```
Pass2.java
 Open V H
                                                                                                                  Save ≡ − □ ×
limport java.io.BufferedReader;
limport java.io.FileReader;
 3 import java.io.FileWriter;
4 import java.io.IOException;
5 import java.util.HashMap;
  public class Pass2 {
    private static final String INTERMEDIATE_CODE_FILE = "/home/student/Desktop/LP1/IC.txt";
    private static final String SYM_TAB_FILE = "/home/student/Desktop/LP1/SYMTAB.txt";
    private static final String LITERALS_FILE = "/home/student/Desktop/LP1/LITTAB.txt";
        private static final String OUTPUT FILE = "/home/student/Desktop/LP1/Pass2.txt";
        public static void main(String[] args) {
             HashMap<Integer, String> symSymbol = new HashMap<>();
HashMap<Integer, String> litAddr = new HashMap<>();
                   BufferedReader b1 = new BufferedReader(new FileReader(INTERMEDIATE_CODE_FILE));
                   BufferedReader b2 = new BufferedReader(new FileReader(SYM TAB FILE));
                   BufferedReader b3 = new BufferedReader(new FileReader(LITERALS FILE));
                   FileWriter f1 = new FileWriter(OUTPUT_FILE)
                   String line;
25
                   // Read symbol table: format assumed [index] [address]
while ((line = b2.readLine()) != null) {
   String[] words = line.trim().split("\\s+");
                         if (words.length >= 2) {
                              int index = Integer.parseInt(words[0]);
                              symSymbol.put(index, words[1].trim());
                   int litIndex = 1;
                   while ((line = b3.readLine()) != null) {
    String[] words = line.trim().split("\\s+");
                         if (words.length >= 2) {
                              litAddr.put(litIndex++, words[1].trim());
                   while ((line = b1.readLine()) != null) {
                         line = line.trim();
if (line.isEmpty()) continue;
                         String[] parts = line.split("\\s+");
                         StringBuilder output = new StringBuilder("+ ");
                         for (String part : parts) {
                              part = part.trim();
                              if (part.startsWith("(IS,")) {
                                    String opcode = part.substring(4, part.length() - 1);
                                    output.append(opcode).append(" ");
                              } else if (part.matches("\\d+")) {
                              output.append(part).append(" ");
} else if (part.startsWith("(R,")) {
                                    String reg = part.substring(3, part.length() - 1);
                              output.append(reg).append(" ");
} else if (part.startsWith("(S,")) {
                                    int index = Integer.parseInt(part.substring(3, part.length() - 1));
                                    output.append(symSymbol.getOrDefault(index, "000")).append(" ");
                              } else if (part.startsWith("(L,")) {
   int index = Integer.parseInt(part.substring(3, part.length() - 1));
   output.append(litAddr.getOrDefault(index, "000")).append(" ");
                              } else if (part.startsWith("(C,")) {
                                    String constant = part.substring(3, part.length() - 1);
output.append(constant).append(" ");
                              } else if (part.contains("+")) {
                             // Handles expressions like (S,1)+1
String[] expr = part.split("\\+");
                             String symbolPart = expr[0].trim();
```

```
String symbolPart = expr[0].trim();
                         String offsetPart = expr[1].trim();
        \textbf{if} \ (\text{symbolPart.startsWith}("(\textbf{S},") \ \&\& \ \text{symbolPart.endsWith}(")")) \ \{ \\
           int index = Integer.parseInt(symbolPart.substring(3, symbolPart.length() - 1));
           int base = Integer.parseInt(symSymbol.getOrDefault(index, "0"));
           int offset = Integer.parseInt(offsetPart);
           output.append(base + offset).append(" ");
                    f1.write(output.toString().trim() + "\n");
           } catch (IOException e) {
                System.err.println("Error occurred while processing files: " + e.getMessage());
          student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN: ~/Desktop/LP1
                                                                                      Q
(base) student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN:~/Desktop/LP1$ javac Pass2.java
(base) student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN:~/Desktop/LP1$ java Pass2
Exception in thread "main" java.lang.NumberFormatException: For input string: "1)+"
at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java
:67)
         at java.base/java.lang.Integer.parseInt(Integer.java:662)
         at java.base/java.lang.Integer.parseInt(Integer.java:778) at Pass2.main(Pass2.java:63)
(base) student@student-ASUS-EXPERTCENTER-D500MD-D500MD-IN:~/Desktop/LP1$
```

