```
package sakshi50;

import java.io.*;

import java.util.*;


public class macroprocessor_passone {

	static List<String>MDT;

	static Map<String,String>MNT;

	static int mntPtr,mdtPtr;

	static Map<String,String>ALA;


	public static void main (String []args) {

		try {

			pass1();

		}catch(Exception ex) {

			ex.printStackTrace();

		}

	}

	static void pass1() throws Exception{

		MDT=new ArrayList<String>();

		MNT=new LinkedHashMap<String,String>();

		ALA=new HashMap<String,String>();

		mntPtr=0;mdtPtr=0;

		BufferedReader input=new BufferedReader(new InputStreamReader(new
FileInputStream("C:\\Users\\student\\Desktop\\sakshi\\sakshi\\input.txt")));

		PrintWriter out_pass1=new PrintWriter(new
FileWriter("C:\\Users\\student\\Desktop\\OUTPUT1.txt"),true);
```

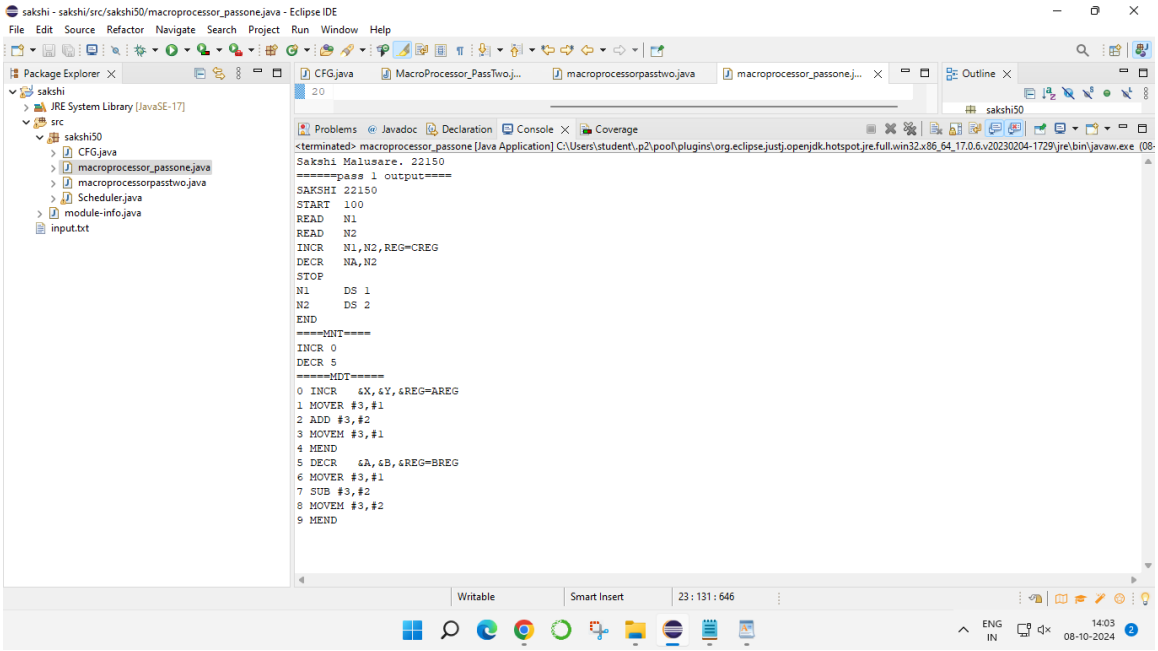
```
PrintWriter out_mnt=new PrintWriter(new
FileWriter("C:\\Users\\student\\Desktop\\MNT.txt"),true);

PrintWriter out_mdt=new PrintWriter(new
FileWriter("C:\\Users\\student\\Desktop\\MDT.txt"),true);

String s;

boolean processingMacroDefinition=false;

boolean processMacroName=false;

System.out.println("======pass 1 output====");

while ((s=input.readLine())!= null){

    String s_arr[]=tokenizeString(s," ");

    String curToken=s_arr[0];

    if(curToken.equalsIgnoreCase("MACRO")) {

        processingMacroDefinition=true;

        processMacroName=true;

    }

    else if(processingMacroDefinition==true) {

        if(curToken.equalsIgnoreCase("MEND")) {

            MDT.add(mdtPtr++,s);

            processingMacroDefinition=false;

            continue;

    }

        if(processMacroName==true) {

            MNT.put(curToken, mdtPtr+"");

            mntPtr++;

            processMacroName=false;

            processArgumentList(s_arr[1]);
```

