INICIO::= INSTRUCCIONES  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }	ACCIONES
ID::= factorial	INICIO=INSTRUCCIONES;  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=factorial;
TIPO::= Entero  PARAMETRO::= TIPO ID []	TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID);
ID::= n  TIPO::= Entero  INSTRUCCION::= SWITCH( EXP ) { LCASOS }	ID=n; TIPO=Entero; INSTRUCCION=newSwitch(EXP, LCASOS);
EXP::= ID  ID::= n	EXP= ID; ID= id_(n);
LCASOS::= CASE EXP : INSTRUCCIONES  EXP::= 0  INSTRUCCION::= PETOPNO EXP.	LCASOS=newCaso(EXP, INSTRUCCIONES);  EXP= primitivo(0);  LCASOS=newPaterno(EXP);
INSTRUCCION::= RETORNO EXP;  EXP::= 1  LCASOS::= CASE EXP : INSTRUCCIONES	LCASOS=newRetorno(EXP);  EXP= primitivo(1);  LCASOS=newCaso(EXP, INSTRUCCIONES);
INSTRUCCION::= RETORNO EXP;  EXP::= EXP1 * EXP2	LCASOS=newRetorno(EXP);  EXP= newMultiplicacion(EXP1,EXP2);
EXP::= ID $ID::= n$ $EXP::= ID (PARAMETROS)$	EXP= ID;  ID= id_(n);  EXP=newLlamadaInstr(ID,PARAMETRO);
INSTRUCCION::= RETORNO EXP;  EXP::= - EXP	LCASOS=newRetorno(EXP);  EXP= newNegacion(EXP);
EXP::= 100  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= potencia	EXP= primitivo(100);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=potencia;
TIPO::= Doble  PARAMETRO::= TIPO ID []	TIPO=Doble;  PARAMETRO=newDecParam(TIPO,ID);
ID::= base  TIPO::= Doble  PARAMETRO::= TIPO ID []	ID=base; TIPO=Doble; PARAMETRO=newDecParam(TIPO,ID);
ID::= exp  TIPO::= Doble	ID=exp; TIPO=Doble;
INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;	INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP);
EXP::= 1  INSTRUCCION::= RETORNO EXP;  EXP::= EXP1 * EXP2	EXP= primitivo(1);  LCASOS=newRetorno(EXP);  EXP= newMultiplicacion(EXP1,EXP2);
EXT ::- EXT 1 EXT 2  EXP::= ID  ID::= base	EXI - newividiplication(EXI 1,EXI 2),  EXP= ID;  ID= id_(base);
EXP::= ID (PARAMETROS)  INSTRUCCION::= RETORNO EXP;  EXP::= - EXP	EXP=newLlamadaInstr(ID,PARAMETRO);  LCASOS=newRetorno(EXP);  EXP= newNegacion(EXP);
EXP::= 100  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }	EXF = newNegacion(EXF),  EXP = primitivo(100);  GLOBALES = newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)
ID::= mcd  TIPO::= Entero  PARAMETRO::= TIPO ID []	ID=mcd; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID);
ID::= n1  TIPO::= Entero	ID=n1; TIPO=Entero;
PARAMETRO::= TIPO ID []  ID::= n2  TIPO::= Entero	PARAMETRO=newDecParam(TIPO,ID);  ID=n2;  TIPO=Entero;
INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }	INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);
INSTRUCCION::= RETORNO EXP; EXP::= ID (PARAMETROS)	LCASOS=newRetorno(EXP); EXP=newLlamadaInstr(ID,PARAMETRO);
GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES } ID::= fibonacci TIPO::= Entero	GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=fibonacci;  TIPO=Entero;
PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero	PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;
INSTRUCCION::=LSUBIF POSELSE LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }	INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= n	LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(n);
INSTRUCCION::= RETORNO EXP; EXP::= EXP1 + EXP2	LCASOS=newRetorno(EXP);  EXP= newSuma(EXP1,EXP2);
EXP::= ID (PARAMETROS)  EXP::= ID (PARAMETROS)  INSTRUCCION::= RETORNO EXP;	EXP=newLlamadaInstr(ID,PARAMETRO);  EXP=newLlamadaInstr(ID,PARAMETRO);  LCASOS=newRetorno(EXP);
EXP::= - EXP  EXP::= 100	EXP= newNegacion(EXP);  EXP= primitivo(100);
GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES } ID::= hanoi TIPO::= Entero	GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hanoi;  TIPO=Entero;
PARAMETRO::= TIPO ID []  ID::= discos	PARAMETRO=newDecParam(TIPO,ID); ID=discos;
TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= origen	TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=origen;
TIPO::= Entero  PARAMETRO::= TIPO ID []	TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID);
ID::= auxiliar  TIPO::= Entero  PARAMETRO::= TIPO ID []	ID=auxiliar; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID);
ID::= destino  TIPO::= Entero	ID=destino; TIPO=Entero;
DECLARACION::= TIPO LDECLA ;  LDECLA::= LDECLA DECLA ;  DECLA::= ID POSC POSEXP	PARAMETRO=newDeclaracion(TIPO,LDECLA);  LDECLA.append(DECLA);  PARAMETRO=newDecla(ID, POSC, POSEXP);
ID::= result INSTRUCCION::=LSUBIF POSELSE	ID=result; INSTRUCCION=newIF(LSUBIF, POSELSE);
LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= VARIABLE INDICES = EXP;  VARIABLE::= result:	INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION=newAsignacion(VARIABLE,INDICES,EXP); VARIABLE=result;
INDICES::= INDICES1 [ EXP ] : INDICES::= [EXP] :	INDICES=INDICES1; INDICES.append(EXP); INDICES=[]; INDICES.append(EXP);
EXP::= EXP1 + EXP2  EXP::= ID  ID::= result	EXP= newSuma(EXP1,EXP2);  EXP= ID;  ID= id_(result);
EXP::= EXP1 + EXP2  EXP::= ID  ID::= origen	EXP= newSuma(EXP1,EXP2);  EXP= ID;  ID= id_(origen);
EXP::= ID  ID::= destino	11 <i>7</i> — 10 (0)195111.
	EXP= ID; ID= id_(destino);
INSTRUCCION::= RETORNO EXP; EXP::= ID	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;
INSTRUCCION::= RETORNO EXP;	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0	EXP= ID; ID= id_(destino); LCASOS=newRetorno(EXP); EXP= ID; ID= id_(result); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderFemenina; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderMasculino; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0);
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= ackermann	EXP= ID; ID= id_(destino); LCASOS=newRetorno(EXP); EXP= ID; ID= id_(result); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderFemenina; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderMasculino; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=ackermann;
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= ackermann  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= m	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=ackermann;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=m;
INSTRUCCION::= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::=LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= ackermann  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= m  TIPO::= Entero  PARAMETRO::= TIPO ID []	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=ackermann;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);
INSTRUCCION:= RETORNO EXP;  EXP:= IID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES,  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newIf(LSUBIF, POSELSE);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  ID=ncemple primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  ID=ncemple primitivo(0);  ID=m;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);
INSTRUCCION:= RETORNO EXP;  EXP:= IID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES)  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  ID=ackermann;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= ENTERO EXP;  EXP:= EXPI + EXP2  EXP:= ID  ID::= n	EXP= ID;  ID= id_(destino);  LCASOS=newRetomo(EXP);  EXP= ID;  ID- id_(result);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID-hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetomo(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID=hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetomo(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID);  ID=n;  TIPO=Entero;  PARAMETRO=newBubIF(EXP, INSTRUCCIONES);  LCASOS=newRetomo(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID=ackermann;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=n
