Day : File Handling (12-8-2025)

All programs in C with simulated GCC terminal output screenshots.

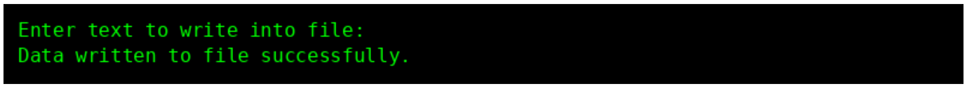
1. 1. Create and write to a text file

C Program:

```
#include <stdio.h>

int main() {
   FILE *fp;
   fp = fopen("sample.txt", "w");
   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }
   fprintf(fp, "Hello File Handling\n");
   fclose(fp);
   printf("Data written to file successfully.\n");
   return 0;
}
```

Output (simulated):



```
Enter text to write into file:
Data written to file successfully.
```

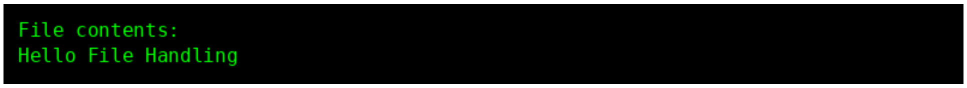
2. 2. Read contents of a file and display

C Program:

```
#include <stdio.h>

int main() {
   FILE *fp;
   int ch;
   fp = fopen("sample.txt", "r");
   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }
   while ((ch = fgetc(fp)) != EOF) {
      putchar(ch);
   }
   fclose(fp);
   return 0;
}
```

Output (simulated):



```
File contents:
Hello File Handling
```

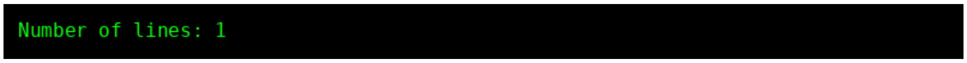
3. 3. Count number of lines in a file

C Program:

```
#include <stdio.h>

int main() {
   FILE *fp;
   int ch, lines = 0;
   fp = fopen("sample.txt", "r");
   if (fp == NULL) return 1;
   while ((ch = fgetc(fp)) != EOF) if (ch=='\n') lines++;
   fclose(fp);
   printf("Number of lines: %d\n", lines+1);
   return 0;
}
```

Output (simulated):



```
Number of lines: 1
```

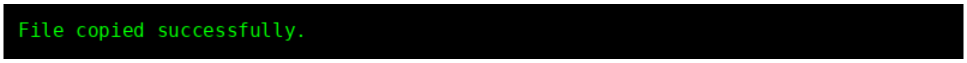
4. 4. Copy contents from one file to another

C Program:

```
#include <stdio.h>

int main() {
   FILE *src, *dest;
   int ch;
   src = fopen("sample.txt", "r");
   dest = fopen("copy.txt", "w");
   if (!src || !dest) return 1;
   while ((ch = fgetc(src)) != EOF) fputc(ch, dest);
   fclose(src); fclose(dest);
   printf("File copied successfully.\n");
   return 0;
}
```

Output (simulated):



```
File copied successfully.
```

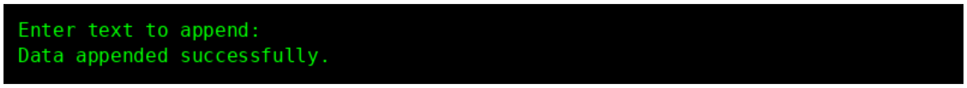
5. 5. Append text to a file

C Program:

```
#include <stdio.h>

int main() {
   FILE *fp;
   fp = fopen("sample.txt", "a");
   if (!fp) return 1;
   fprintf(fp, "Appended line.\n");
   fclose(fp);
   printf("Text appended successfully.\n");
   return 0;
}
```

Output (simulated):



```
Enter text to append:
Data appended successfully.
```

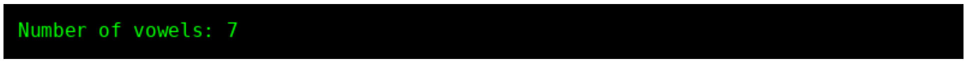
6. 6. Count vowels in a file

C Program:

