

# SPOS

## Practical no 1

```
import java.io.FileInputStream;import
java.io.FileOutputStream;import
java.util.ArrayList; import
java.util.HashMap;
import java.util.StringTokenizer;import
java.io.IOException;

class Tuple {
    String mnemonic, mclass, opcode;int
    length;

    Tuple(String s1, String s2, String s3, String s4) {mnemonic =
        s1;
        mclass = s2;
        opcode = s3;
        length = Integer.parseInt(s4.trim());
    }
}

public class Prog {
    public static HashMap<String, Tuple> map = new HashMap<String,
Tuple>();
    public static HashMap<String, Integer> registers = new
HashMap<String, Integer>();
    public static ArrayList<String> literals = new ArrayList<String>();public static
ArrayList<String> symbols = new ArrayList<String>();

    Prog() {
        registers.put("AREG", 1);
        registers.put("BREG", 2);
        registers.put("CREG", 3);
        registers.put("DREG", 4);
    }

    public static void mapper() {try {
        String newSt = "";
        FileInputStream input = new FileInputStream("your file
path");

        int i = input.read();while
        (i != -1) {
            newSt += (char) i;i =
            input.read();
        }
        input.close();
        StringTokenizer st = new StringTokenizer(newSt, " ");String sst =
        "";
        while (st.hasMoreTokens()) { sst +=
            st.nextToken() + " ";
        }
    }
}
```

```

        System.out.println(e);
    }

}

public static String[] inputFileRead() {String
    newSt = "";
    String[] arr2 = {};try {
        //Reading the input file
        FileInputStream input = new FileInputStream("your file
path");

        int i = input.read();while
        (i != -1) {
            newSt += (char) i;i =
            input.read();
        }
        input.close();
        //Tokenization line by line
        StringTokenizer st = new StringTokenizer(newSt, " ");String sst =
        "";
        while (st.hasMoreTokens()) { sst +=
            st.nextToken() + " ";
        }
        sst = sst.toString(); arr2 =
        sst.split("\n");
    } catch (Exception e) { System.out.println("Something went
        wrong!" + e);
    }
    return arr2;
}

```

```

public static void intermediateCoder() {mapper();
    String[] inputArr = inputFileRead();

    String sst = ""; String
    forLiteral = "";String
    forSymbol = "";
    int addressStart = 0, address = 0;
    int addressCounter = 1; //Location counterfor (int i =
    0; i < inputArr.length; i++) {
        inputArr[i] = inputArr[i].trim();
        inputArr[i] = inputArr[i].replaceAll(", ", "");

        Tuple value = map.get(inputArr[i].split(" ")[0]);if
        (value==null){
            value = map.get(inputArr[i].split(" ")[1]);
        }

        String mclass = value.mclass;String
        opcode = value.opcode;int length =
        value.length;
        //For handling Assembler Directives
        if (value.mclass.equalsIgnoreCase("AD")) {
            if(inputArr[i].split(" ").length>1){
                sst = sst + "(" + String.format("%s, %s", mclass, opcode)

```

```

        + String.format("C,%s", inputArr[i].split(" ")[1].trim()
+ ")" + "\n");
        System.out.println(inputArr[i].split(" ")[1].trim()); addressStart =
        Integer.parseInt(inputArr[i].split("
")[1].trim());
    }
    else {
        sst = sst + "(" + String.format("%s, %s", mclass,
opcode) + ")" + "\n";
        System.out.println(sst);
    }

    } else if (value.mclass.equalsIgnoreCase("DL")) {
        sst = sst + "(" + String.format("%s, %s", mclass, opcode)
+ ")" + "\t" + "("
        + String.format("%s", inputArr[i].split("
")[2].trim() + ") + "\n");
        System.out.println(sst);

    } else {

        // Literal Case address =
        addressStart;
        if (inputArr[i].split(" ")[2].startsWith("=")) {
            //System.out.println("Here1");
            System.out.println(inputArr[i].split(" ")[2]);
            literals.add(inputArr[i].split("
")[2].split("=")[1]);
            sst = sst + "(" + String.format("%s, %s", mclass, opcode) +
)" + "\t" + "(" + String.format("%s",
registers.get(inputArr[i].split("
")[1].split(",")[0].trim() + ") + "("
        + String.format("L,%s",
literals.indexOf(inputArr[i].split(" ")[2].split("=")[1]))
+ ")" + "\n");
            address = addressCounter + address; forLiteral =
forLiteral + inputArr[i].split("
")[2].trim() + "\t" + String.format("%s", address)
+ "\n";
            addressCounter++;
        }
        // Symbol case
        else {
            System.out.println(inputArr[i].split(" ")[2]);
            symbols.add(inputArr[i].split(" ")[2]); address =
addressCounter + address;
            sst = sst + "(" + String.format("%s, %s", mclass, opcode) +
)" + "\t" + "("
        + String.format("%s",
registers.get(inputArr[i].split("
")[1].split(",")[0].trim() + ") + "("
        + String.format("S,%s",
symbols.indexOf(inputArr[i].split(" ")[2])) + ")"
+ "\n");
            forSymbol = forSymbol + inputArr[i].split(" ")[2] + "\t" +

