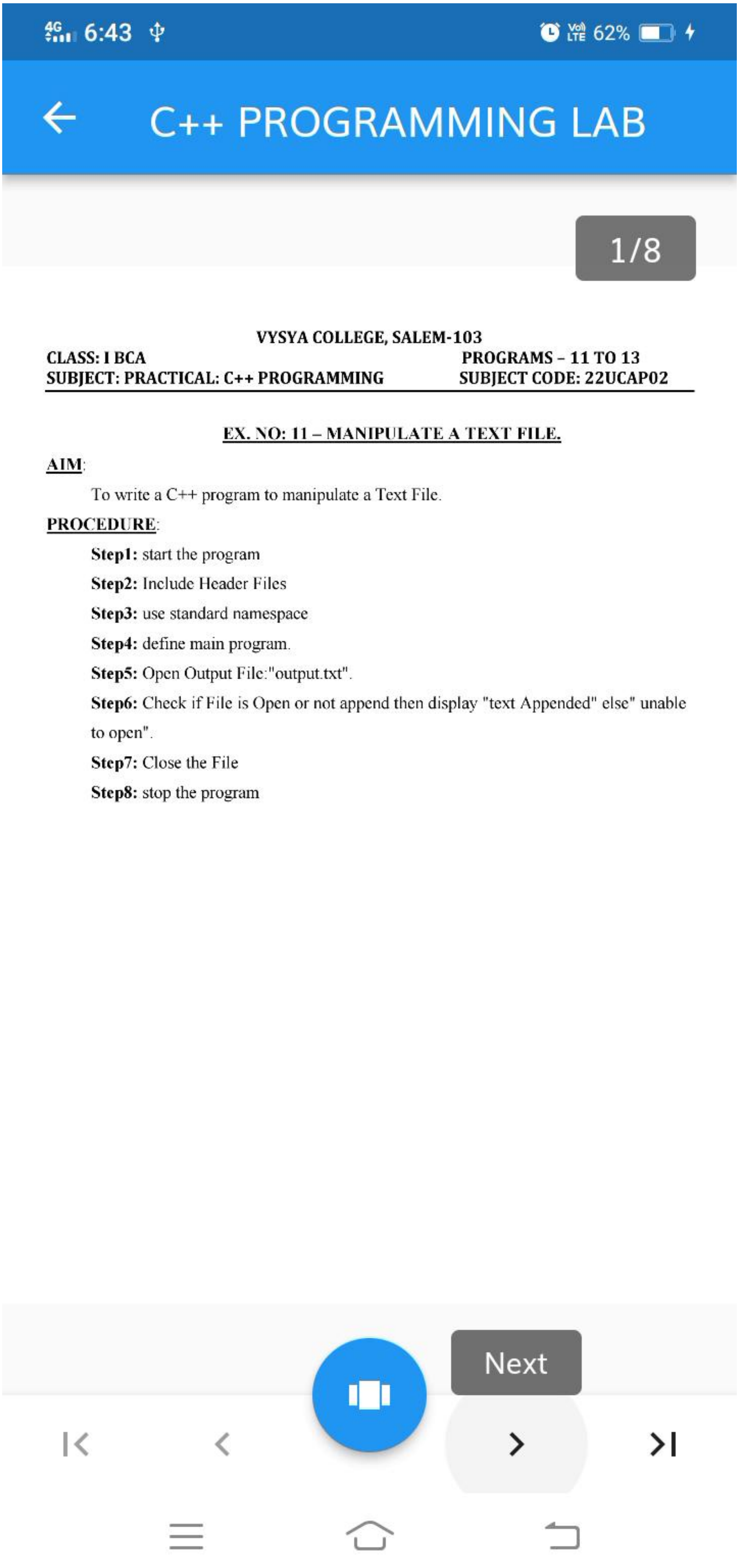


VYSYA COLLEGE, SALEM-103

CLASS: I BCA PROGRAMS – 11 TO 13

SUBJECT: PRACTICAL: C++ PROGRAMMING SUBJECT CODE: 22UCAP02

EX. NO: 11 – MANIPULATE A TEXT FILE.

AIM:

To write a C++ program to manipulate a Text File.

