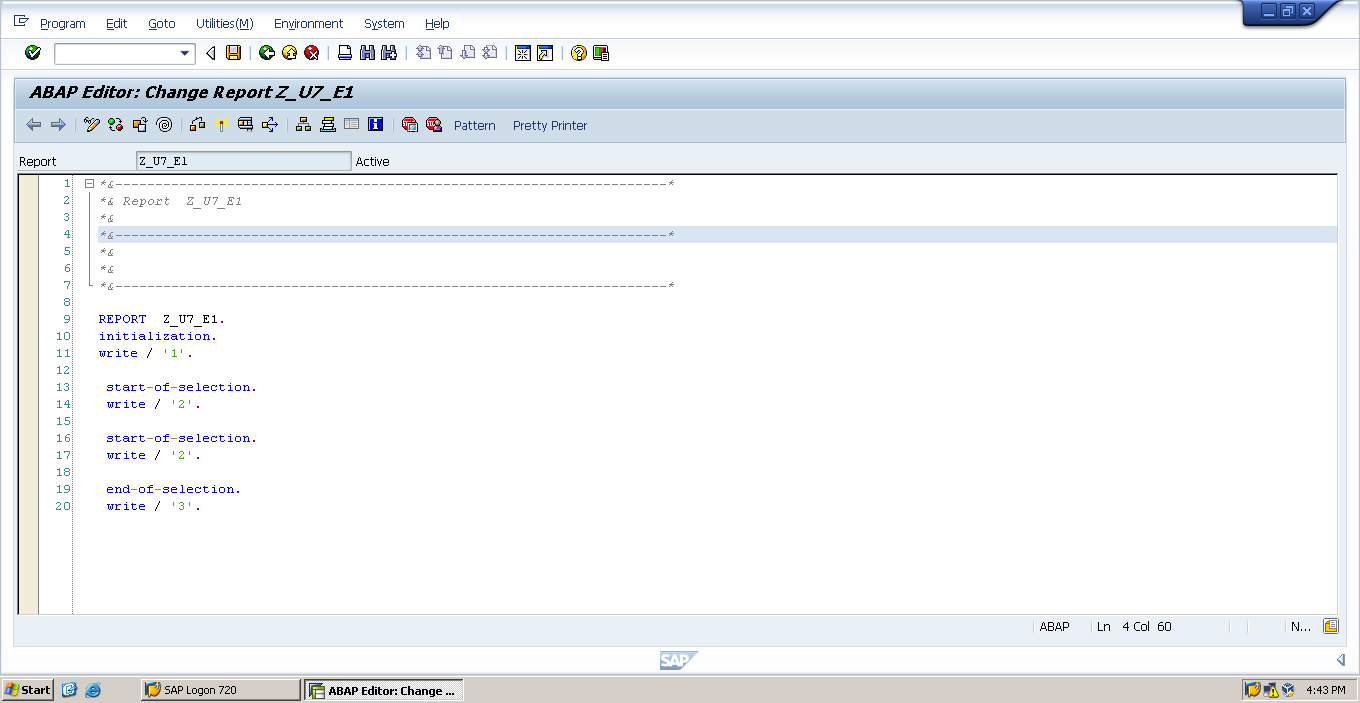
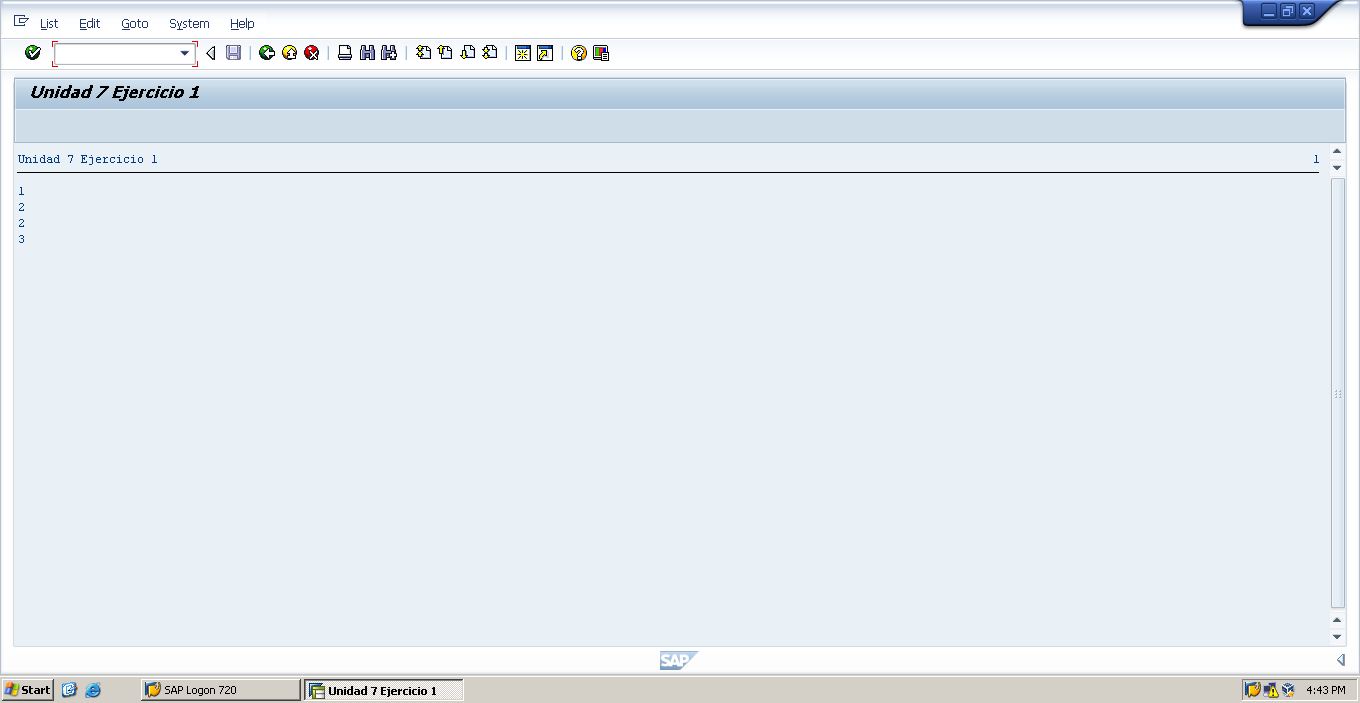
Ejercicio 1 Tarea 1

