```
void Main()
{
  return;
}
```

```
int i;

int devuelveGlobal()
{
  return i;
}
void setGlobal(int x)
{
  i = x;
  return;
}

void Main()
{
  int i;
  i = 15;

  WRITE "Local:";
  WRITE i;
  WRITE "";
  WRITE "Global antes de modificar:";
  WRITE devuelveGlobal();
  setGlobal(48);
  WRITE "Global tras modificar:";
  WRITE devuelveGlobal();
  WRITE "Local:";
  WRITE i;



  return;
}
```

```
#este programa ademas pone "a prueba" el trabajar con estructuras
#de datos grandes
int Main()
{
  struct dia_t {
    struct hora_t {
      int minuto[60];
      double segundo[3600];
    } hora;
  } dia;

  dia.hora.minuto[30] = 1;
  dia.hora.segundo[1234] = 4321;

  WRITE dia.hora.minuto[30];
  WRITE dia.hora.segundo[1234];


  return 0;
}
```

```
int Main(){

  double foo[100];
  i=1;
  while( i < 100 ){
    foo[i] = (i*i);
    i = i+1;
  }

  i=1;
  while( i < 100 ){
    WRITE foo[i];
    i = i+1;
  }

  return 1;
}
```

```
void Main()
{
  struct structConArray_t {
    double arr[12];
    int a[12];
  } structConArray;

  int i;
  i=0;
  while( i < 12){
    structConArray.a[i] = i;
    structConArray.arr[i] = 3.1415;
    i = i+1;
  }

  i=0;
  while( i < 10 ){
    WRITE structConArray.a[i];
    WRITE structConArray.arr[i] * structConArray.a[i];
    WRITE "";
    i = i+1;
  }


  return;
}
```

```
int Main()
{
  struct foobar{
    double secs;
    int minutos;
  } TPO[2];

  TPO[0].minutos = 60;
  TPO[0].secs = 3.141592;

  TPO[1].minutos = 6;
  TPO[1].secs = 0.141592;

  WRITE TPO[0].minutos;
  WRITE TPO[1].secs;

  return 0;
}
```

```
void Main()
{
  int entero;
  double real;

  WRITE "Introducir entero:";
  READ entero;
  WRITE "Introducir real:";
  READ real;

  WRITE "";

  WRITE "Entero introducido:";
  WRITE entero;
  WRITE "Real introducido:";
  WRITE real;

  return;
}
```

```
void Main()
{
  int a;
  int b;

  a = 1;
  b = 0;

 if( a AND b )
    WRITE "’a’ y ’b’ son ciertos";
  else
    WRITE "’a’ o ’b’ son falsos...";

  if( a OR b )
    WRITE "o bien ’a’ o ’b’ es cierto";
  else
    WRITE "’a’ y ’b’ son ambos falsos...";

  if( NOT a )
    WRITE "’a’ es falso";
  else
    WRITE "’a’ es cierto";

  return;
}
```

```
int Main()
{
  int i;
  i = 0;
  while( i < 10 ){
    WRITE i;
    i = i+1;
  }
  return 0;
}
```

```
int Main()
{
  int i;
  double d;

  READ i;
  READ d;

  WRITE (double)i;
  WRITE (int)d;

  return 0;
}
```

```
void Foo(int i, double d)
{
  WRITE "VERSION BIPARAMETRO";
  int x;
  int y;

  WRITE i;
  WRITE d;

  return;
}
void Foo(int i){
  WRITE "VERSION MONOPARAMETRO";
  WRITE i;

  return;
}

int Main()
{
  int param;
  param = 14;
  Foo(14, 14.23);
  Foo(41);

  return 0;
}
```

```
int Factorial(int x)
{
  int tmp;
  WRITE x;
  if( x > 1) {
    tmp = (x*Factorial(x-1));
  }
  else {
    tmp = 1;
  }

  return tmp;
}

int Main()
{
 int x;
#READ x;
 x = 5;
 WRITE Factorial(x);
 return 0;
}
```