```

```

        }
    }

    sst = sst.toString();
    forLiteral = forLiteral.toString(); forSymbol
    = forSymbol.toString(); try {
        FileOutputStream output = new FileOutputStream("./out.txt");
        FileOutputStream literalOut = new
FileOutputStream("./literalTable.txt");
        FileOutputStream symbolOut = new
FileOutputStream("./symbolTable.txt");
        output.write(sst.getBytes());
        literalOut.write(forLiteral.getBytes());
        symbolOut.write(forSymbol.getBytes());
        output.close();
        literalOut.close();
        symbolOut.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {Prog pg =
    new Prog(); intermediateCoder();
}
}

```

## INPUT CODE

100 (01,AD) (C,100)  
100 (04,IS) 1 (S,1)  
101 (01,IS) 2 (L,1)  
102 (05,IS) 1 (S,2)  
103 (02,IS) 3 (L,2)  
(05,AD)  
104 6  
105 1  
106 (01,IS) 4 (L,3)  
107 (01,DL) (C,10)  
(05,AD)  
117 5  
118 (02,IS) 1 (L,4)  
119 (02,DL) (C,1)  
(02,AD)  
120 1

## Literal Table

1 ='6' 104  
2 ='1' 105  
3 ='5' 117  
4 ='1' 120

## MOT

START AD 01 0  
END AD 02 0  
LORG AD 05 0  
ADD IS 01 1  
SUB IS 02 1  
MULT IS 03 1  
MOVER IS 04 1  
MOVEM IS 05 1  
DS DL 01 0  
DC DL 02 1

## Symbol Table

1 B 119  
2 A 107

## MOT

```
START  AD 01 0
END     AD 02 0
LTORG   AD 05 0
ADD     IS 01 1
SUB     IS 02 1
MULT    IS 03 1
MOVER   IS 04 1
MOVEM   IS 05 1
DS      DL 01 0
DC      DL 02 1
```

## INPUT

```
START 100
      MOVER AREG,B
      ADD   BREG,='6'
      MOVEM AREG,A
      SUB   CREG,='1'
      LTORG
      ADD   DREG,='5'
A      DS   10
      LTORG
      SUB   AREG,='1'
B      DC   1
C      DC   1
      END
```

## OUTPUT

### Intermidate code

```
100 (01,AD) (C,100)
100 (04,IS) 1 (S,1)
101 (01,IS) 2 (L,1)
102 (05,IS) 1 (S,2)
103 (02,IS) 3 (L,2)
(05,AD)
104 6
105 1
106 (01,IS) 4 (L,3)
107 (01,DL) (C,10)
(05,AD)
117 5
118 (02,IS) 1 (L,4)
119 (02,DL) (C,1)
(02,AD)
120 1
```

