PRACTICAL FILE

OF

“Introduction to Programming Methodology using C ”

(13060112)

SGT UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

Submitted to:

Ms. Simmi Madaan

Assistant professor, CSE

FEAT

Submitted by

Name: Aman

Reg. no.:241306050

BCA 1st Sem (AI/ML)

INDEX

S.No	Name of the Program	Date	Page No.	Signature
1	Program to Convert Celsius to Fahrenheit	16/09/2024	1	
2	Program to Find the Greatest Number Among Three Numbers	16/09/2024	2	
3	Program to Determine the Size of Data Types in C	23/09/2024	3	
4	Program to Check Whether a Character is a Vowel or Consonant	23/09/2024	4	
5	Program to Display an Inverse Triangle Pattern	23/09/2024	5	
6	Program to Display a Pyramid Pattern	30/09/2024	6	
7	Program to Calculate the Average Marks of 50 Students	30/09/2024	7-8	
8	Program to Find the Factorial of a Number Using a Function	30/09/2024	9-10	
9	Program to Count Characters in a String	07/09/2024	11	
10	Program to Find the Smallest Value Among Three Numbers	07/10/2024	12	
11	Program to Calculate Compound Interest	07/10/2024	13	
12	Program to Find the Area of a Circle	14/10/2024	14	
13	Program to Check if a Year is a Leap Year	14/10/2024	15	
14	Program to Calculate Simple Interest	14/10/2024	16	
15	Program to Display Sizes of Different Data Types	28/10/2024	17	

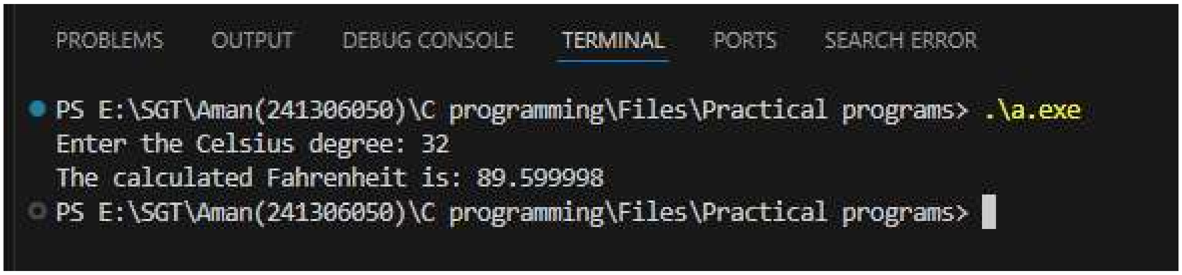
16	Program to Find the Sum of Natural Numbers up to N	28/10/2024	18	
17	Program to Print Multiplication Tables up to N	04/11/2024	19	
18	Program to Swap Two Numbers Using Call by Value and Call by Reference	11/11/2024	20	
19	Program to Sort an Array in Ascending Order Using Bubble Sort	11/11/2024	21	
20	Program to Perform Matrix Addition for 2x2 Arrays	25/112024	22	
21	Program to Find the Transpose of a Matrix for 2x2 Arrays	25/11/2024	23	
22	Program to Calculate Factorial Using Recursion	02/12/2024	24	
23	Program to Print Fibonacci Series	02/12/2024	25	
24	Program to Perform Matrix Multiplication for 2x2 Arrays	02/12/2024	26-27	
25	Menu-driven program for string operations Basic operations	16/12/2024	28-30	
26	Program to perform Array of Pointers Displaying a List of Fruits	16/12/2024	31	
27	Program to Implement Pointer to Array Example Accessing an Integer Array	16/12/2024	32	
28	Program to Implement Binary Search	23/12/2024	33	
29	Program to Perform Insertion in an Array	23/12/2024	34	
30	Program to Implement Linear Search	23/12/2024	35	

Program 1

Write a program to convert temperature from Celsius to Fahrenheit by taking input from the user.

```
#include<stdio.h>
int main() {
float c, f;
printf("Enter the Celsius degree: ");
scanf("%f", &c);
f = (c * 9.0 / 5.0) + 32.0;

printf("The calculated Fahrenheit is: %f\n", f);
return 0;
}
```