INSTRUCCION:= RETORNO EXP;  EXP::= ID  ID::= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderFemenina  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= hofstaderMasculino  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= 0  GLOBALES::= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID::= ackermann  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= m  TIPO::= Entero  PARAMETRO::= TIPO ID []  ID::= n  TIPO::= Entero  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= LSUBIF POSELSE  LSUBIF::= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION::= RETORNO EXP;  EXP::= EXPI + EXP2  EXP::= ID	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID=n;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  ID=ackerman;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDeeParam(TIPO,ID);  ID=n;  TIPO=Entero;  NSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newIF(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= newSuma(EXP1,EXP2);  EXP= newSuma(EXP1,EXP2);  EXP= ID;
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= ENTERONO EXP;  EXP:= EXPI + EXPI + EXPI = EXPI	EXP= ID; ID= id_(destino); LCASOS=newRetorno(EXP); EXP= ID; ID= id_(result); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderFemenina; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; ITPO=Entero; INSTRUCCION=newIF(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderMasculino; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newBoulif(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= primitivo(0); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=ackermann; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=m; TIPO=Entero; PARAMETRO=newDecParam(TIPO,ID); ID=m; TIPO=Entero; INSTRUCCION=newIf(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP= newSuma(EXP1,EXP2); EXP= ID; ID= id_ (n); EXP= primitivo(1); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetorno(EXP); EXP=newLlamadalnstr(ID,PARAMETRO); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= ENTERO EXP;  EXP:= ENTERO EXP;  EXP:= EXP != HEXP2  EXP:= EXP!= EXP! + EXP2  EXP:= ID  ID:= n  EXP:= I  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)	EXP= ID;  ID= id_(destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID= id_(result);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES ID=hofstaderFemenina;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newFit_(LSUBIF, POSELSE);  INSTRUCCION=newFit_(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES ID-hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecParam(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES=newDecFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES ID=ackermann;  TIPO=Entero;  PARAMETRO=newDec Param(TIPO,ID);  ID=m;  TIPO=Entero;  PARAMETRO=newDec Param(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newIf_(LSUBIF, POSELSE);  INSTRUCCION=newSubIf_(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= newSuma(EXP1,EXP2);  EXP= ID;  ID= id_(n);  EXP= primitivo(1);  INSTRUCCION=newSubIf_(EXP, INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP=newLlamadaInstr(ID,PARAMETRO);
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }	INTERPRETABLE IN
INSTRUCCION:= RETORNO EXP;  EXP:= ID  D:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  DD:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  DD:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  DD:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP1 + EXP2  EXP:= ID  DD:= n  EXP:= ID  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP1 + EXP2  EXP:= ID (PARAMETROS)  GLOBALES:= TIPO ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= SUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE	EXP= ID,   ID= id_(destino);   LCASOS=newRetomo(EXP);   EXP= ID;   ID= id_(result);   ID= id_(result);   ID= id_(result);   ID= id_(result);   ID= not id_(res
