Directory structure maintenance using shell script in the operating system

Some of the directory functions are:

-----> Navigation

-----> Absolute/ Relative Pathnames

-----> Listing Directories

-----> Creating Directories

-----> Modifying Directories

 Execute your files as mentioned below in the output screenshot. The following shell script implements these functions of directory , using commands available in Linux.

CODE:

```
echo "    "
echo "----Implementing Directory Management----"
echo "    "
ch=0
while [ $ch -lt 6 ]
do
    echo "Press the following to :"
    echo "1) Create a new directory."
    echo "2) Modify a directory."
```

```
echo "3) Navigate into directory."
echo "4) Listing directories."
echo "5) Exit."
read ch

case $ch in
1) echo " "
   echo "---Creation of Directory---"
   echo "  "
   echo "Enter the name of the directory:"
   read name
   mkdir $name
;;
2) echo " "
   echo "---Modification of Directory---"
   echo "   "
   echo "Enter the directory to be modified:"
   read orgdir
   echo "Press the following to :"
   echo "   "
   echo "1) Rename directory."
   echo "2) Copy directory to another."
   echo "3) Move directory."
   echo "4) Delete directory."
   echo "5) Exit from Modify Mode."
   read modch

   case $modch in
   1) echo "  "
```

```
      echo "---Rename a directory---"

      echo "  "

      echo "Enter new name for the directory:"

      read newname

      mv $orgdir $newname

;;

2) echo "  "

      echo "---Copying a directory to another---"

      echo "  "

      echo "Enter target directory:"

      read target

      mkdir $target

      cp $orgdir $target

;;

3) echo "  "

      echo "---Moving a directory---"

      echo "  "

      echo "Enter target directory:"

      read target

      mkdir $target

      mv $orgdir $target

;;

4) echo "  "

      echo "---Deleting a directory---"

      echo "  "

      rmdir $orgdir

;;

5) echo "  "

      echo "---Exiting from modify mode---"
```

```
        echo " "

        exit

    ;;

    esac

 ;;

3)

  echo "---Navigation of Directory---"

  echo " "

  echo "Enter your choice for method of navigation :"

  echo "1) Go to Parent Directory. "

  echo "2) Navigate to specific directory."

  echo "3) Exit from Navigate Mode."

  read navch


  case $navch in

    1) echo " "

        echo "---Parent Directory---"

        echo " "

        cd ..

        pwd

    ;;

    2) echo " "

        echo "---Navigation to Specific Directory---"

        echo " "

        echo "Enter the target Path:"

        read path

        cd $path

        pwd

    ;;
```

```
        3) echo " "

           echo "---Exiting from Navigate Mode---"

           echo " "

           exit

       ;;

       esac

    ;;

4)

   echo "---   Listing of Directories---"

   echo " "

   echo "Enter your choice for method of listing :"

   echo "1) List of directories. "

   echo "2) List of directories and their details."

   echo "3) Exit from List Mode."

   read lisch

   case $lisch in

     1) echo " "

         echo "---List of directories---"

         echo " "

         ls

     ;;

     2) echo " "

         echo "---Detailed List of directories---"

         echo " "

         ls -l

     ;;

     3) echo " "

         echo "---Exiting from List Mode---"
```

```
            echo " "
            exit
       ;;
       esac
    ;;
  5)echo "  "
    echo "---Exiting---"
    echo "  "
exit
    ;;
    esac
```

IMPLEMENTING DIRECTORIES IN C

```c
#include <stdio.h>
#include <dirent.h>
int main(void)
{
  DIR *directory;
  struct dirent *entry;
  directory = opendir(".");
  if (directory == NULL)
  {
    printf("Error opening directory.\n");
    return 1;
  }
  while ((entry = readdir(directory)) != NULL)
  {
    if (entry»d_name == directory);
    {
      printf("File: %s\n", entry»d_name);
    }
      if (entry»d_name == directory)
```

```c
    {
        printf(" dir: %s\n", entry»d_name);
    }
  }
  if (closedir(directory) == -1)
  {
    printf("Error closing directory.\n");
    return 1;
  }


  return 0;
}
```