National University Ho Chi Minh City

HO CHI MINH UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE REPORT
LAB 1

Instructor: Prof. Phạm Hoàng Anh
Class: CC04

Name: Nguyễn Quang Thiện

Student ID: 2152994

Completion Day: February 22th, 2023

Exercise 3.6

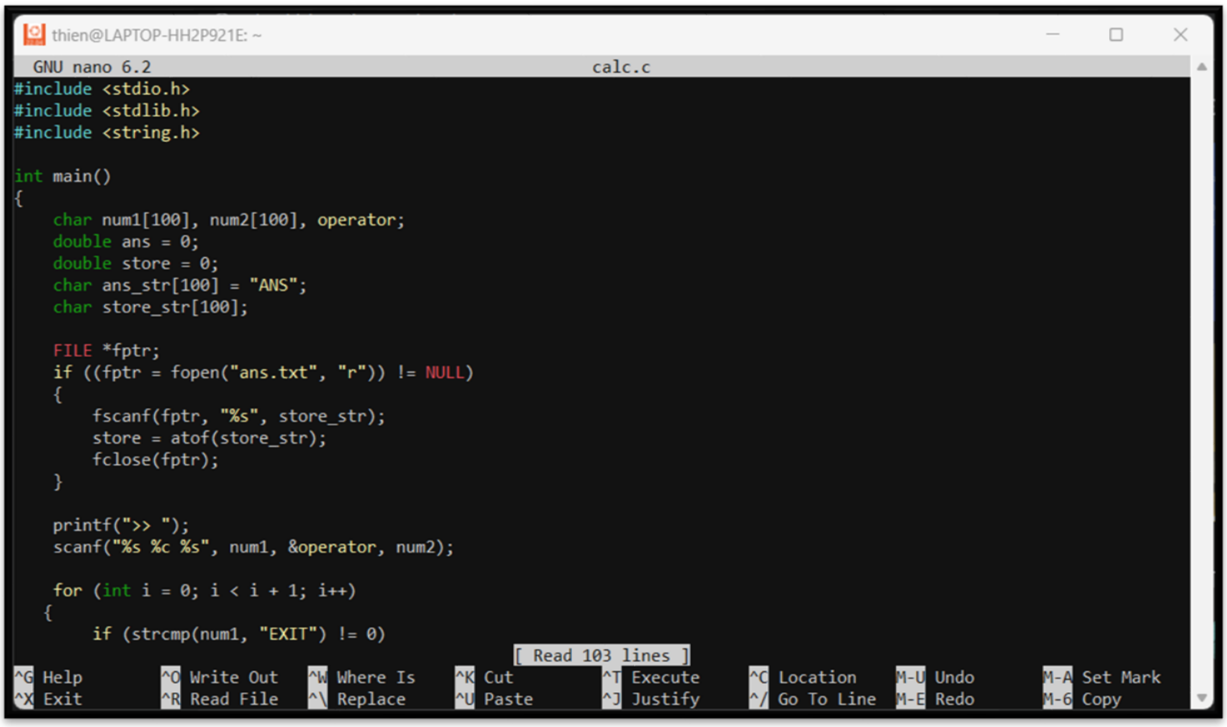
Explanation Code

```
#! /bin/bash

# Creating hist array to store history
declare -a hist

# Input numbers and operator to implement calculation
read -p ">> " num1 operator num2

# Allowing access like an admin
if [ -f ans.txt ]
then
	 # If ans.txt has a char, assigning it to store variable
	read store < ans.txt
else
	 # Else assigning 0 to the store variable
	store=0
fi

# If input num1 is exit, end the program. Else, do the following lines
while [ "$num1" != "EXIT" ]
do

# If the input is HIST then print the result with the HIST array with n size.
if [ "$num1" = "HIST" ]
then
	size=${#hist[@]}
	for ((i = 0; i <= $size; i++))
	do
		echo ${hist[$i]}
	done

# If the input of num1, num2 or operator is wrong. Printing “syntax error”
elif [[ ! "$num1" =~ ^-?[0-9]*(\.[0-9]+)?$|ANS ]] || [[ ! "$num2" =~ ^-?[0-9]*(\.[0-9]+)?$|ANS ]] ||
[[ ! "$operator" =~ [+x/%-] ]]
then
	echo -e "SYNTAX ERROR"

# Checking if the num2 is 0 when implementing the division. Printing “math error”
elif ([ "$operator" = "/" ] || [ "$operator" = "%" ]) && [ $num2 = 0 ]
then
	echo -e "MATH ERROR"

# Checking if the num1 or num2 is euqal to ANS in order to store the results in the store variable
and ans_flag
```

```
else
        ans_flag=-1
        if [ "$num1" = "ANS" ] && [ "$num2" = "ANS" ]
        then
                num1=$store
                num2=$store
                ans_flag=0