### Symbol Table

```
1 B 119
2 A 107
```





## Practical no 2

### PASS1:

```
import java.util.*;
import java.io.*;

class MDT {
    int index;
    String def;
    MDT(int i, String a) {
        this.index = i;
        this.def = a;
    }
}

class MNT {
    int index,mdtind;
    String name;
    Vector<String> ala;

    MNT(int i, String a, int ind, Vector<String> b) {
        this.index = i;
        this.name = a;
        this.mdtind = ind;
        this.ala = b;
    }
}

class macro1{
    public static int searchmnt(Vector<MNT> a, String b) {
        int i,pos = -1;
        for(i = 0; i < a.size(); i++) {
            MNT x = a.get(i);
            if(x.name.equals(b)) { pos = i; break; }
```

```

}

return pos;

}

public static void main(String[] args) throws Exception {

BufferedReader r = new BufferedReader(new
FileReader("input.txt"));

BufferedReader r1 = new BufferedReader(new InputStreamReader(System.in));

Vector<MDT> mdt = new Vector<MDT>();

Vector<MNT> mnt = new Vector<MNT>();

Vector<String> isc = new Vector<String>();

String d;

boolean flag = false;

/* PASS 1 */

while((d = r.readLine()) != null) {

d = d.trim();

if(d.isEmpty()) continue;

String temp[] = d.split("\\s+");

if(d.contains("MACRO")) flag = true; //start entering into MDT

if(flag) {

d = r.readLine();

String t[] = d.split("\\s+");

String t1[] = t[1].split("\\.");

Vector<String> arg = new Vector<String>();

for(int i = 0; i < t1.length; i++) {

String p = t1[i];

if(t1[i].contains("=")) p = t1[i].substring(0,t1[i].indexOf('='));

arg.addElement(p);

}

mnt.addElement(new MNT(mnt.size() + 1, t[0], mdt.size() + 1, arg));

mdt.addElement(new MDT(mdt.size() + 1, d));

d = r.readLine();

```

```

while(!d.equals("MEND")) {
    if(d.contains("&")) {
        int i = d.indexOf('&');
        MNT h = mnt.get(mnt.size() - 1);
        int j;
        for(j = 0; j < h.ala.size(); j++) {
            if(d.substring(i).equals(h.ala.get(j)))
                break;
        }
        mdt.addElement(new MDT(mdt.size() + 1, d.substring(0,i)+"#"+(j+1)));
    }
    else mdt.addElement(new MDT(mdt.size() + 1, d));
    d = r.readLine();
}
if(d.equals("MEND")) mdt.addElement(new MDT(mdt.size() + 1, d));
flag = false;
}
else {
    isc.addElement(d);
}
}

System.out.println("\nPASS 1\n");
System.out.println("MDT");
for(int i = 0; i < mdt.size(); i++) {
    MDT t = mdt.get(i);
    System.out.println(t.index+" "+t.def);
}

System.out.println("\nMNT");
for(int i = 0; i < mnt.size(); i++) {
    MNT t = mnt.get(i);
    System.out.print(t.index+" "+t.name+" "+t.mdtind+"\tALA: ");

```

```

for(int j = 0; j < t.ala.size(); j++) {
    System.out.print(t.ala.get(j)+" ");
}
System.out.println();
}
System.out.println("\nIntermediate code");
for(int i = 0; i < isc.size(); i++) System.out.println(isc.get(i));

}
}

```

## INPUT:

MACRO

INCR1 &FIRST,&SECOND=DATA9

A 1,&FIRST

L 2,&SECOND

MEND

MACRO

INCR2 &ARG1,&ARG2

L 3,&ARG1

ST 4,&ARG2

MEND

PRG2 START

USING \*,BASE

INCR1 DATA1,DATA12

INCR2 DATA3,DATA4

FOUR DC F&#39;4&#39;

FIVE DC F&#39;5&#39;