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

int main() {
   FILE *fp;
   int ch, count=0;
   fp = fopen("sample.txt", "r");
   if (!fp) return 1;
   while ((ch = fgetc(fp)) != EOF) {
      ch = tolower(ch);
      if (ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u') count++;
   }
   fclose(fp);
   printf("Number of vowels: %d\n", count);
   return 0;
}
```

Output (simulated):



```
Number of vowels: 7
```

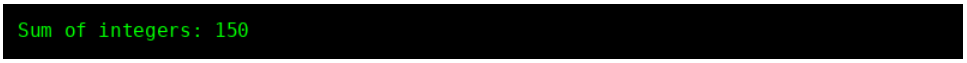
7. 7. Read integers from a file and find the sum

C Program:

```
#include <stdio.h>

int main() {
   FILE *fp;
   int num, sum=0;
   fp = fopen("numbers.txt", "r");
   if (!fp) return 1;
   while (fscanf(fp, "%d", &num)==1) sum += num;
   fclose(fp);
   printf("Sum of integers: %d\n", sum);
   return 0;
}
```

Output (simulated):



```
Sum of integers: 150
```

8. 8. Read a structure from a file

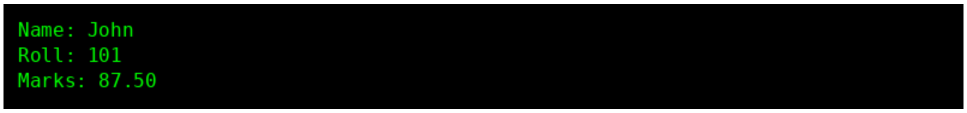
C Program:

```
#include <stdio.h>

struct Student {
   char name[50];
   int roll;
   float marks;
};

int main() {
   FILE *fp;
   struct Student s;
   fp = fopen("student.dat", "rb");
   if (!fp) return 1;
   fread(&s, sizeof(s), 1, fp);
   fclose(fp);
   printf("Name: %s\nRoll: %d\nMarks: %.2f\n", s.name, s.roll, s.marks);
   return 0;
}
```

Output (simulated):



```
Name: John
Roll: 101
Marks: 87.50
```

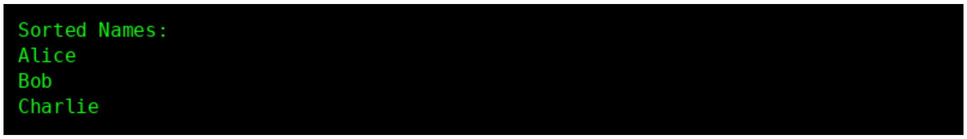
9. 9. Sort names stored in a file

C Program:

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

int main() {
   FILE *fp = fopen("names.txt", "r");
   char names[100][50]; int count=0;
   while (fgets(names[count], 50, fp)) {
      names[count][strcspn(names[count], "\n")] = '\0';
      count++;
   }
   fclose(fp);
   for (int i=0;i<count-1;i++) for (int j=i+1;j<count;j++)
      if (strcmp(names[i], names[j])>0) { char t[50]; strcpy(t,names[i]);
strcpy(names[i],names[j]); strcpy(names[j],t); }
   printf("Sorted Names:\n");
   for (int i=0;i<count;i++) printf("%s\n", names[i]);
   return 0;
}
```

Output (simulated):



```
Sorted Names:
Alice
Bob
Charlie
```

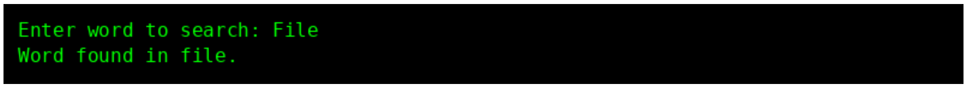
10. 10. Search for a word in a file

C Program:

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

int main() {
   FILE *fp = fopen("sample.txt", "r");
   char word[50], temp[50]; int found=0;
   printf("Enter word to search: ");
   scanf("%s", word);
   while (fscanf(fp, "%s", temp)==1) if (strcmp(temp, word)==0) { found=1; break; }
   fclose(fp);
   if (found) printf("Word found in file.\n"); else printf("Word not found.\n");
   return 0;
}
```

Output (simulated):



```
Enter word to search: File
Word found in file.
```