        fi

        if [ "$num1" = "ANS" ]
        then
                num1=$store
                ans_flag=1
        fi

        if [ "$num2" = "ANS" ]
        then
                num2=$store
                ans_flag=2
        fi

# Creating the calculation code with each case
        case $operator in
        "+")res=`echo $num1 + $num2 | bc`
        ;;
        "-")res=`echo $num1 - $num2 | bc`
        ;;
        "x")res=`echo $num1 \* $num2 | bc`
        ;;
        "/")res=`echo "scale=2; $num1 / $num2" | bc`
        ;;
        "%")res=`echo "scale=0; $num1 / $num2" | bc`
        ;;
        esac

# If the result is not an integer, then convert it to float number
        if [[ ! "$res" =~ ^-?[0-9]+$ ]]
        then
                res=`printf "%.2f" $res`
        fi

# The value of ans_flag is the condition that deciding num1 or num2 will be assigned to ANS for
hist to print.
        case $ans_flag in
        0)num1=ANS
        num2=ANS
        ;;
        1)num1=ANS
        ;;
        2)num2=ANS
        ;;
        esac

# Storing the results to the store variable
        store=$res

# Transferring the store variable to the ans.txt
        echo $store > ans.txt

# If the calculation equals to the res so that assigning to hist variable and push it to the array
        hist+=("$num1 $operator $num2 = $res")

# If the size array of hist > 5, pop out the oldest and push the newest into the array
        if [ ${#hist[@]} -gt 5 ]; then
                hist=("${hist[@]:1}")
        fi

# Printing the result
        echo -e "$res"
fi

# Reading, refreshing, and continuously implementing next calculations
read -n 1
clear
read -p ">> " num1 operator num2

done
```