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= m  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= EXP! + EXP2  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= ID  ID:= n  EXP:= I  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= n  EXP:= ID  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)	EXP=ID; ID=id (destino); LCASOS=newRetomo(EXP); EXP-ID; ID-id_(result); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES ID=hofstaderFemenina; TIPO=Entero; PARAMETRO=newDee Param(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newF(LSUBIF, POSELSE); INSTRUCCION=newFubf(EXP, INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderMasculino; TIPO=Entero; PARAMETRO=newDeeParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ID=ackernann; TIPO=Entero; PARAMETRO=newDeeParam(TIPO,ID); ID=n; TIPO=Entero; PARAMETRO=newDeeParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP= newSuma(EXP); EXP= newSuma(EXP); EXP= newSuma(EXP); EXP= newSuma(EXP); EXP= newLlamadalnstr(ID,PARAMETRO); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP=newLlamadalnstr(ID,PARAMETRO); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); LCASOS=newRetomo(EXP); EXP=newLlamadalnstr(ID,PARAMETRO); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); EXP=newLlamadalnstr(ID,PARAMETRO);
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:=LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP1 + EXP2  EXP:= ID  ID:= n  EXP:= EXP:= EXP1 + EXP2  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= SUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }	EXP=1D; ID=id_(destino); ICASOS=newRetomo(EXP); EXP=1D; ID=id_(result); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderFemenina; TIPO=Entero; PARAMETRO=newDee Param(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newFit(LSUBIF, POSELSE); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID=hofstaderMasculino; TIPO=Entero; PARAMETRO=newDeeParam(TIPO,ID); ID=n; TIPO=Entero; INSTRUCCION=newFit(LSUBIF, POSELSE); INSTRUCCION=newFit(LSUBIF, POSELSE); INSTRUCCION=newFit(LSUBIF, POSELSE); ICASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= primitivo(0); GLOBALES=newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONIS) ID=ackermann; TIPO=Entero; PARAMETRO=newDee Param(TIPO,ID); ID=m; TIPO=Entero; PARAMETRO=newDee Param(TIPO,ID); ID=m; TIPO=Entero; INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= newSuma(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= newSuma(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= newSuma(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP= newFulamadalnstr(ID, PARAMETRO); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); ICASOS=newRetomo(EXP); EXP=newLlamadalnstr(ID, PARAMETRO); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION=newSubIF(EXP, INSTRUCCIONES); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); EXP=newLlamadalnstr(ID, PARAMETRO); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES);