PROCEDURE:

Step1: start the program

Step2: Include Header Files

Step3: use standard namespace

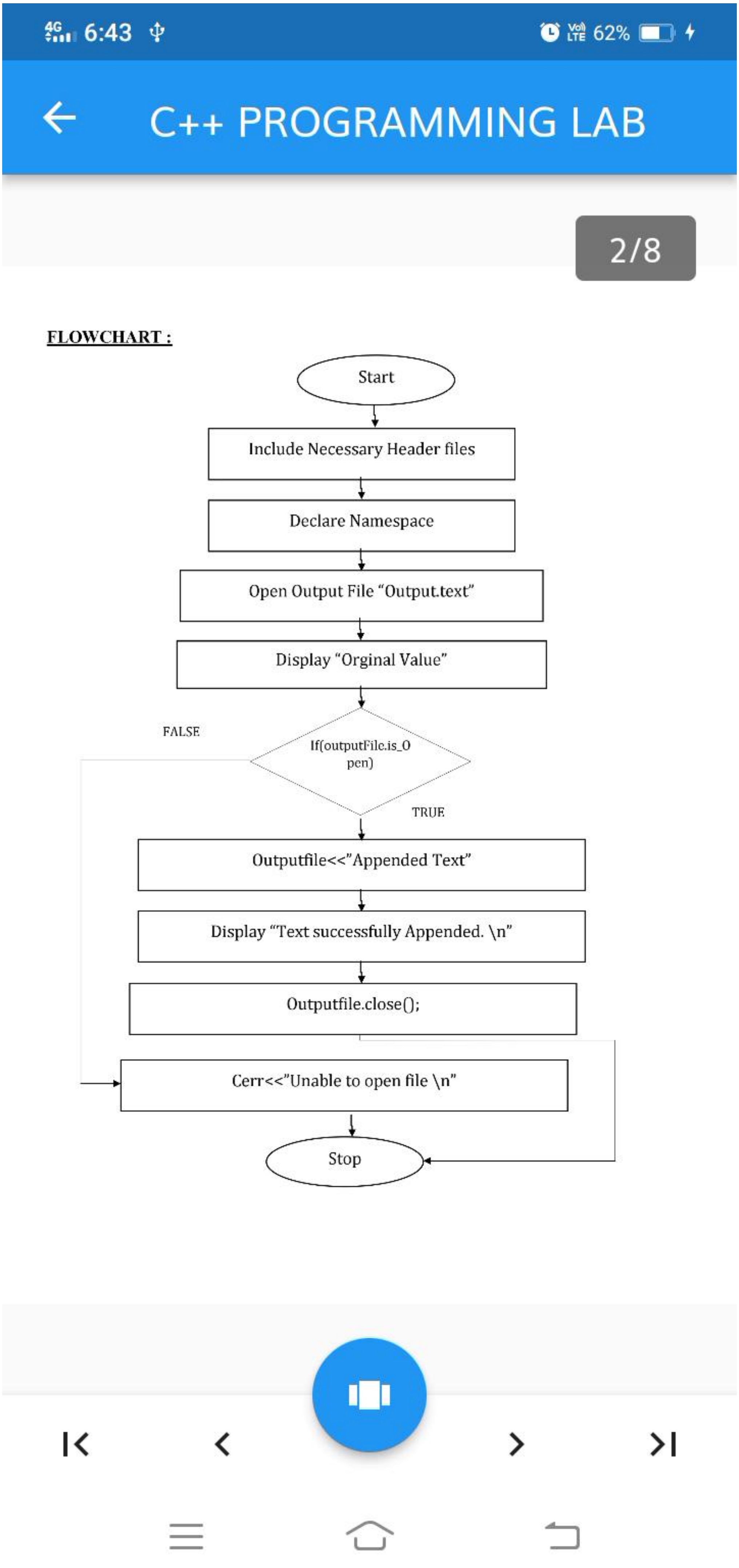
Step4: define main program.

Step5: Open Output File:"output.txt".

Step6: Check if File is Open or not append then display "text Appended" else" unable to open".