Exercise 5.3

<u>calc.sh convert to calc.c</u>



```
thien@LAPTOP-HH2P921E: ~
 GNU nano 6.2                              calc.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
    char num1[100], num2[100], operator;
    double ans = 0;
    double store = 0;
    char ans_str[100] = "ANS";
    char store_str[100];

    FILE *fptr;
    if ((fptr = fopen("ans.txt", "r")) != NULL)
    {
        fscanf(fptr, "%s", store_str);
        store = atof(store_str);
        fclose(fptr);
    }

    printf(">> ");
    scanf("%s %c %s", num1, &operator, num2);

    for (int i = 0; i < i + 1; i++)
   {
        if (strcmp(num1, "EXIT") != 0)
                              [ Read 103 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy
```

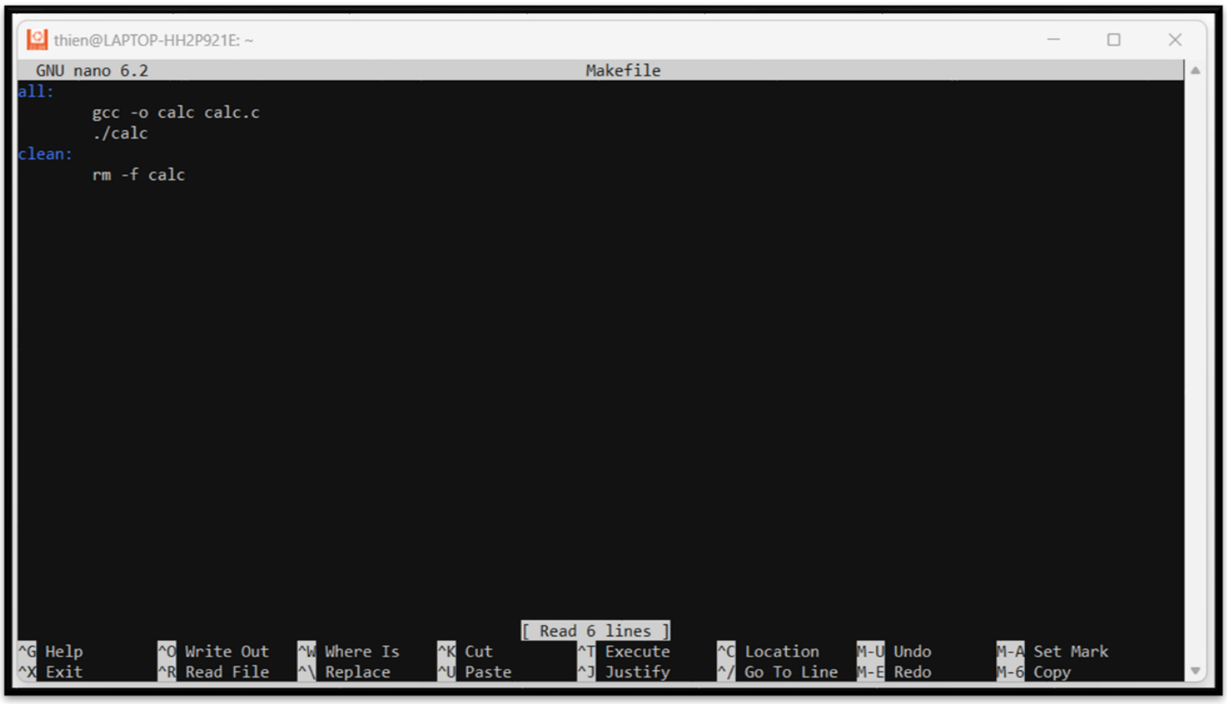
Code C and explanation

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

int main()
{
   # Creating input variables
   char num1[100], num2[100], operator;
   double ans = 0;
   double store = 0;
   char ans_str[100] = "ANS";
   char store_str[100];

   # Creating a pointer to ans.txt and reading ans.txt file in order to assign the reasonable value to
store variable and file must be exist
   FILE *fptr;
   if ((fptr = fopen("ans.txt", "r")) != NULL)
   {
      fscanf(fptr, "%s", store_str);
      store = atof(store_str);
      fclose(fptr);
   }

   # Entering the variables
   printf(">> ");
   scanf("%s %c %s", num1, &operator, num2);
```

```
# Creating infinite loop and breaking when requiring “EXIT”
for (int i = 0; i < i + 1; i++)
{
   # num1 is not EXIT then do the following lines
   if (strcmp(num1, "EXIT") != 0)
  {

   # num1 or num2 is equal to ANS so that assign ANS to the variable equivalent ANS
   if (strcmp(num1, "ANS") == 0 && strcmp(num2, "ANS") == 0)
   {
      sprintf(num1, "%f", ans);
      sprintf(num2, "%f", ans);
   }

   else if (strcmp(num1, "ANS") == 0)
   {
      sprintf(num1, "%f", ans);
   }
   else if (strcmp(num2, "ANS") == 0)
   {
      sprintf(num2, "%f", ans);
   }