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderfemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP!= ID  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (D:= n  EXP:= ID (D:= n  EXP:= ID (D:= n  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  EXP:= ID (PARAMETROS)  EXP:= ID (PARAMETROS)	EXP-ID: ID- id (destino); ICASOS -newRetorno(EXP); EXP- ID: ID- id (result); GLOBALES-newDeeFuncion(TIPO,ID)-PARAMETROS,INSTRUCCIONES ID-hofstaderFemenina; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; ITPO-Entero; NSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES) ID-hofstaderMasculino; TIPO-Fntero; PARAMETRO-newPeeParam(TIPO,ID); ID-n; ITPO-Fntero; PARAMETRO-newPeeParam(TIPO,ID); ID-n; ITPO-Fntero; PARAMETRO-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(1); ID-n; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(1); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- printityo(1); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- newSuma(EXP1,EXP2); EXP- ID; ID- id_(n); EXP- printityo(1); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- newSuma(EXP1,EXP2); EXP- ID; ID- id_(n); EXP- printityo(1); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetorno(EXP); EXP- newLamadalnstr(ID,PARAMETRO); INSTRUCCION-newIF(LSUBIF, POSELSE); INSTRUCCION-newIF(LSUBIF, POSELS
INSTRUCCION:= RETORNO EXP; EXP::- ID ID::- result GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES } ID::- hosatader/ememina TIPO::- Entero PARAMETRO::- TIPO ID [] ID::- n TIPO::- Entero INSTRUCCION::- LSUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } INSTRUCCION::- ESUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } ID::- hosataderMasculino TIPO::- Entero PARAMETRO::- TIPO ID [] ID::- n TIPO::- Entero INSTRUCCION::- LSUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } INSTRUCCION::- LSUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } INSTRUCCION::- ESUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } ID::- ackerman TIPO::- Entero PARAMETRO::- TIPO ID [] ID:- m TIPO::- Entero PARAMETRO:- TIPO ID [] ID:- m TIPO::- Entero PARAMETRO:- TIPO ID [] ID:- n TIPO::- Entero INSTRUCCION::- LSUBIF POSELSE LSUBIF::- ELSE? IF (EXP) { INSTRUCCIONES } INSTRUCCION:- RETORNO EXP; EXP::- ID ID:- n ID:- n ID:- n EXP::- ID ID:- n ID:	EXP-ID; ID-ai_d(dsstino); ICASOS=newRetomo(EXP); EXP= ID; ID-id_(result); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES ID-hofstaderFemenina; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; INSTRUCCION-newFiceLSUBIF, POSELSE); INSTRUCCION-newFiceDifferon(EXP); EXP-primitivo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ID-hofstaderMasculino; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); IO-NOSELSE; INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetomo(EXP); EXP-primitivo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ICASOS-newRetomo(LPP); EXP-primitivo(0); GLOBALES-newDeeFuncion(TIPO,ID,PARAMETROS,INSTRUCCIONES); ID-ackermann; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-m; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-m; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-m; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; PARAMETRO-newDeeParam(TIPO,ID); ID-n; TIPO-Entero; INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); ICASOS-newRetomo(EXP); EXP- newSumat(EXP, INSTRUCCIONES); ICASOS-newRetomo(EXP); EXP- newBumat(EXP, INSTRUCCIONES); ICASOS-newRetomo(EXP); EXP- newBumat(EXP, INSTRUCCIONES); ICASOS-newRetomo(EXP); EXP- newBumat(EXP, INSTRUCCIONES); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); EXP-newLlamadalnstr(ID, PARAMETRO); INSTRUCCION-newSubIF(EXP, INSTRUCCIONES); EXP-newLlamadalnstr(ID, PARAMETR
INSTRUCCION:= RETORNO EXP;  EXP:= ID  D:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES } ID:= hofstaderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  D:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSI? IF (EXP) { INSTRUCCIONES } INSTRUCCIONES } INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES } ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES } INSTRUCCIONES } INSTRUCCIONES } INSTRUCCIONES   INSTRUCC	EXP= ID; ID= id_(destino); ILCASOS -newRetomo(EXP); EXP ID; ID= id_(result); ID= id_(result