Step7: Close the File

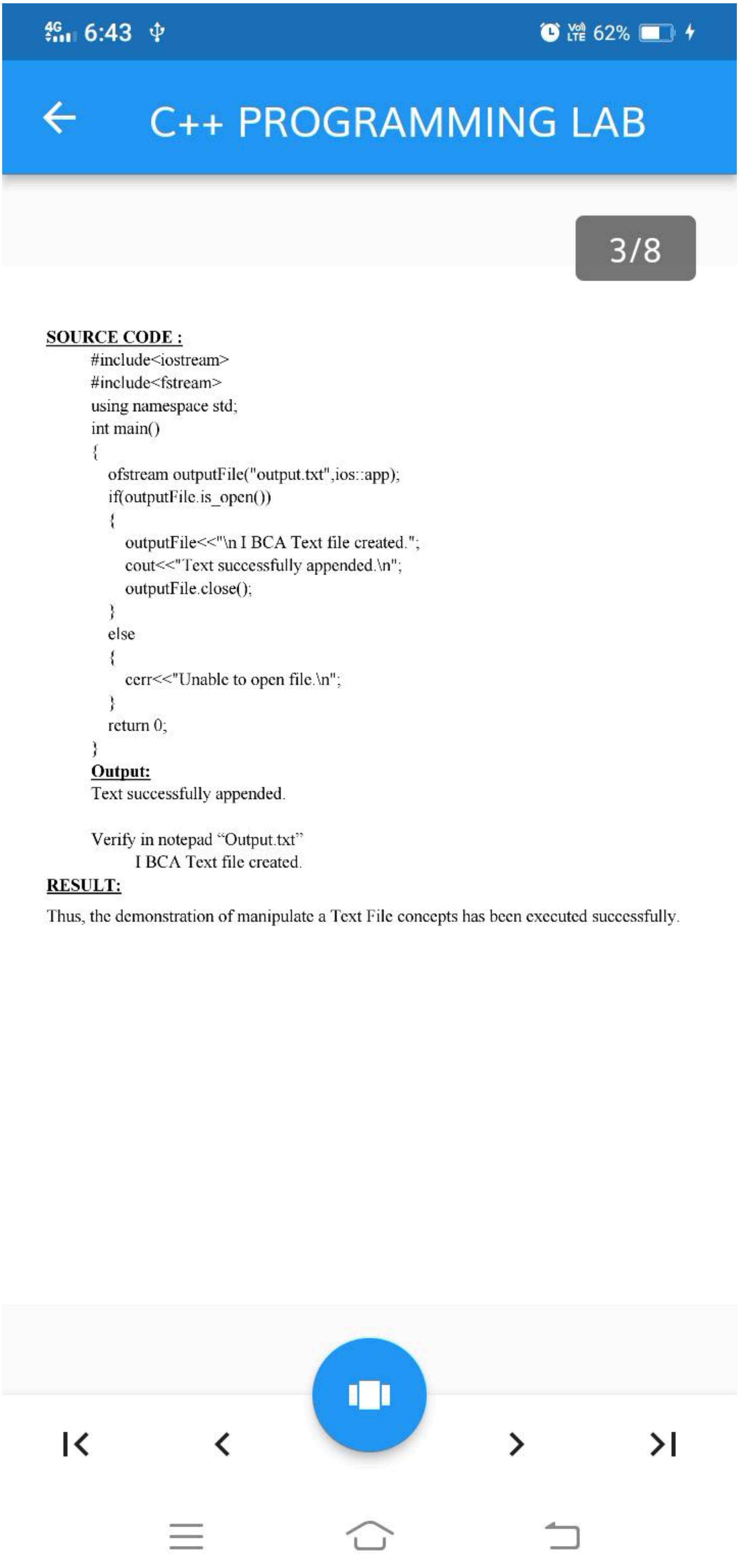
Step8: stop the program



6:43
62%
C++ PROGRAMMING LAB
2/8

FLOWCHART :

Start
Include Necessary Header files
Declare Namespace
Open Output File “Output.text”
Display “Orginal Value”
FALSE
If(outputFile.is_O pen)
TRUE
Outputfile<<”Appended Text”
Display “Text successfully Appended. \n”
Outputfile.close();
Cerr<<”Unable to open file \n”
Stop



SOURCE CODE :

