```
-- file: Output.mesa
-- edited by Sandman on August 3, 1978  11:50 AM

DIRECTORY
  IODefs: FROM "iodefs",
  OutputDefs: FROM "outputdefs",
  SegmentDefs: FROM "segmentdefs",
  StreamDefs: FROM "streamdefs",
  StringDefs: FROM "stringdefs",
  TimeDefs: FROM "timedefs";

Output: PROGRAM
  IMPORTS SegmentDefs, StreamDefs, StringDefs, TimeDefs
  EXPORTS OutputDefs =
PUBLIC BEGIN

outStream: StreamDefs.StreamHandle ← NIL;

GetOutputStream: PROCEDURE RETURNS [StreamDefs.StreamHandle] =
  BEGIN
  RETURN[outStream];
  END;

PutTab: PROCEDURE = BEGIN PutChar[IODefs.TAB]; RETURN END;
PutCR: PROCEDURE = BEGIN PutChar[IODefs.CR]; RETURN END;

PutChar: PROCEDURE [c:CHARACTER] =
  BEGIN
  outStream.put[outStream,c];
  END;

PutString: PROCEDURE [s:STRING] =
  BEGIN
  i: CARDINAL;
  FOR i IN [0..s.length) DO
    outStream.put[outStream,s[i]];
    ENDLOOP;
  END;

PutSubString: PROCEDURE [s:StringDefs.SubString] =
  BEGIN
  i: CARDINAL;
  FOR i IN [s.offset..s.offset+s.length) DO
    outStream.put[outStream,s.base[i]];
    ENDLOOP;
  END;

PutTime: PROCEDURE [t: TimeDefs.PackedTime] =
  BEGIN
  OPEN TimeDefs;
  s: STRING ← [20];
  AppendDayTime[s,UnpackDT[t]];
  PutString[s];
  RETURN
  END;

PutNumber: PROCEDURE [val: CARDINAL, format: IODefs.NumberFormat] =
  BEGIN
  i: CARDINAL;
  neg: BOOLEAN ← FALSE;
  fill: CHARACTER ← (IF format.zerofill THEN '0 ELSE ' );
  s: STRING ← [10];
  IF INTEGER[val] < 0 AND ~format.unsigned THEN
    BEGIN val ← -INTEGER[val]; neg ← TRUE; END;
  StringDefs.AppendNumber[s, val, format.base];
  i ← s.length;
  IF neg THEN
    BEGIN
    i ← i + 1;
    IF format.zerofill THEN
      BEGIN outStream.put[outStream,'-]; neg ← FALSE END;
    END;
  THROUGH (i..format.columns] DO outStream.put[outStream,fill] ENDLOOP;
  IF neg THEN outStream.put[outStream,'-];
  PutString[s];
  RETURN
  END;

PutDecimal: PROCEDURE [val: CARDINAL] =
  BEGIN
  PutNumber[val,[10,FALSE,FALSE,1]];
  RETURN
  END;

PutOctal: PROCEDURE [val: UNSPECIFIED] =
  BEGIN
  PutNumber[val,[8,FALSE,TRUE,1]];
  IF val ~IN [0..7] THEN outStream.put[outStream,'B];
  RETURN
  END;

OpenOutput: PROCEDURE [root: STRING, ext: STRING] =
  BEGIN OPEN StringDefs, SegmentDefs;
  i: CARDINAL;
  name: STRING ← [40];
  AppendString[name,root];
  FOR i IN [0..name.length) DO
    IF name[i] = '. THEN
      BEGIN name.length ← i; EXIT END;
    ENDLOOP;
  AppendString[name,ext];
  outStream ← StreamDefs.CreateByteStream[
    NewFile[name,Write+Append,DefaultVersion],Write+Append];
  END;

CloseOutput: PROCEDURE =
  BEGIN
  outStream.destroy[outStream];
  outStream ← NIL;
  END;

END...
```