\0' && *str2 != '\0')
   {        if (*str1 !=
*str2)
       {
           return 0; // Not equal
       }
str1++;
str2++;
   }
   if (*str1 == '\0' && *str2 == '\0')
   {
       return 1; // Equal
   }
   return 0; // Not equal
}
```

```
int main()
{
   char str1[50], str2[50];     // Input
two strings     printf("Enter the first
string: ");     gets(str1);
printf("Enter the second string: ");
gets(str2);
   // Compare the strings using pointers
if (compareStrings(str1, str2))
   {
       printf("Strings are equal.\n");
   }
else
   {
       printf("Strings are not equal.\n");
   }
   return 0;
}
```

Q .5 Program to find the largest among three numbers using pointers :

```
#include <stdio.h>     void findLargest(int *num1,
int *num2, int *num3)
{
   int *max = num1;
if (*num2 > *max)
   {
       max = num2;
   }
   if (*num3 > *max)
   {
       max = num3;
   }
   printf("Largest number: %d\n", *max);
}

int main()
```

```
{
   int num1, num2, num3;     // Input three
numbers     printf("Enter three numbers: ");
scanf("%d %d %d", &num1, &num2, &num3);
   // Find and display the largest using pointers findLargest(&num1,
   &num2, &num3);
   return 0;
}
```

Q .6 Program to find the largest among three numbers using pointers :

```
#include <stdio.h>     void findLargest(int *num1,
int *num2, int *num3)
{
   int *max = num1;
if (*num2 > *max)
   {
      max = num2;
   }
   if (*num3 > *max)
   {
      max = num3;
   }
   printf("Largest number: %d\n", *max);
}

int main()
{
   int num1, num2, num3;     // Input three
numbers     printf("Enter three numbers: ");
scanf("%d %d %d", &num1, &num2, &num3);
// Find and display the largest using pointers
findLargest(&num1, &num2, &num3);
   return 0;
}
```

Q .7 Program to find the factorial of a number using pointers :

```
#include <stdio.h>     void findFactorial(int num,
long long *factorial)
```

```
{
    *factorial = 1;     for (int i =
1; i <= num; ++i)
    {
        *factorial *= i;
    }
}

int main()
{
    int num;     long long factorial;     // Input a
number     printf("Enter a number: ");
scanf("%d", &num);     findFactorial(num,
&factorial);     printf("Factorial of %d: %lld\n",
num, factorial);     return 0;
}
```

Q .8 Program to print the largest even number present in an array using a pointer to an array :

```
#include <stdio.h>     void
findLargestEven(int *arr, int size)
{
    int *maxEven = NULL;
    for (int i = 0; i < size; ++i)
    {         if (*(arr + i) % 2
== 0)
        {
            if (maxEven == NULL || *(arr + i) > *maxEven)
            {
                maxEven = (arr + i);
            }
        }
    }
    if (maxEven != NULL)
    {
        printf("Largest even number: %d\n", *maxEven);
```

```
   }
else
   {
      printf("No even numbers in the array.\n");
   }
}

int main()
{    int
size;
   // Input the size of the array
printf("Enter the size of the array: ");
scanf("%d", &size);    int arr[size];
   // Input array elements
printf("Enter the array elements:\n");
   for (int i = 0; i < size; ++i)
   {
      scanf("%d", &arr[i]);
   }
   // Find and display the largest even number using a pointer to an array
findLargestEven(arr, size);    return 0; }
```

Q .9 Program to find the sum of elements of an array using an array of pointers :

```
#include <stdio.h>
   void    findSum(int *arr[],
int size)
{
   int sum = 0;    for (int i =
0; i < size; ++i)
   {
      sum += *arr[i];
   }
   printf("Sum of elements: %d\n", sum);
}

int main()
{ int size;
```

```
  // Input the size of the array printf("Enter
  the size of the array: "); scanf("%d",
  &size);
  int arr[size];
  // Input array elements
printf("Enter the array elements:\n");
  for (int i = 0; i < size; ++i)
  {
     scanf("%d", &arr[i]);
  }
  // Create an array of pointers to array elements
  int *ptrArr[size];     for
(int i = 0; i < size; ++i)
  {        ptrArr[i] =
&arr[i];
  }
  // Find and display the sum using an array of pointers
findSum(ptrArr, size);     return 0;
}
```

Q .10 Program to compute simple interest using pointers :

