Program 1

```
#include <stdio.h>

#define MAX 20


int frames[MAX], ref[MAX], mem[MAX][MAX], time[MAX], faults = 0, m, n, counter = 0;


void accept() {
   printf("Enter number of frames: ");
   scanf("%d", &n);
   printf("Enter number of references: ");
   scanf("%d", &m);
   printf("Enter reference string:\n");
   for (int i = 0; i < m; i++) {
      printf("[%d] = ", i);
      scanf("%d", &ref[i]);
   }
}


int search(int pno) {
   for (int i = 0; i < n; i++) {
      if (frames[i] == pno) return i;
   }
   return -1;
}


int get_lru() {
   int min = 9999, min_i = 0;
   for (int i = 0; i < n; i++) {
      if (time[i] < min) {
         min = time[i];
         min_i = i;
      }
   }
   return min_i;
}


void lru() {
   for (int i = 0; i < m; i++) {
```

```
        int k = search(ref[i]);
        if (k == -1) {
            if (counter < n) {
                frames[counter] = ref[i];
                time[counter] = i;
                counter++;
            } else {
                int pos = get_lru();
                frames[pos] = ref[i];
                time[pos] = i;
            }
            faults++;
        } else {
            time[k] = i;
        }
        for (int j = 0; j < n; j++) {
            mem[j][i] = frames[j];
        }
    }
}

void disp() {
    printf("\nReference String:\n");
    for (int i = 0; i < m; i++) {
        printf("%3d", ref[i]);
    }
    printf("\n\nFrame Allocation:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (mem[i][j]) {
                printf("%3d", mem[i][j]);
            } else {
                printf("   ");
            }
        }
        printf("\n");
    }
    printf("\nTotal Page Faults: %d\n", faults);
}
```

```
int main() {
    accept();
    lru();
    disp();
    return 0;
}
```

Program 2

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

void count_lines(char *filename) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        printf("File %s not found.\n", filename);
        return;
    }

    int lines = 0;
    char ch;
    while ((ch = fgetc(file)) != EOF) {
        if (ch == '\n') lines++;
    }

    printf("Total lines: %d\n", lines);
    fclose(file);
}

void count_words(char *filename) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        printf("File %s not found.\n", filename);
        return;
    }

    int words = 0;
    char word[100];
```

```
    while (fscanf(file, "%s", word) != EOF) {
        words++;
    }

    printf("Total words: %d\n", words);
    fclose(file);
}

void count_chars(char *filename) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        printf("File %s not found.\n", filename);
        return;
    }

    int chars = 0;
    char ch;
    while ((ch = fgetc(file)) != EOF) {
        chars++;
    }

    printf("Total characters: %d\n", chars);
    fclose(file);
}

int main() {
    char command[100], *args[10];
    while (1) {
        printf("\nmyshell$ ");
        fgets(command, 100, stdin);
        command[strlen(command) - 1] = '\0';  // Remove newline

        char *token = strtok(command, " ");
        int i = 0;
        while (token != NULL) {
            args[i++] = token;
            token = strtok(NULL, " ");
        }
        args[i] = NULL;
```

```
        if (strcmp(args[0], "count") == 0) {
            if (strcmp(args[1], "l") == 0) {
                count_lines(args[2]);
            } else if (strcmp(args[1], "w") == 0) {
                count_words(args[2]);
            } else if (strcmp(args[1], "c") == 0) {
                count_chars(args[2]);
            }
        } else if (strcmp(args[0], "exit") == 0) {
            exit(0);
        } else {
            int pid = fork();
            if (pid == 0) {
                execvp(args[0], args);
                exit(0);
            } else {
                wait(NULL);
            }
        }
    }
    return 0;
}
```