INSTRUCCION:= RETORNO EXP;  EXP:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderf-emenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF := ELSE? IF (EXP) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  RNSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= n ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID  EXP:= ID  ID:= n  EXP:= ID  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= ID  ID:= n  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  ID:= main  TIPO:= Entero  EXP:= ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)	EXP=ID;  ID id. (destino);  LCASOS=newRetorno(EXP);  EXP= ID;  ID id. (result);  GLOBALES=newDecf-oncion(TIPO,ID,PARAMETROS,INSTRUCCIONIS  ID=hofstaderf-emenina;  TIPO=Intero;  PARAMETRO=newDecf-param(TIPO,ID);  ID=n;  TIPO=Entero;  INSTRUCCION=newBe(LSUBIF, POSELSE);  INSTRUCCION=newBe(LSUBIF, POSELSE);  INSTRUCCION=newBe(LSUBIF, POSELSE);  ICASOS newRetorno(EXP);  IXP= primitivo(0);  GLOBALES=newDecf-uncion(TIPO,ID,PARAMETROS,INSTRUCCIONES  ID hofstaderMasculino;  TIPO=Entero;  PARAMETRO=newDecf-param(TIPO,ID);  ID n;  TIPO=Entero;  INSTRUCCION=newBe(LSUBIF, POSELSE);  INSTRUCCION=newBe(LSUBIF, POSELSE);  INSTRUCCION-newBe(LSUBIF, POSELSE);  INSTRUCCION-newBe(LSUBIF, POSELSE);  INSTRUCCION-newDecf-param(TIPO,ID);  GLOBALES newDecf-uncion(TIPO,ID,PARAMETROS,INSTRUCCIONES);  LCASOS=newRetorno(EXP);  EXP= primitivo(0);  GLOBALES newDecf-uncion(TIPO,ID,RAMAMETROS,INSTRUCCIONES);  ID=nic  TIPO=Entero;  PARAMETRO newDecf-param(TIPO,ID);  ID=nic  TIPO=Entero;  INSTRUCCION=newBe(LSUBIF, POSELSE);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetomo(EXP);  EXP=newLlamadalnstr(ID,PARAMETRO);  RNSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  LCASOS=newRetomo(EXP);  INSTRUCCION newSubIF(EXP, INSTRUCCIONES);  ICASOS=newRetomo(EXP);  INSTRUCCION newSubIF(EXP, INSTRUCCIONES);  EXP=newLlamadalnstr(ID,PARAMETRO);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  EXP=newLlamadalnstr(ID,PARAMETRO);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  EXP=newLlamadalnstr(ID,PARAMETRO);  INSTRUCCION=newSubIF(EXP, INSTRUCCIONES);  EXP=newLlamadalnstr(ID,PARAMETRO);  EXP=newLlamadalnstr(ID,PARAMETRO);  EXP=newLlamadalnstr
INSTRUCCION:= RETORNO EXP;  EXP;= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= holstaderf-emenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LUBIF POSELSE  LSUBIF:= ELSIP! F (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= holstaderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= selement  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSIP IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= EXP:= ID  ID:= n  EXP:= ID (PARAMETROS)  INSTRUCCION:= RETORNO EXP;  EXP:= ELSIP IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= ID (PARAMETROS)  INSTRUCCION:= SUBIF POSELSE  LSUBIF:= ELSIP IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)  INSTRUCCION:= LUBIF POSELSE  LSUBIF:= ELSIP IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMETROS)	FEXP   ID;   ID = id (destino);
INSTRUCCION:= RETORNO EXP;  EXP:::::::::::::::::::::::::::::::::::	INPO-Edictory  ID-edictory  ICASOS-enewRestorno(EXP);  EXP ID;  ID-edictory  ID-bofstader+emenina;  TIPO-Entero;  PARAMETRO newbe-param(TIPO,ID);  ID-n;  TIPO-Entero;  INSTRUCCION newlf(LSUBH, POSHLSE);  ID hofstader Wasculino;  TIPO-Entero;  ID hofstader Wasculino;  TIPO-Entero;  PARAMETRO-newDee Param(TIPO,ID);  ID n;  TIPO-Entero;  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newlf(LSUBH, POSHLSE);  ID-ackermann;  TIPO-Entero;  PARAMETRO-newDeeParam(TIPO,ID);  ID n;  TIPO-Entero;  PARAMETRO-newDeeParam(TIPO,ID);  ID n;  TIPO-Entero;  PARAMETRO-newDeeParam(TIPO,ID);  ID n;  TIPO-Entero;  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newlf(LSUBH, POSHLSE);  INSTRUCCION-newSubH(EXP, INSTRUCCIONES);  LCASOS-newRestomo(EXP);  EXP newSuma(EXP), EXTRUCCIONES);  LCASOS-newRestomo(EXP);  EXP newLamadalastr(ID, PARAMETRO);  INSTRUCCION-newSubH(EXP, INSTRUCCIONES);  LCASOS newRestomo(EXP);  EXP-newLamadalastr(ID, PARAMETRO);  INSTRUCCION-newSubH(EXP, INSTRUCCIONES);  INSTRUCCION-newSubH(EXP, INSTRUCCIONES);  INSTRUCCION-newSubH(EXP, INSTRUCCIONES);  EXP-newLamadalastr(ID, PARAMETRO);  INSTR