```
#include <stdio.h>     void computeSimpleInterest(float *principal, float *rate, float
*time, float *interest)
{
  *interest = (*principal * *rate * *time) / 100.0;
}

int main()
{
  float principal, rate, time, interest;     //
Input principal amount, rate, and time
printf("Enter principal amount: ");
scanf("%f", &principal);     printf("Enter rate
of interest: ");     scanf("%f", &rate);
printf("Enter time (in years): "); scanf("%f",
&time);
```

```
    // Compute and display simple interest using pointers
    computeSimpleInterest(&principal, &rate, &time, &interest);
    printf("Simple Interest: %.2f\n", interest);

   return 0;
}
```

Q .11 Program to print the largest even number present in an array using a pointer to an array :

```
#include <stdio.h>     void
findLargestEven(int *arr, int size)
{
   int *maxEven = NULL;
   for (int i = 0; i < size; ++i)
   {        if (*(arr + i) % 2
== 0)
       {
           if (maxEven == NULL || *(arr + i) > *maxEven)
           {
               maxEven = (arr + i);
           }
       }
   }
   if (maxEven != NULL)
   {
       printf("Largest even number: %d\n", *maxEven);
   }
else
   {
       printf("No even numbers in the array.\n");
   }
}

int main()
{    int
size;
   // Input the size of the array
printf("Enter the size of the array: ");
scanf("%d", &size); int arr[size];
```

```
    // Input array elements printf("Enter
    the array elements:\n");
    for (int i = 0; i < size; ++i)
    {        scanf("%d",
&arr[i]);
    }
    // Find and display the largest even number using a pointer to an array
findLargestEven(arr, size);     return 0;
}
```

WEEK 11

Q .1 Function to return the maximum of three integers :

```
#include <stdio.h>

    int findMax(int num1, int num2, int num3)
{
    int max = num1;
if (num2 > max)
    {
        max = num2;
    }
    if (num3 > max)
    {
        max = num3;
    }
    return max;
}

int main()
{
    int num1, num2, num3;     // Input three numbers     printf("Enter
three numbers: ");     scanf("%d %d %d", &num1, &num2, &num3);
// Find and display the maximum using the function
printf("Maximum number: %d\n", findMax(num1, num2, num3));
    return 0;
}
```

Q .2 Function to check if a given number is prime or not :

```
#include <stdio.h>
int isPrime(int num)
{
    if (num <= 1)
    {
        return 0; // Not prime
    }
    for (int i = 2; i * i <= num; ++i)
    {
        if (num % i == 0)
        {
            return 0; // Not prime
        }
    }
    return 1; // Prime
}

int main()
{
    int num;     // Input a
number     printf("Enter a
number: ");     scanf("%d",
&num);
    // Check and display if the number is prime using the function
if (isPrime(num))
    {
        printf("%d is prime.\n", num);
    }
else
    {
        printf("%d is not prime.\n", num);
```

```
   }
   return 0;
}
```

Q .3 Function to compute the factorial of a non - negative integer : #include <stdio.h> long long computeFactorial(int num)

```
{
   long long factorial = 1;
for (int i = 1; i <= num; ++i)
   {        factorial
*= i;
   }
   return factorial;
}

int main()
{
   int num;
   // Input a non-negative integer
printf("Enter a non-negative integer: ");
scanf("%d", &num);
   // Compute and display the factorial using the function
printf("Factorial of %d: %lld\n", num, computeFactorial(num));
   return 0;
}
```

Q .4 Function to swap the values of two integers in actual arguments :

```
#include <stdio.h>
   void    swap(int
*a, int *b)
{
   int temp = *a;
*a = *b;
   *b = temp;
}

int main()
```

```
{
   int num1, num2;     // Input
two numbers     printf("Enter
two numbers: ");     scanf("%d
%d", &num1, &num2);     //
Display original numbers
printf("Original numbers: %d,
%d\n", num1, num2);
   // Swap numbers using the function
swap(&num1, &num2);     // Display swapped
numbers     printf("Swapped numbers: %d, %d\n",
num1, num2);
   return 0;
}
```

Q .5 Function to compute the sum and average of an array of integers :