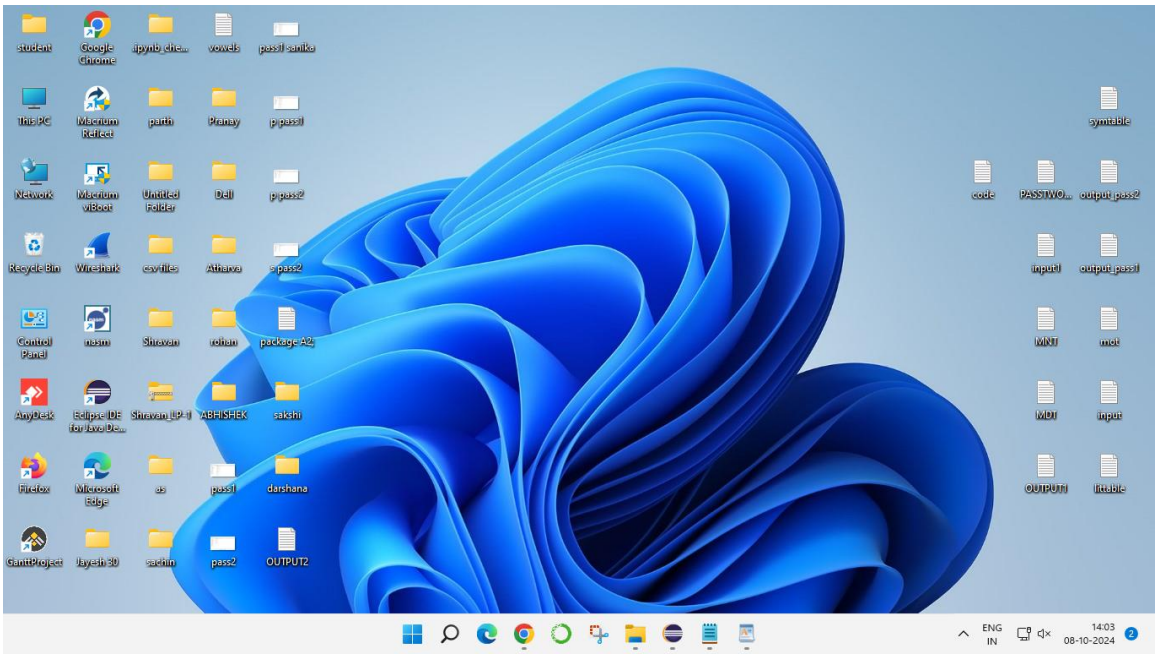
```
                    MDT.add(mdtPtr,s);
                    mdtPtr++;
                    continue;

                }
                String indexedArgList=processArguments(s_arr[1]);
                MDT.add(mdtPtr++,curToken+" "+indexedArgList);
}
        else{
                System.out.println(s);
                out_pass1.println(s);
                }
        }
input.close();
System.out.println("====MNT====");
Iterator<String>itMNT=MNT.keySet().iterator();
String key,mntRow,mdtRow;
while(itMNT.hasNext()) {
        key=(String)itMNT.next();
        mntRow=key+" "+MNT.get(key);
        System.out.println(mntRow);
        out_mnt.println(mntRow);
        }
System.out.println("=====MDT=====");
for(int i=0;i<MDT.size();i++) {
```

```
                mdtRow=i+" "+MDT.get(i);
                System.out.println(mdtRow);
                out_mdt.println(mdtRow);
                }
        out_pass1.close();
        out_mnt.close();
        out_mdt.close();

        }
static void processArgumentList(String argList) {
        StringTokenizer st= new StringTokenizer(argList, ",",false);
        ALA.clear();
        int argCount =st.countTokens();
        String curArg;
        for(int i=1;i<=argCount;i++) {
                curArg=st.nextToken();
                if(curArg.contains("=")) {
                        curArg=curArg.substring(0,curArg.indexOf("="));
                }
                ALA.put(curArg, "#"+i);
        }
}
static String processArguments(String argList) {
        StringTokenizer st=new StringTokenizer(argList,",",false);
        int argCount=st.countTokens();
```

```
		String curArg,argIndexed;
		for(int i=0;i<argCount;i++) {
			curArg=st.nextToken();
			argIndexed=ALA.get(curArg);
			argList=argList.replaceAll(curArg,argIndexed);
		}
		return argList;
	}
	static String[]tokenizeString(String str,String seperator){
		StringTokenizer st=new StringTokenizer(str,seperator,false);
		String s_arr[]=new String [st.countTokens()];
		for(int i=0;i<s_arr.length;i++) {
			s_arr[i]=st.nextToken();
		}
		return s_arr;
	}
}
```