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
   ofstream outputFile("output.txt",ios::app);
   if(outputFile.is_open())
   {
      outputFile<<"\n I BCA Text file created.";
      cout<<"Text successfully appended.\n";
      outputFile.close();
   }
   else
   {
      cerr<<"Unable to open file.\n";
   }
   return 0;
}
```

Output:

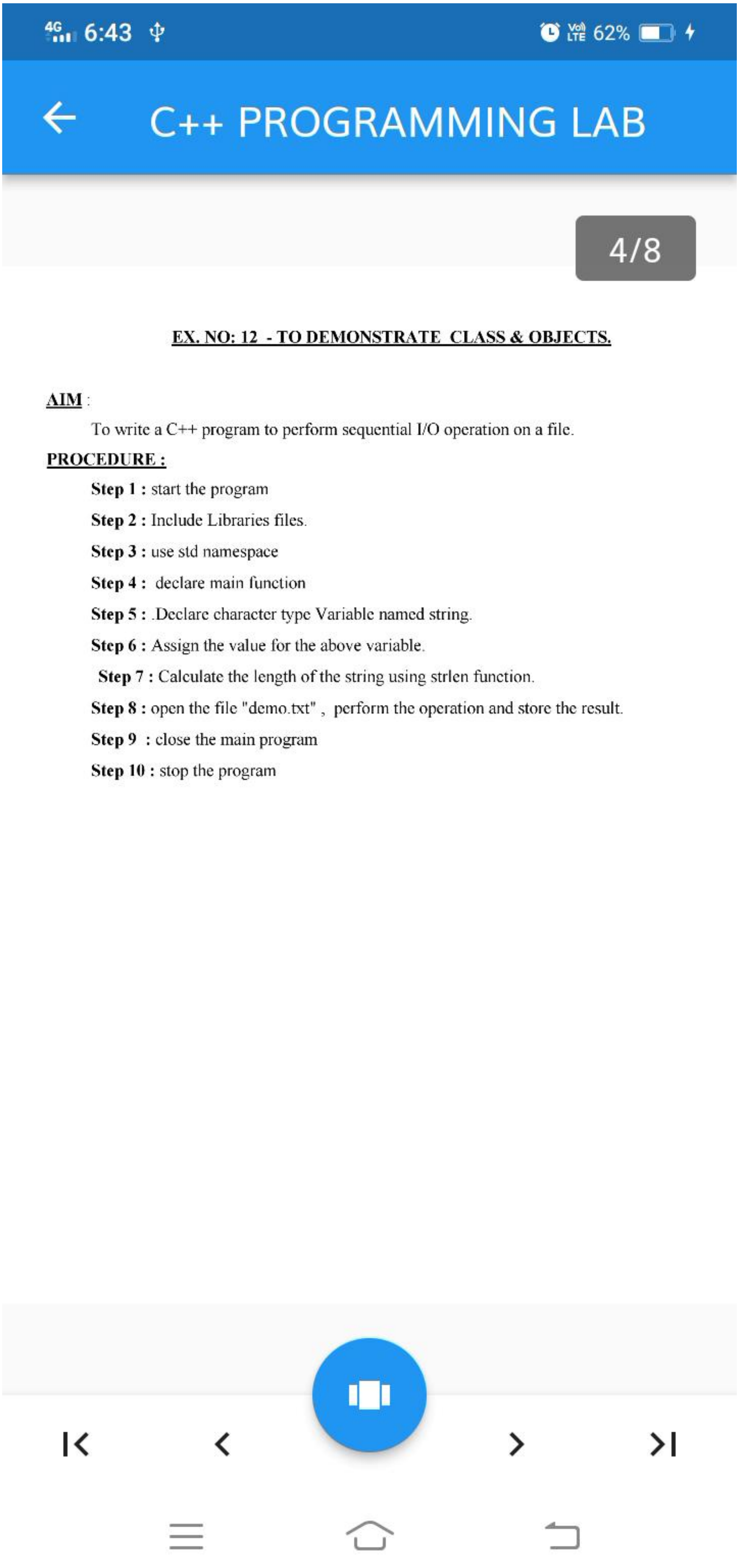
```
Text successfully appended.
```

Verify in notepad “Output.txt”