```
#include <stdio.h>     void computeSumAndAverage(int arr[], int size, int
*sum, double *average)
{
   *sum = 0;     for (int i =
0; i < size; ++i)
   {
      *sum += arr[i];
   }
   *average = (double)(*sum) / size;
}

int main()
{     int arr[] = {1, 2, 3, 4, 5};     int size = sizeof(arr) /
sizeof(arr[0]);     int sum;     double average;
computeSumAndAverage(arr, size, &sum, &average);
printf("Sum: %d\n", sum);     printf("Average: %.2lf\n",
average);     return 0;
}
```

Q .6 C function to find the GCD(Greatest Common Divisor) of two nonnegative integers using Euclid's algorithm:

```
#include <stdio.h>     int
findGCD(int a, int b) {
   while (b != 0)
   {
       int temp = b;
b = a % b;         a
= temp;
   }
   return a;
}

int main()
{
   int num1, num2;

   printf("Enter two nonnegative integers: ");
scanf("%d %d", &num1, &num2);     int gcd
= findGCD(num1, num2);     printf("GCD:
%d\n", gcd);     return 0;
}
```

Q .7 C function to check if a given string is a valid palindrome, considering only alphanumeric characters and ignoring cases :

```
#include <stdio.h>
#include <ctype.h>
#include <stdbool.h> #include
<string.h>     bool
isAlphanumeric(char ch)
{
   return isalnum(ch) != 0;
}
bool isValidPalindrome(char str[])
{     int i = 0;     int j =
strlen(str) - 1;
```

```
    while (i < j)
    {
        while (i < j && !isAlphanumeric(str[i]))
        {
i++;
        }
        while (i < j && !isAlphanumeric(str[j]))
        {
j--;
        }
        if (tolower(str[i]) != tolower(str[j]))
        {             return
false;
        }
i++;
j--;
    }
    return true;
}

int main()
{     char str[50];
printf("Enter a string: ");
gets(str);     if
(isValidPalindrome(str))
    {
        printf("Valid Palindrome\n");
    }
else
    {
        printf("Not a Valid Palindrome\n");
    }
    return 0;
}
```

Q .8 C function to calculate the sum and difference of two complex numbers :

```
#include <stdio.h>
typedef struct
{
   double real;
   double imag; } Complex; void addComplex(Complex num1,
Complex num2, Complex *result)
{
   result->real = num1.real + num2.real;     result-
>imag = num1.imag + num2.imag;
}
void subtractComplex(Complex num1, Complex num2, Complex *result)
{
   result->real = num1.real - num2.real;     result-
>imag = num1.imag - num2.imag;
}

int main()
{
   Complex num1, num2, sum, diff;     printf("Enter the real and imaginary parts
of the first complex number: ");     scanf("%lf %lf", &num1.real, &num1.imag);
printf("Enter the real and imaginary parts of the second complex number: ");
scanf("%lf %lf", &num2.real, &num2.imag);     addComplex(num1, num2,
&sum);     subtractComplex(num1, num2, &diff);     printf("Sum: %.2lf + %.2lfi\n",
sum.real, sum.imag);     printf("Difference: %.2lf + %.2lfi\n", diff.real, diff.imag);
return 0;
}
```

H.O.T.S Questions :

Q .9 C function to find the second largest and second smallest elements in an array of integers :

```
#include <stdio.h>     void findSecondLargestAndSmallest(int arr[], int size, int *secondLargest,
int *secondSmallest)
{
   // Initialize with the maximum and minimum possible values
   *secondLargest = *secondSmallest = arr[0];
   for (int i = 1; i < size; ++i)
```

```
    {
        if (arr[i] > *secondLargest)
        {
            *secondLargest = arr[i];
        }
        else if (arr[i] < *secondSmallest)
        {
            *secondSmallest = arr[i];
        }
    }
}

int main()
{     int arr[] = {5, 3, 8, 1, 6, 2, 7, 4};
int size = sizeof(arr) / sizeof(arr[0]);
int secondLargest, secondSmallest;

    findSecondLargestAndSmallest(arr, size, &secondLargest, &secondSmallest);
printf("Second Largest: %d\n", secondLargest);     printf("Second Smallest:
%d\n", secondSmallest);
    return 0;
}
```

Q .10 C function to find the number of occurrences of each unique element in an array :