output:


sakshi - sakshi/src/sakshi50/macroprocessor_passone.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
sakshi
JRE System Library [JavaSE-17]
src
sakshi50
CFG.java
macroprocessor_passone.java
macroprocessorpasstwo.java
Scheduler.java
module-info.java
input.txt
CFG.java
MacroProcessor_PassTwo.j...
macroprocessorpasstwo.java
macroprocessor_passone.j...
Outline
sakshi50
Problems
Javadoc
Declaration
Console
Coverage
<terminated> macroprocessor_passone [Java Application] C:\Users\student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (08-
Sakshi Malusare. 22150
======pass 1 output====
SAKSHI 22150
START 100
READ N1
READ N2
INCR N1,N2,REG=CREG
DECR NA,N2
STOP
N1 DS 1
N2 DS 2
END
====MNT====
INCR 0
DECR 5
=====MDT=====
0 INCR &X,&Y,&REG=AREG
1 MOVER #3,#1
2 ADD #3,#2
3 MOVEM #3,#1
4 MEND
5 DECR &A,&B,&REG=BREG
6 MOVER #3,#1
7 SUB #3,#2
8 MOVEM #3,#2
9 MEND
Writable
Smart Insert
23 : 131 : 646
ENG
IN
14:03
08-10-2024


student
Google Chrome
ipynb_che...
vowels
pass1 sanika
This PC
Macrium Reflect
parth
Pranay
p pass1
symtable
Network
Macrium viBoot
Untitled Folder
Dell
p pass2
code
PASSTWO...
output_pass2
Recycle Bin
Wireshark
csvfiles
Athawa
s pass2
input1
output_pass1
Control Panel
nasm
Shravan
rohan
package A2
MNT
mot
AnyDesk
Eclipse IDE for Java De...
Shravan_LP-I
ABHISHEK
sakshi
MDT
input
Firefox
Microsoft Edge
os
pass1
darshana
OUTPUT1
littable
GanttProject
layesh 30
sachin
pass2
OUTPUT2
ENG
IN
14:03
08-10-2024

output_pass1.txt | MNT.txt | MDT.txt | OUTPUT1.txt | PASSTWOOUTPUT.txt | input1.txt

File Edit View

```
INCR 0
DECR 5
```

Ln 1, Col 1 | 14 characters | 100% | Windows (CRLF) | UTF-8

ENG IN 14:03 08-10-2024

output_pass1.txt | MNT.txt | MDT.txt | OUTPUT1.txt | PASSTWOOUTPUT.txt | input1.txt

File Edit View

```
0 INCR   &X,&Y,&REG=AREG
1 MOVER #3,#1
2 ADD #3,#2
3 MOVEM #3,#1
4 MEND
5 DECR   &A,&B,&REG=BREG
6 MOVER #3,#1
7 SUB #3,#2
8 MOVEM #3,#2
9 MEND
```

Ln 1, Col 1 | 144 characters | 100% | Windows (CRLF) | UTF-8

ENG IN 14:04 08-10-2024

output_pass1.txt | MNT.txt | MDT.txt | OUTPUT1.txt | PASSTWOOUTPUT.txt | input1.txt

File Edit View

```
SAKSHI 22150
START  100
READ   N1
READ   N2
INCR   N1,N2,REG=CREG
DECR   NA,N2
STOP
N1     DS 1
N2     DS 2
END
```

Ln 1, Col 1 | 112 characters | 100% | Windows (CRLF) | UTF-8

ENG IN 14:04 08-10-2024