```
I BCA Text file created.
```

RESULT:

Thus, the demonstration of manipulate a Text File concepts has been executed successfully.



EX. NO: 12 - TO DEMONSTRATE CLASS & OBJECTS.

AIM :

To write a C++ program to perform sequential I/O operation on a file.

PROCEDURE :

Step 1 : start the program

Step 2 : Include Libraries files.

Step 3 : use std namespace

Step 4 : declare main function

Step 5 : .Declare character type Variable named string.

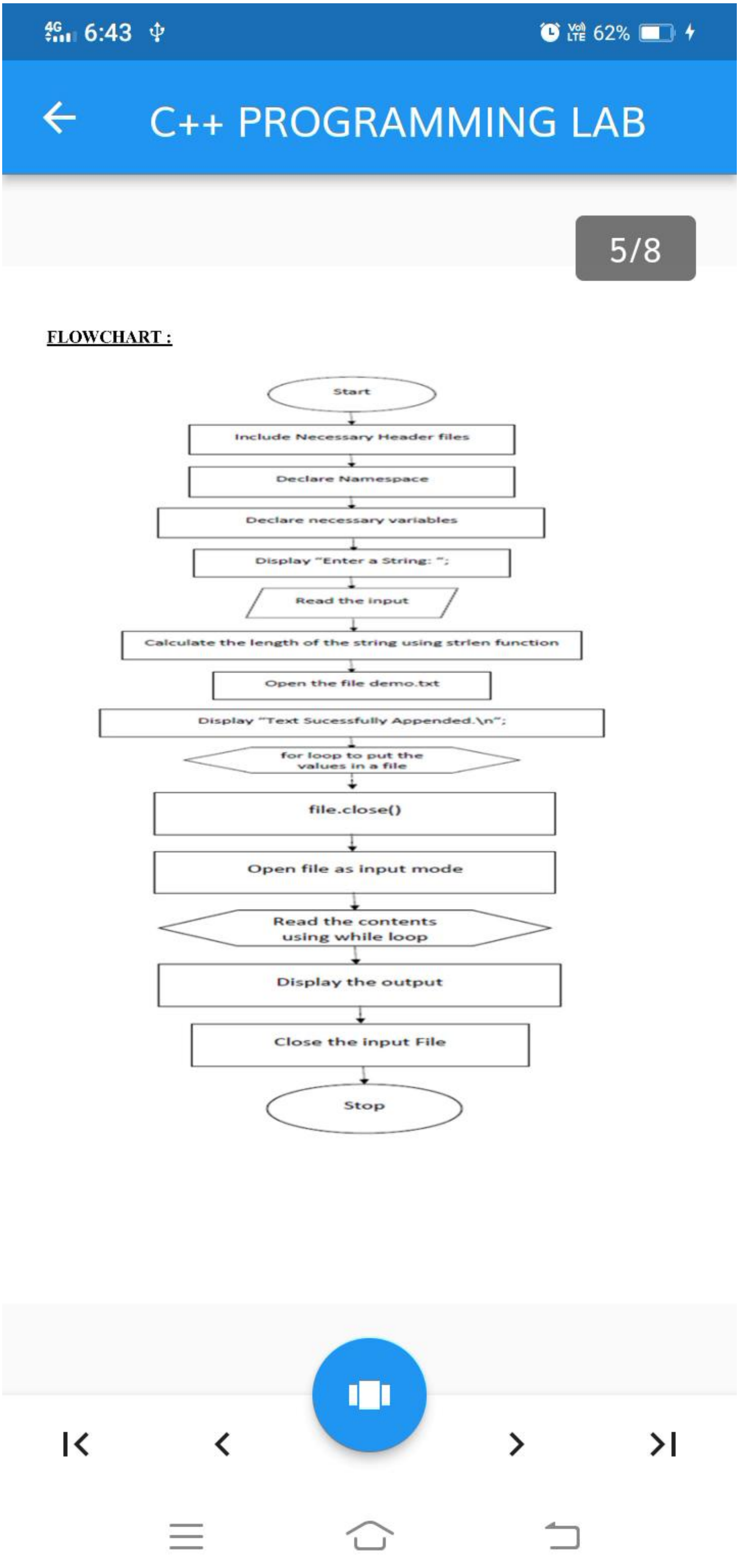
Step 6 : Assign the value for the above variable.

Step 7 : Calculate the length of the string using strlen function.