```
#include <stdio.h>
    void     countOccurrences(int arr[],
int size)
{     for (int i = 0; i < size;
++i)
    {
        int count = 1;
        // Check if the element is already counted
        if (arr[i] != -1)
        {              for (int j = i + 1; j <
size; ++j)
            {                  if (arr[i]
== arr[j])
```

```
            {                 count++;                 arr[j] = -1;
// Mark the element as counted              }

         }
         printf("Element %d occurs %d times\n", arr[i], count);
      }
   }
}

int main()
{    int arr[] = {1, 2, 3, 2, 4, 1, 5, 2};
int size = sizeof(arr) / sizeof(arr[0]);
countOccurrences(arr, size);
return 0;
}
```

END SEMESTER PROJECT

Q. Write a program for a voting system that takes name, age & gender into considerations and checks eligibility to vote also gives out result at the end of voting.

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <windows.h>

int main()
{
   char gender, choice = 'Y', vote;    char name[25];
char lastname[25];    int age;    int vote1 = 0, vote2 = 0,
vote3 = 0, vote4 = 0, vote5 = 0;    while (choice == 'Y')
   {
   tryagain:
      printf("\aPlease choose your gender (M) or (F): ");
scanf("%s", &gender);
```

```
        gender = toupper(gender);
        if (gender == 'M')
        {
            printf("\nMale\n");
        }
        else if (gender == 'F')
        {
            printf("\nFemale\n");
        }
else
        {
            printf("\nPlease enter a valid gender.\n");
goto tryagain;
        }

        printf("\n\aEnter your first name: ");
scanf(" %s", &name);         printf("\n\aEnter
your last name: ");         scanf(" %s",
&lastname);

        name[0] = toupper(name[0]);
lastname[0] = toupper(lastname[0]);

        if (gender == 'M')
        {
            printf("\nOkay, Mister %s %s\n", name, lastname);
        }
        else if (gender == 'F')
        {
            printf("\nOkay, Miss %s %s\n", name, lastname);
        }
else
        {
            printf("\nOkay, Mister/Miss %s %s\n", name, lastname);
        }

        printf("\n\aWhat's your age? ");
scanf("%d", &age);
```

```
   jump:
       if (age >= 18)
       {
           printf("\nYou're eligible to vote, let's continue.\n");

       move:
           printf("\nChoose which party you would like to vote:\n\n
A. BJP\n\n
B.  AAP\n\n
C. Congress\n\n
D. Samajvadi Party\n\n
E.  BSP\n\n
          Please select\t(A)  (B)  (C)  (D)  (E) :  ");

           scanf(" %c", &vote);
vote = toupper(vote);
switch (vote)
           {           case 'A':               vote1++;               break;
case 'B':               vote2++;               break;           case 'C':
vote3++;               break;           case 'D':               vote4++;
break;           case 'E':               vote5++;               break;
default:               Beep(550, 400);               printf("\nInvalid option,
please choose a valid option.\n");               goto move;
break;
           }
       }
       else if (age <= 0 || age > 150)
       {
           Beep(550, 400);           printf("\nError, please
enter a valid age.\n");           printf("\nWhat's your
age? ");           scanf(" %d", &age);           goto
jump;
       }
else
       {
          Beep(550, 400);            printf("\nSorry, you're
not eligible to vote.\n");
      }

      printf("\nDo you want to vote again? (Y) or (N): ");
scanf(" %c", &choice);        choice = toupper(choice);
   }

   printf("\a\nVoting has ended, here are the results:\n\n
         BJP: %d\n\n
         AAP: %d\n\n
         Congress: %d\n\n
         Samajvadi Party: %d\n\n
         BSP: %d\n\n", vote1, vote2, vote3, vote4, vote5);

   if (vote1 > vote2 > vote3 > vote4 > vote5)
   {
      printf("Narendra Modi (BJP) is the winner!");
   }
   else if (vote2 > vote1 > vote3 > vote4 > vote5)
   {
      printf("Kejariwal (AAP) is the winner!");
   }
   else if (vote3 > vote2 > vote1 > vote4 > vote5)
   {
      printf("Rahul Gandhi (Congress) is the winner!");
   }
   else if (vote4 > vote2 > vote3 > vote1 > vote5)
   {
      printf("Samajvadi party is the winner!");
   }
   else if (vote5 > vote2 > vote3 > vote4 > vote1)
   {
      printf("Mayawati (BSP) is the winner!");
   }
else
   {
      printf("It's a tie or there were not enough votes!");
   }
   printf("\n\n");
return 0;
}
```