OUTPUT

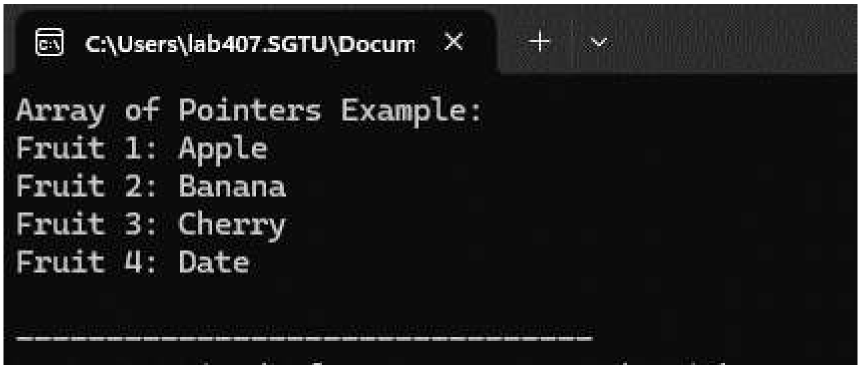


```
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs> .\a.exe
Enter the Celsius degree: 32
The calculated Fahrenheit is: 89.599998
PS E:\SGT\Aman(241306050)\C programming\Files\Practical programs>
```

Program 26
Array of Pointers Example Displaying a List of Fruits

```
#include <stdio.h>
int main() {
  const char *fruits[] = {"Apple", "Banana", "Cherry", "Date"};
  int n = sizeof(fruits) / sizeof(fruits[0]);
  printf("Array of Pointers Example:\n");
  for (int i = 0; i < n; i++) {
    printf("Fruit %d: %s\n", i + 1, fruits[i]);
  }
  return 0;
}
```

OUTPUT

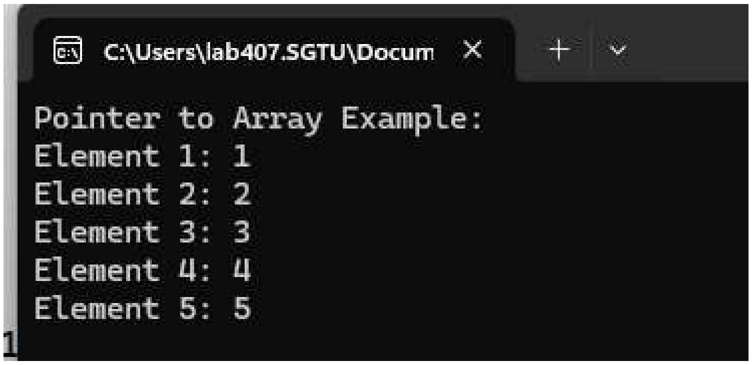


```
Array of Pointers Example:
Fruit 1: Apple
Fruit 2: Banana
Fruit 3: Cherry
Fruit 4: Date

--------------------------------
```

Program 27
Pointer to Array Example Accessing an Integer Array

```
#include <stdio.h>
int main() {
   int numbers[5] = {1, 2, 3, 4, 5};
   int (*ptr)[5] = &numbers;
   printf("Pointer to Array Example:\n");
   for (int i = 0; i < 5; i++) {
      printf("Element %d: %d\n", i + 1, (*ptr)[i]);
   }
   return 0;
}
```

OUTPUT

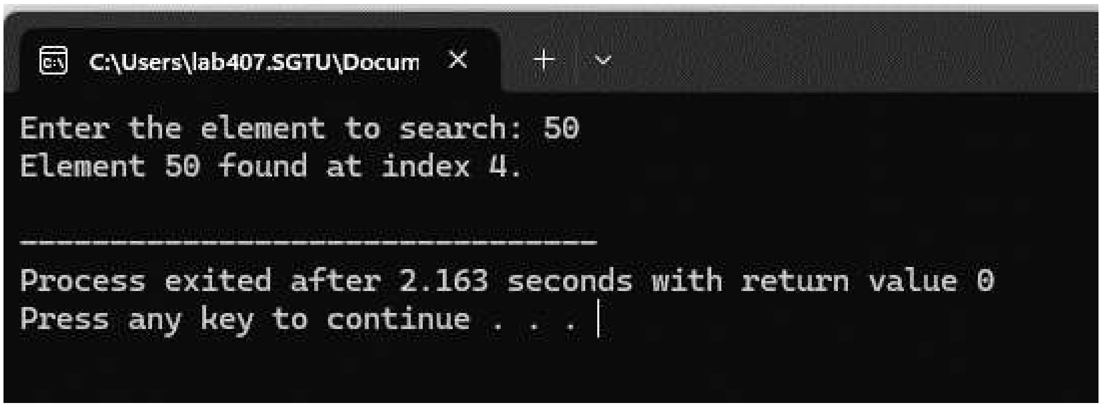


```
Pointer to Array Example:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Element 5: 5
```