   # Checking if num1, num2 or operator is wrong. Printing “syntax error”
   if ((atof(num1) == 0 && num1[0] != '0') || (atof(num2) == 0 && num2[0] != '0'))
   {
      printf("SYNTAX ERROR\n");
   }
   else if (operator != '+' && operator != '-' && operator != 'x' && operator != '/' && operator != '%')
   {
      printf("SYNTAX ERROR\n");
   }

   # Checking the condition of the division
   else if ((operator == '/' || operator == '%') && atof(num2) == 0)
   {
      printf("MATH ERROR\n");
   }

   # Creating the calculation code with each case
   else
   {
      double num1_val = atof(num1);
      double num2_val = atof(num2);
      double res;
```

```
        switch (operator)
        {
            case '+':
                res = num1_val + num2_val;
                break;
            case '-':
                res = num1_val - num2_val;
                break;
            case 'x':
                res = num1_val * num2_val;
                break;
            case '/':
                res = num1_val / num2_val;
                break;
            case '%':
                res = (int)num1_val % (int)num2_val;
                break;
        }

        # Printing the result
        printf("%.2f\n", res);
        ans = res;

        # Storing to the ans.txt file and removing the previous value then closing the file
        fptr = fopen("ans.txt", "w");
        fprintf(fptr, "%.2f", ans);
        fclose(fptr);
    }

        # Continuing input of new variables and also pause the infinte loop
        printf(">> ");
        scanf("%s %c %s", num1, &operator, num2);

  }

  # Else stop the program if num1 is EXIT
  else
  {
      break;
  }
 }

 return 0;
}
```

Makefile



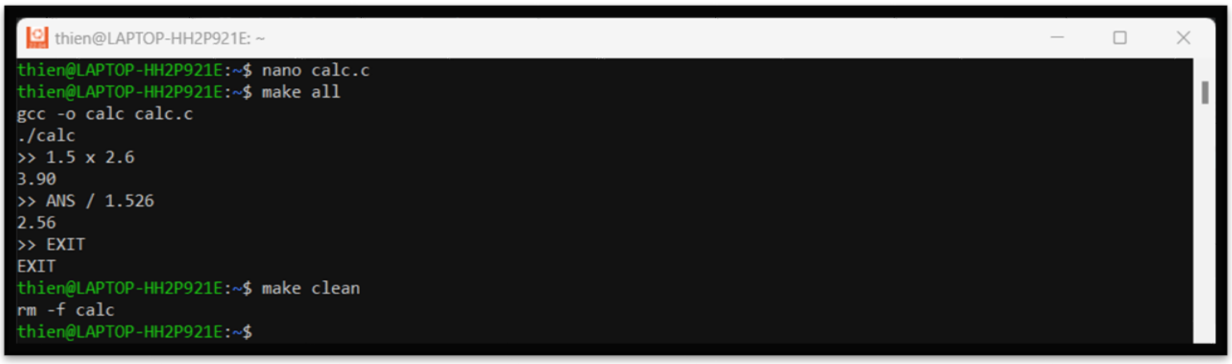
```
thien@LAPTOP-HH2P921E: ~
  GNU nano 6.2                                   Makefile
all:
        gcc -o calc calc.c
        ./calc
clean:
        rm -f calc

                                          [ Read 6 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy
```

```
all:
        gcc -o calc calc.c
        ./calc
clean:
        rm -f calc
```

Explanation:

Make all: To create an execution file, define all targets and run the C program (calc.c).
Make clean: To delete executable files alongside the object files from a directory.

Running Command Lines



```
thien@LAPTOP-HH2P921E: ~
thien@LAPTOP-HH2P921E:~$ nano calc.c
thien@LAPTOP-HH2P921E:~$ make all
gcc -o calc calc.c
./calc
>> 1.5 x 2.6
3.90
>> ANS / 1.526
2.56
>> EXIT
EXIT
thien@LAPTOP-HH2P921E:~$ make clean
rm -f calc
thien@LAPTOP-HH2P921E:~$
```