END SEMESTER PROJECT 2

Q. Write a program for a quiz which also has an embedded timer so that user completes the quiz on time and show the results at the end.

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <windows.h>
#include <time.h>
#include <stdlib.h>

int main()
{
   int marks = 0, i = 0, ongoing = 1;
char ans, navigate;
srand(time(NULL));    time_t
startTime, currentTime;    int
elapsedTime, totalTime = 60;
startTime = time(NULL);

   printf("Welcome to the quiz. Let's begin!\n");

   char *questions[] = {
     "\nQ1. What is the capital of India?\n",
      "\nQ2. Which one is correct?\n",
      "\nQ3. Who is the father of C Programming?\n",
      "\nQ4. What is the largest planet in our solar system?\n",
        "\nQ5. What is the name of C Compiler.\n"};

    char *answers[] = {
        "A. Delhi\t\tB. Mumbai\nC. Banglore\t\tD. Jaipur\n",
        "A. int a : 1;\t\tB. a = 0\nC. int a: 2\t\tD. int a = 5;\n",
        "A. Bennett\t\tB. Ritchie\nC. Nikola\t\tD. Hawking\n",
        "A. Mercury\t\tB. Neptune\nC. Jupiter\t\tD. Sun\n",
        "A. G++\t\tB. VSCode\nC. GCC\t\tD. DevC++\n"};

    char *correctAnswers[] = {"A", "D", "B", "C", "C"};
int calculateMarks[] = {0, 0, 0, 0, 0};

    while (ongoing)
    {
retry:
        printf(questions[i]);
printf(answers[i]);
scanf(" %c", &ans);         ans
= toupper(ans);

        currentTime  =  time(NULL);               elapsedTime  =
(int)difftime(currentTime, startTime);        if (elapsedTime
>= totalTime)
        {
            printf("\nTime's up! Quiz ended.\n");
ongoing = 0;            break;
        }

        printf("\n");

        if (ans == 'A' || ans == 'B' || ans == 'C' || ans == 'D')
        {
            if (ans == *correctAnswers[i])
            {
                calculateMarks[i] = 1;
            }
```

```
      Beep(550, 400);          printf("Time Remaining: %d seconds\n\n",
totalTime - elapsedTime);
       if (i <
4)
      {
      tryagain:
        printf("Press 1 for next question, 2 for previous question, or 0 to exit the quiz.\n");
if (elapsedTime >= totalTime)
        {
          printf("\nTime's up! Quiz ended.\n");
ongoing = 0;                 break;
        }
        scanf("  %c", &navigate);

        switch (navigate)
        {
case '1':
if (i < 5)
          {
i++;                }
break;
case '2':
if (i > 0)
          {
i = i;               }
break;              case
'0':                ongoing
= 0;                 break;
default:
          printf("\nPlease enter a valid value (1, 2 or 0).\n\n");
          goto tryagain;
        }
}
else
      {
        ongoing = 0;
```

```
            }
}
else
        {
            printf("Please press a valid key (A, B, C or D).\n");
goto retry;
        }
    };

    for (int i = 0; i < 5; i++)
    {
        if (calculateMarks[i] == 1)
        {
            marks++;
        }
    }

    if (marks >= 2)
    {
        printf("Congratulations! You passed the quiz.\n");
    }

    else
    {
        printf("\nSorry, you did not pass the quiz. Please try again.\n");
    }

    printf("\nYour score: %d/5\n\n", marks);
return 0;
}
```