INSTRUCCION:= RETORNO EXP;  LEXP.:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstuderFemenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= SUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstuderMasculino  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSE? IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP;  EXP:= LXP] + EXP2  EXP:= ID [ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP;  EXP:= LXP] + EXP2  EXP:= ID [ID:= n  LXP:= ID:= ID:= ID:= ID:= ID:= ID:= ID:= ID	EXP = ID:
INSTRUCCION:= RETORNO EXP;  EXP.:= ID  ID:= result  GLOBALES:= TIPO ID (PARAMITROS) { INSTRUCCIONES }  ID:= hofstaderf emenina  TIPO:= Entero  PARAMETRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= LSUBIF POSILSE  LSUBIF:= LISP! IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP; EXP:= 0  GLOBALES:= TIPO ID (PARAMETROS) { INSTRUCCIONES }  ID:= hofstader/Masculino  TIPO:= Entero  PARAMITRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMITRO:= TIPO ID []  ID:= n  TIPO:= Entero  INSTRUCCION:= RETORNO EXP; EXP:= 0  GLOBALES:= TIPO ID (PARAMITROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP; EXP:= 0  GLOBALES:= TIPO ID (PARAMITROS) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP; EXP:= 0  GLOBALES:= TIPO ID (PARAMITROS) { INSTRUCCIONES }  ID:= ackermann  TIPO:= Entero  PARAMITRO:= TIPO ID []  ID:= n  TIPO:= Entero  PARAMITRO:= TIPO ID []  ID:= n  TIPO:= Entero  NSTRUCCION:= RETORNO EXP; EXP:= LSP! IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP; EXP:= LSP! IT (EXP) { INSTRUCCIONES }  INSTRUCCION:= RETORNO EXP; EXP:= ID [ID:= n  EXP:= ID (PARAMITROS)  INSTRUCCION:= RETORNO EXP; EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  ID:: main  TIPO:= Entero  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTRUCCION:= LSUBIF POSELSE  LSUBIF:= ELSP! IF (EXP) { INSTRUCCIONES }  EXP:= ID (PARAMITROS)  INSTR	EXP   D:  ID = id (destino);  LCASOS=newRecomo(EXP);  EXP = ID:  ID = id (destino);  GLOBALES=newDeeFuncion(IPO_ID_PARAMETROS_INSTRUCCIONES)  ID=hofstadefremenians:  IIPO=Enisto;  PARAMETRO-newDeeParam(IPO_ID);  ID = ii  IIRO=Enisto;  INSTRUCCION=newIf(LSUBIF, POSELSE);  INSTRUCCION=newSubif(EXP, INSTRUCCIONES);  LCASOS=newRecomo(EXP);  EXP = primitivo(0);  GLOBALES=newDeerFuncion(ITPO_ID_PARAMETROS_INSTRUCCIONES)  ID=hofstaderMasculio;  IIPO=Enisto;  PARAMETRO=newDeeParam(ITPO_ID);  ID=ii;  IIPO=Finitro;  PARAMETRO=newBef [SUBIF, POSELSE);  INSTRUCCION=newSubif(EXP, INSTRUCCIONES);  LCASOS=newRecomo(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(ITPO_ID_PARAMETROS_INSTRUCCIONES);  LCASOS=newRecomo(EXP);  EXP= primitivo(0);  GLOBALES=newDeeFuncion(ITPO_ID_PARAMETROS_INSTRUCCIONTS);  INSTRUCCION-newSubif(EXP, INSTRUCCIONTS);  ID=ne, triplo=Finitro;  PARAMETRO=newDee Param(TIPO_ID);  ID=ne, triplo=Finitro;  PARAMETRO=newDee Param(TIPO_ID);  ID=ne, triplo=Finitro;  PARAMETRO=newDee Param(TIPO_ID);  ID=ne, triplo=Finitro;  INSTRUCCION-newIf(ISUBIF, POSELSE);  INSTRUCCION-newIf(ISUBIF, POSELSE);  INSTRUCCION-newIf(ISUBIF, POSELSE);  INSTRUCCION-newIf(ISUBIF, POSELSE);  INSTRUCCION-newIf(ISUBIF, POSELSE);  INSTRUCCION newIf(ISUBIF, POSELSE);  INSTRUCCION newSubif(EXP, INSTRUCCIONES);  EXP newLamadalnstr(ID_PARAMETRO);  EXP newLamadalnstr(ID_PARAMETRO);  EXP newLamadalnstr(ID_PARAMETRO);  INSTRUCCION newSubif(EXP, INSTRUCCIONES);  EXP newLamadalnstr(I