BASE EQU 8

TEMP DS &#39;1&#39;F

DROP 8

END

## PASS2:

```
import java.util.*;
import java.io.*;

class MDT {
    int index;
    String def;
    MDT(int i, String a) {
        this.index = i;
        this.def = a;
    }
}

class MNT {
    int index,mdtind;
    String name;
    Vector<String> ala;
    MNT(int i, String a, int ind, Vector<String> b) {
        this.index = i;
        this.name = a;
        this.mdtind = ind;
        this.ala = b;
    }
}

class ass2pass2{
    public static int searchmnt(Vector<MNT> a, String b) {
        int i,pos = -1;
        for(i = 0; i < a.size(); i++) {
            MNT x = a.get(i);
            if(x.name.equals(b)) { pos = i; break; }
        }
        return pos;
    }
}
```

```

public static void main(String[] args) throws Exception {
    BufferedReader r = new BufferedReader(new
    FileReader("input.txt"));

    BufferedReader r1 = new BufferedReader(new InputStreamReader(System.in));

    Vector<MDT> mdt = new Vector<MDT>();
    Vector<MNT> mnt = new Vector<MNT>();
    Vector<String> isc = new Vector<String>();

    String d;

    boolean flag = false;

    while((d = r.readLine()) != null) {

        d = d.trim();

        if(d.isEmpty()) continue;

        String temp[] = d.split("\\s+");

        if(d.contains("MACRO")) flag = true; //start entering into MDT

        if(flag) {

            d = r.readLine();

            String t[] = d.split("\\s+");

            String t1[] = t[1].split("\\.");

            Vector<String> arg = new Vector<String>();

            for(int i = 0; i < t1.length; i++) {

                String p = t1[i];

                if(t1[i].contains("=")) p = t1[i].substring(0,t1[i].indexOf('='));

                arg.addElement(p);

            }

            mnt.addElement(new MNT(mnt.size() + 1, t[0], mdt.size() + 1, arg));

            mdt.addElement(new MDT(mdt.size() + 1, d));

            d = r.readLine();

            while(!d.equals("MEND")) {

                if(d.contains("&")) {

                    int i = d.indexOf('&');

                    MNT h = mnt.get(mnt.size() - 1);

```

```

int j;
for(j = 0; j < h.ala.size(); j++) {
    if(d.substring(i).equals(h.ala.get(j)))
        break;
}
mdt.addElement(new MDT(mdt.size() + 1, d.substring(0,i)+"#"+(j+1)));
}
else mdt.addElement(new MDT(mdt.size() + 1, d));
d = r.readLine();
}
if(d.equals("MEND")) mdt.addElement(new MDT(mdt.size() + 1, d));
flag = false;
}
else {
    isc.addElement(d);
}
}
System.out.println(" -----\nPASS 2\n");
/* PASS 2 */
for(int i = 0; i < isc.size(); i++) {
    String temp[] = isc.get(i).split("\\s+");
    int pos1 = searchmnt(mnt,temp[0]);
    if(pos1 == -1) System.out.println(isc.get(i));
    else if(pos1 != -1) {
        MNT x = mnt.get(pos1);
        int mdt_ind = x.mdtind;
        String ala1[] = temp[1].split("\\.");
        //System.out.println(ala1[0] + " " + ala1[1]);
        StringBuffer ss = new StringBuffer();
        for(int j = mdt_ind; j < mdt.size(); j++) {
            flag = false;

```

```

String def = mdt.get(j).def;
if(def.equals("MEND")) break;
else {
for(int k = 0; k < def.length(); k++) {
if(!flag && def.charAt(k) != '#') System.out.print(def.charAt(k));
else if(!flag && def.charAt(k) == '#') flag = true;
else if(flag && def.charAt(k) == ',') {
//System.out.println(ss.toString());
int pos = Integer.parseInt(ss.toString());
System.out.print(ala1[pos - 1]);
ss.delete(0, ss.length());
flag = false;
}
else if(flag) {
ss.append(def.charAt(k));
}
}
if(ss.length() > 0) {
int pos = Integer.parseInt(ss.toString());
System.out.print(ala1[pos - 1]);
System.out.println();
ss.delete(0, ss.length());
}
}
}
}
}
}
}
}
}
}
}

```



## OUTPUT:

```
student@student:~$ javac ass1pass1.java
student@student:~$ java macro1 input.txt

PASS 1

MDT
1 INCR1 & ;FIRST,& ;SECOND=DATA9
2 A 1,#1
3 L 2,#2
4 MEND
5 INCR2 & ;ARG1,& ;ARG2
6 L 3,#1
7 ST 4,#2
8 MEND
```

```
MNT
1 INCR1 1          ALA: & ;FIRST & ;SECOND
2 INCR2 5          ALA: & ;ARG1 & ;ARG2
```

```
Intermediate code
PRG2 START
USING *,BASE
INCR1 DATA1,DATA12
INCR2 DATA3,DATA4
FOUR DC F&#39;4&#39;
FIVE DC F&#39;5&#39;
BASE EQU 8
TEMP DS &#39;1&#39;F
DROP 8
END
```

```
student@student:~$ javac ass2pass2.java
student@student:~$ java ass2pass2 input.txt
```

```
-----
PASS 2
```

```
PRG2 START
USING *,BASE
A 1,DATA1
L 2,DATA12
L 3,DATA3
ST 4,DATA4
FOUR DC F&#39;4&#39;
FIVE DC F&#39;5&#39;
BASE EQU 8
TEMP DS &#39;1&#39;F
DROP 8
END
```