Step 8 : open the file "demo.txt" , perform the operation and store the result.

Step 9 : close the main program

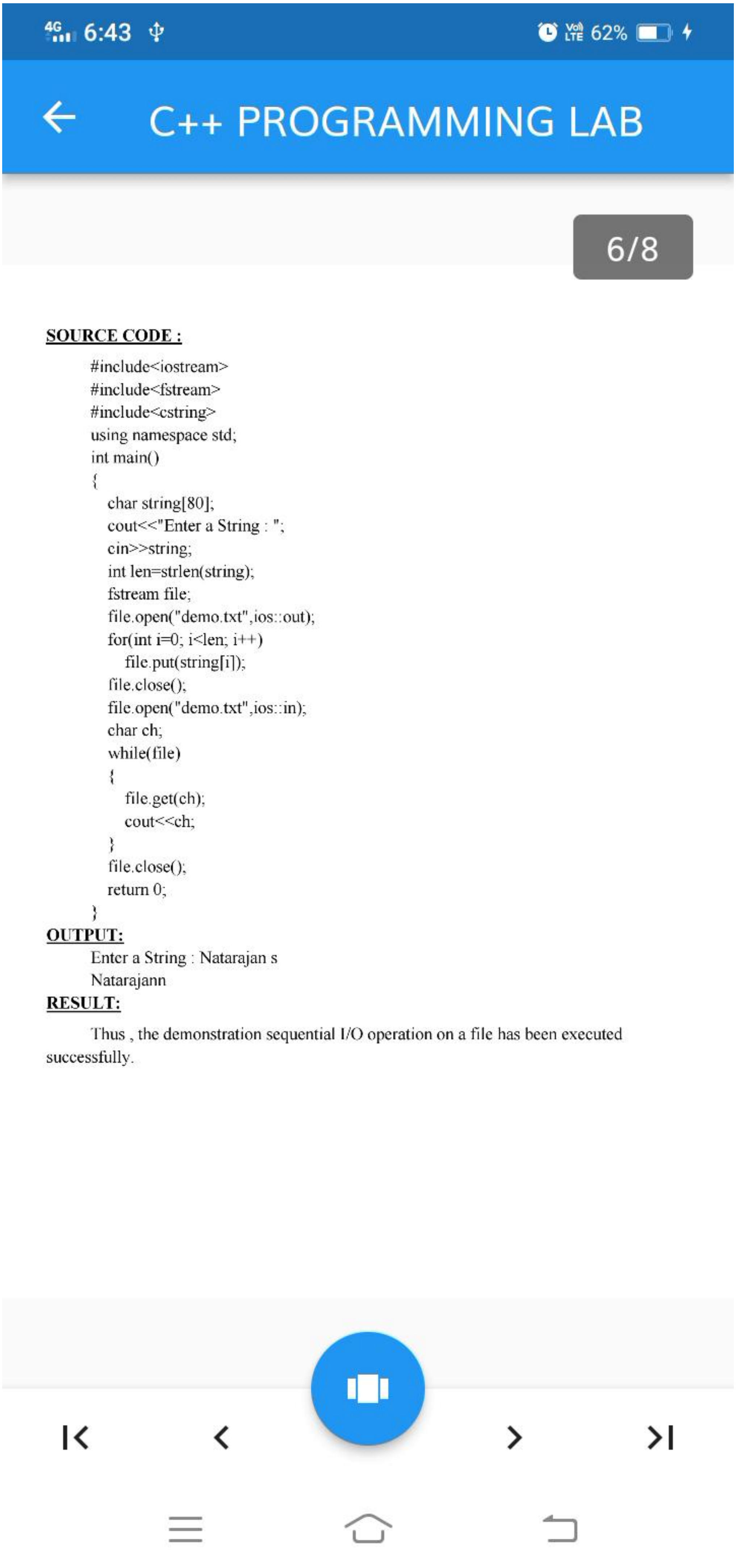
Step 10 : stop the program



C++ PROGRAMMING LAB

FLOWCHART :

Start
Include Necessary Header files
Declare Namespace
Declare necessary variables
Display "Enter a String: ";
Read the input
Calculate the length of the string using strlen function
Open the file demo.txt
Display "Text Sucessfully Appended.\n";
for loop to put the values in a file
file.close()
Open file as input mode
