PRACTICAL – 9

PROGRAM -1

AIM- Write a program in C to create and store information in a text file.

CODE

|  |
| --- |
| /\* Write a program in C to create and store information in a text file. \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  int main()  {  FILE \*fp;  char str[100];  fp = fopen("file.txt", "w");  if (fp == NULL)  {  printf("Error opening file\n");  }  else  {  printf("Enter a string: ");  }  fgets(str, 100, stdin);  fputs(str, fp);  fclose(fp);  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -2

AIM- Write a program in C to write multiple lines to a text file.

CODE

|  |
| --- |
| /\* Write a program in C to write multiple lines to a text file. \*/  #include <stdio.h>  int main()  {  FILE \*fp;  fp = fopen("test.txt", "w");  fprintf(fp, "This is line 1\n");  fprintf(fp, "This is line 2\n");  fprintf(fp, "This is line 3\n");  fclose(fp);  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -3

AIM- Write a program in C to read the file and store the lines in an array.

CODE

|  |
| --- |
| /\* .Write a program in C to read the file and store the lines in an array. \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  int main(void)  {  FILE \*fp;  char \*line = NULL;  size\_t len = 0;  ssize\_t read;  int i = 0;  fp = fopen("file.txt", "r");  if (fp == NULL)  {  printf("Error opening file\n");  }  while ((read = getline(&line, &len, fp)) != -1)  {  printf("Line %d: %s", i, line);  }  fclose(fp);  if (line)  {  free(line);  }  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -4

AIM- Write a program in C to find the number of lines in a text file.

CODE

|  |
| --- |
| /\* Write a program in C to find the number of lines in a text file. \*/  #include <stdio.h>  int main()  {  printf("\n HARSH D \n");  FILE \*fp;  int count = 0;  char c;  fp = fopen("main.c", "r");  if (fp == NULL)  {  printf("Could not open file");  }  for (c = getc(fp); c != EOF; c = getc(fp))  {  if (c == '\n')  {  count = count + 1;  }  }  fclose(fp);  printf("The file has %d lines\n ", count);  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -5

AIM- Write a program in C to count the number of words and characters in a file .

CODE

|  |
| --- |
| /\* .Write a program in C to count the number of words and characters in a file \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  int main()  {  printf("\n HARSH D \n");  FILE \*fp;  char ch;  int count\_char = 0, count\_word = 0;  fp = fopen("file.txt", "r");  if (fp == NULL)  {  printf("Error opening file\n");  }  else  {  while ((ch = fgetc(fp)) != EOF)  {  count\_char++;  }  fclose(fp);  }  fp = fopen("file.txt", "r");  if (fp == NULL)  {  printf("Error opening file\n");  }  else  {  while ((ch = fgetc(fp)) != EOF)  {  if (ch == ' ' || ch == '\n')  {  count\_word++;  }  }  }  fclose(fp);  printf("Number of characters: %d\n", count\_char);  printf("Number of words: %d\n", count\_word);  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -6

AIM- Write a program in C to merge two files and write them to another file.

CODE

|  |
| --- |
| /\* Write a program in C to  merge two files and write them to another file. \*/  #include <stdio.h>  #include <stdlib.h>  int main()  {  printf("\n HARSH D \n");  FILE \*f1, \*f2, \*f3;  char c;  f1 = fopen("file1.txt", "r");  f2 = fopen("file2.txt", "r");  f3 = fopen("file3.txt", "w");  if (f1 == NULL || f2 == NULL || f3 == NULL)  {  puts("Could not open files");  }  else  {  while ((c = fgetc(f1)) != EOF)  {  fputc(c, f3);  }  while ((c = fgetc(f2)) != EOF)  {  fputc(c, f3);  }  }  fclose(f1);  fclose(f2);  fclose(f3);  return 0;  } |

OUTPUT

|  |
| --- |
|  |

PROGRAM -7

AIM- Write a program in C to use of miscellaneous Function.

CODE

|  |
| --- |
| /\* Write a program in C to use of miscellaneous Function. \*/  #include <stdio.h>  #include <math.h>  #include <stdlib.h>  #include <time.h>  // Function to calculate the factorial of a number  int factorial(int n) {  if (n == 0)  return 1;  else  return n \* factorial(n - 1);  }  // Function to generate a random number between a given range  int generateRandom(int min, int max) {  return min + rand() % (max - min + 1);  }  // Function to find the square root of a number  float squareRoot(float num) {  return sqrt(num);  }  int main()  {  printf("\n HARSH D \n");  int num, fact;  float number, sqrtResult;  int min, max;  // Use of factorial function  printf("Enter a number to calculate its factorial: ");  scanf("%d", &num);  fact = factorial(num);  printf("Factorial of %d is: %d\n", num, fact);  // Use of random number generation function  srand(time(0)); // Seed the random number generator  printf("\nEnter the range (min and max) to generate a random number: ");  scanf("%d %d", &min, &max);  printf("Random number between %d and %d is: %d\n", min, max, generateRandom(min, max));  // Use of square root function  printf("\nEnter a number to find its square root: ");  scanf("%f", &number);  sqrtResult = squareRoot(number);  printf("Square root of %.2f is: %.2f\n", number, sqrtResult);  return 0;  } |

OUTPUT

|  |
| --- |
|  |