## Practical no 3

### **dllclient.cpp**

```
#include "windows.h"

#include <iostream>

#include "dlldemo.h"

using namespace std;

int main()
{
    double a;

    int b;

    cout << "Enter value of a";

    cin >> a;

    cout << "Enter value of b";

    cin >> b;

    cout << "a + b = " <<

        dlldemo::Functions::Add(a, b) << endl;

    cout << "a - b = " <<

        dlldemo::Functions::sub(a, b) << endl;

    cout << "a / b = " <<

        dlldemo::Functions::div(a, b) << endl;

    cout << "a * b = " <<

        dlldemo::Functions::Multiply(a, b) << endl;

    cout << "a + (a * b) = " <<

        dlldemo::Functions::AddMultiply(a, b) << endl;

    return 0;
}
```

### **dlldemo.h**

```
namespace dlldemo
{
    // This class is exported from the MathLibrary.dll
```

```

class Functions
{
public:
    // Returns a + b
    static MATHLIBRARY_API double Add(double a, double b);

    // Returns a - b
    static MATHLIBRARY_API double sub(double a, double b);

    // Returns a / b
    static MATHLIBRARY_API double div(double a, double b);

    // Returns a * b
    static MATHLIBRARY_API double Multiply(double a, double b);

    // Returns a + (a * b)
    static MATHLIBRARY_API double AddMultiply(double a, double b);
};
};

```

### **dlldemo.cpp**

```

#include "windows.h"
#include "dlldemo.h"
namespace dlldemo
{
    double Functions::Add(double a, double b)
    {
        return a + b;
    }

    double Functions::sub(double a, double b)
    {
        return a - b;
    }

    double Functions::div(double a, double b)
    {

```

```

        return a / b;
    }

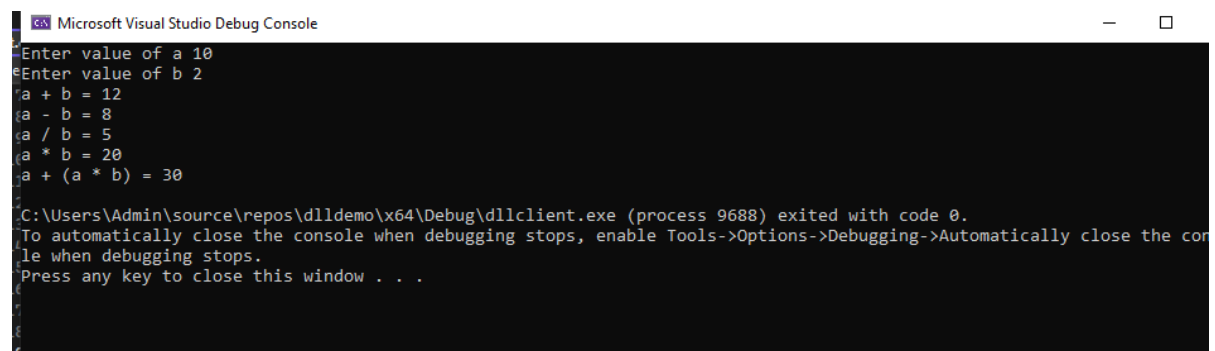
double Functions::Multiply(double a, double b)
{
    return a * b;
}

double Functions::AddMultiply(double a, double b)
{
    return a + (a * b);
}

};

```

### OUTPUT:



The screenshot shows the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```

Enter value of a 10
Enter value of b 2
a + b = 12
a - b = 8
a / b = 5
a * b = 20
a + (a * b) = 30

C:\Users\Admin\source\repos\dlldemo\x64\Debug\dllclient.exe (process 9688) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```