Read the contents using while loop
Display the output
Close the input File
Stop



SOURCE CODE :

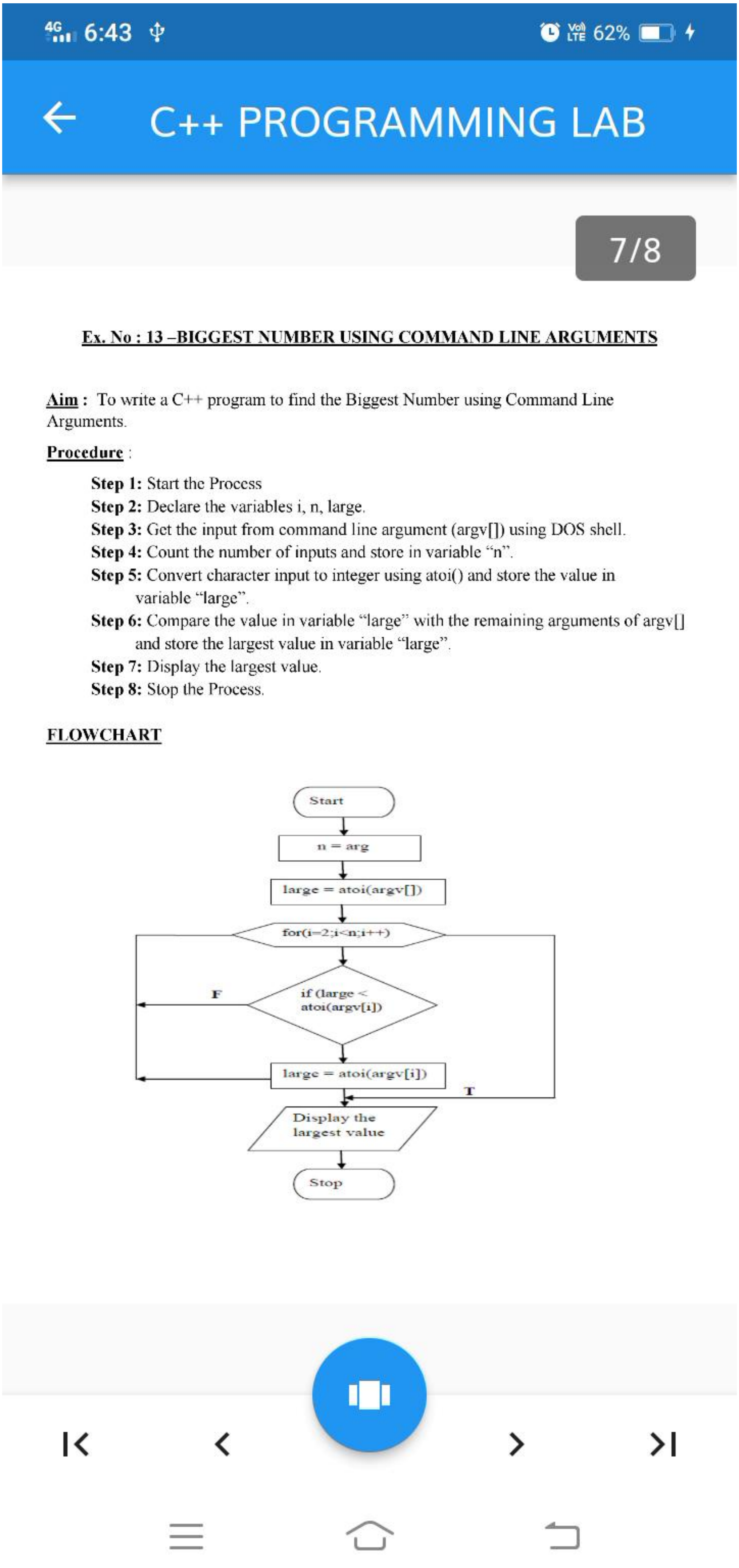
```
#include<iostream>
#include<fstream>
#include<cstring>
using namespace std;
int main()
{
   char string[80];
   cout<<"Enter a String : ";
   cin>>string;
   int len=strlen(string);
   fstream file;
   file.open("demo.txt",ios::out);
   for(int i=0; i<len; i++)
      file.put(string[i]);
   file.close();
   file.open("demo.txt",ios::in);
   char ch;
   while(file)
   {
      file.get(ch);
      cout<<ch;
   }
   file.close();
   return 0;
}
```

OUTPUT:

