The purpose of this program is to write a shell that allows the user to give commands to the system and to implement the built-in commands correctly.

The first part of the program reads the command line entered by the user and splits it into words using the function splitCommandLine. Splitting a line into words is achieved through the following steps. Firstly, the skipChar function is used to search and return the address of the first non-space character from the beginning of the given string. The returned pointer is stored in an array of char pointers. Next, we change the first space character after that non-space character to a null character to terminate each word, and then change the address of the string to the address of the next character after the null character and repeat the steps above until the end of the line.

The second part uses the first word of the command line to check if the given command is included in a set of built-in commands. This is done by comparing the user input to an array of structures containing names commands and a pointer to the respective function. If the command is found, it will be executed by calling the associated command handling function. The command dispatch table is used to associate a command name with a corresponding function. The function doCommand compares the name of the command inside the dispatch table to the first word entered by the user and calls the associated command handling function if they are the same. The command handling functions are built to implement the commands accurately. The pwd command handling function prints the current working directory. The cd command function changes the current directory to the home directory if there is no argument, otherwise it changes it to the argument if it is a directory that exists. If the second argument is invalid it prints an appropriate error. The ls command function, if there is no argument, will only list the files that do not start with the dot character, if the argument is “-a”, it will list all the files, and if the second argument is invalid it will print an appropriate error.

Some advanced features of C have been used in this program. The first is the library function char* strchr (const char* str, int c). This function searches for the first occurrence of the character c in the string pointed by str and returns a pointer to that character or returns NULL if the character is not found. This function is used to find the first space in order to skip over all the non-space characters. The second is structure, which is a user-defined datatype in C languages. It enables us to combine a char pointer to the name of the command with a pointer to the function that implements the command. The third is the getcwd function that returns the address of the current working directory name in the function that implements pwd command. The fourth is the chdir function, which is used to change the current directory to the new directory that the parameter passed to the chdir function points to and returns 0 if the change is completed. Otherwise, a non-zero value will be returned.