Program 28
Program to Implement Binary Search

```
#include <stdio.h>
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;
}
int main() {
   int arr[] = {10, 20, 30, 40, 50, 60, 70, 80, 90};
   int size = sizeof(arr) / sizeof(arr[0]);
   int key;

   printf("Enter the element to search: ");
   scanf("%d", &key);

   int result = binarySearch(arr, size, key);

   if (result != -1) {
      printf("Element %d found at index %d.\n", key, result);
   } else {
      printf("Element %d not found in the array.\n", key);
   }
   return 0;}
```

OUTPUT

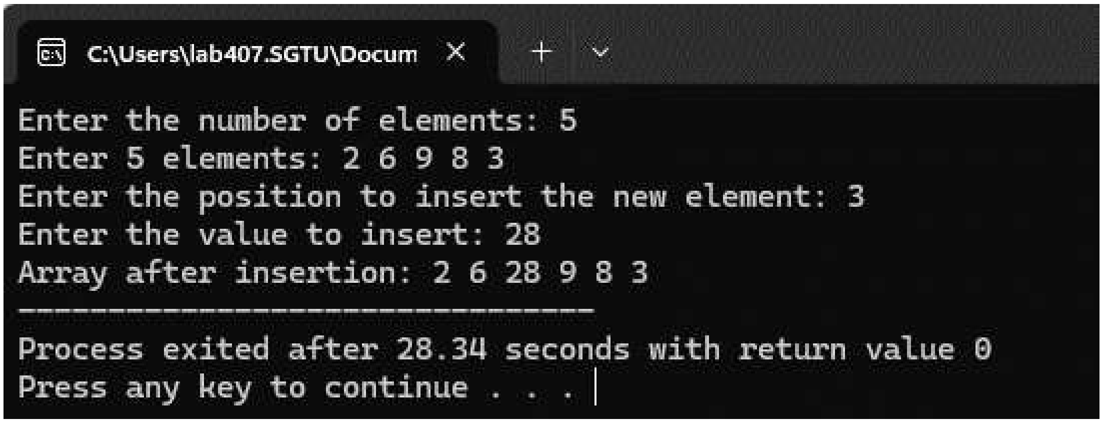


```
Enter the element to search: 50
Element 50 found at index 4.

--------------------------------
Process exited after 2.163 seconds with return value 0
Press any key to continue . . .
```

Program 29
Program to Perform Insertion in an Array

```
#include <stdio.h>
int main() {
   int arr[100], n, pos, value;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements: ", n);
   for (int i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }
   printf("Enter the position to insert the new element: ");
   scanf("%d", &pos);
   printf("Enter the value to insert: ");
   scanf("%d", &value);
   for (int i = n; i >= pos; i--) {
      arr[i] = arr[i - 1];
   }
   arr[pos - 1] = value;

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

OUTPUT

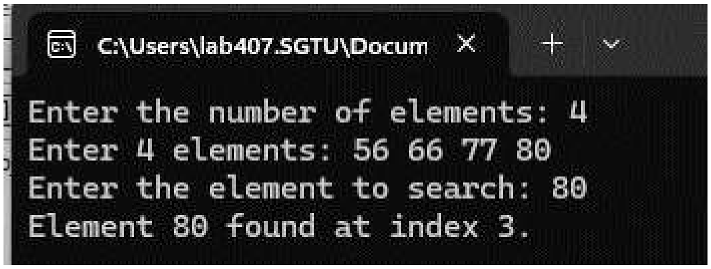


```
Enter the number of elements: 5
Enter 5 elements: 2 6 9 8 3
Enter the position to insert the new element: 3
Enter the value to insert: 28
Array after insertion: 2 6 28 9 8 3
--------------------------------
Process exited after 28.34 seconds with return value 0
Press any key to continue . . .
```

Program 30
Program to Implement Linear Search

```
#include <stdio.h>
int main() {
   int arr[100], n, key, found = 0;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   printf("Enter %d elements: ", n);
   for (int i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }
   printf("Enter the element to search: ");
   scanf("%d", &key);

   for (int i = 0; i < n; i++) {
      if (arr[i] == key) {
         printf("Element %d found at index %d.\n", key, i);
         found = 1;
         break;
      }
   }
   if (!found) {
      printf("Element %d not found in the array.\n", key);
   }

   return 0;
}
```

OUTPUT



```
Enter the number of elements: 4
Enter 4 elements: 56 66 77 80
Enter the element to search: 80
Element 80 found at index 3.
```