```
Enter a String : Natarajan s
Natarajann
```

RESULT:

Thus , the demonstration sequential I/O operation on a file has been executed successfully.



Ex. No : 13 –BIGGEST NUMBER USING COMMAND LINE ARGUMENTS

Aim : To write a C++ program to find the Biggest Number using Command Line Arguments.

Procedure :

Step 1: Start the Process

Step 2: Declare the variables i, n, large.

Step 3: Get the input from command line argument (argv[]) using DOS shell.

Step 4: Count the number of inputs and store in variable "n".

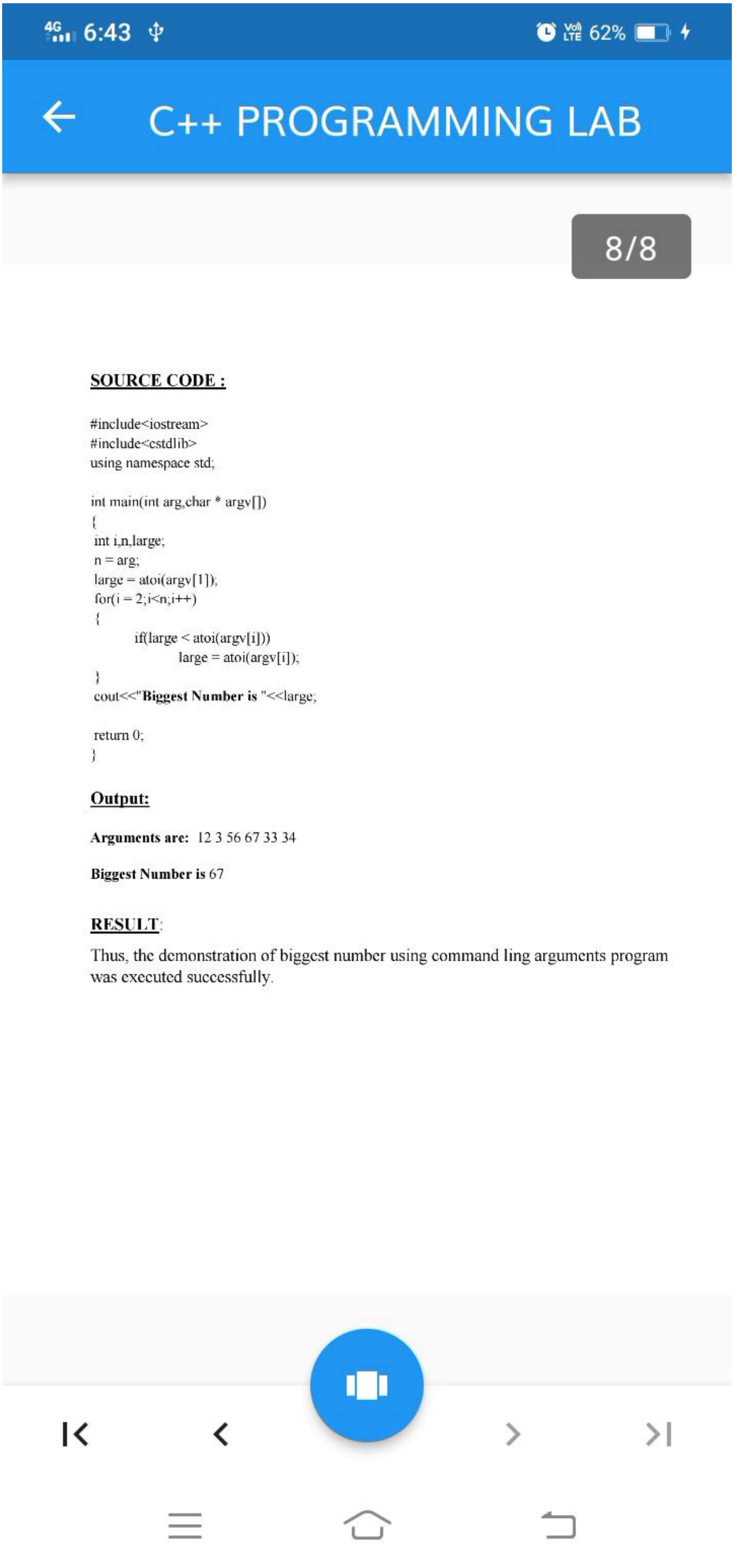
Step 5: Convert character input to integer using atoi() and store the value in variable "large".

Step 6: Compare the value in variable "large" with the remaining arguments of argv[] and store the largest value in variable "large".

Step 7: Display the largest value.

Step 8: Stop the Process.

FLOWCHART



SOURCE CODE :

```
#include<iostream>
#include<cstdlib>
using namespace std;

int main(int arg,char * argv[])
{
 int i,n,large;
 n = arg;
 large = atoi(argv[1]);
 for(i = 2;i<n;i++)
 {
         if(large < atoi(argv[i]))
                 large = atoi(argv[i]);
 }
 cout<<"Biggest Number is "<<large;

 return 0;
}
```

Output:

```
Arguments are:  12 3 56 67 33 34

Biggest Number is 67
```

RESULT:

Thus, the demonstration of biggest number using command ling arguments program was executed successfully.