1. **¿Qué pasa si tengo dos eventos del mismo nombre en el mismo programa?**

Estuve probando en la máquina virtual y no ocurrió ningún error. También probé con los otros eventos y no hubo ningún problema. Supongo que es como si el evento continuara y/o sigue el debido orden.

1. **¿Cómo puedo saber la diferencia entre un nombre de evento y una declaración?**

Si ve una sola palabra seguida de un punto, generalmente es un evento. Más allá de eso, los eventos no tienen ninguna característica identificable aparte de sus nombres. Son simplemente palabras reservadas y para reconocer una debe memorizar sus nombres, así como memoriza las palabras clave en las declaraciones.[[1]](#footnote-0)

1. **¿Puedo ver el código del programa driver?**

Para ver un programa de controlador, comience desde dentro del editor de código fuente. Elija la ruta del menú Utilidades-> Ayuda activada. Elija el botón de opción Base de datos lógica y escriba una identificación de base de datos lógica (KDF , por ejemplo). Luego verá la estructura de la base de datos lógica. Elija la ruta del menú Ir a-> Programa de base de datos para ver el programa del controlador. [[2]](#footnote-1)

1. **¿Puedo elegir el driver?**

Puede elegir el programa del controlador completando los campos de la base de datos lógica en la pantalla de atributos del programa. Si no especifica uno, se asigna automáticamente un valor predeterminado.[[3]](#footnote-2)

1. **¿Puedo crear un driver?**

Sí, puede crea su propio programa de base de datos lógica, no puede cambiar la secuencia relativa de eventos o sus nombres, ni puede crear otros nuevos.[[4]](#footnote-3)

1. **¿Cuáles son las diferencias entre los programas driver disponibles?**

La vez que desea elegir el controlador es cuando desea que lea los datos de la base de datos y los suministre a su programa automáticamente. Las bases de datos lógicas predeterminadas no leen de la base de datos, solo desencadenan eventos.

1. **Si pongo una declaración write en un evento que viene antes start-of-selection, veo que funciona, pero solo cuando no hay una pantalla de selección. Si hay una pantalla de selección, no veo la salida. ¿Por qué?**

R / 3 utiliza algo llamado contexto de pantalla para aceptar entradas y mostrarle resultados. Puede pensar en el contexto de la pantalla como una sola pantalla. Cada contexto de pantalla se procesa en dos fases: la fase de diálogo y la fase de lista. La fase de diálogo siempre ocurre primero, seguida de la lista. La fase de diálogo se utiliza para mostrar los campos de entrada al usuario, como los generados por una declaración de parámetros. La fase de lista se utiliza para mostrar la salida de la declaración de escritura al usuario. Ambas fases usan el mismo contexto de pantalla, por lo que se sobrescriben entre sí. La fase de diálogo, sin embargo, hace una pausa para la entrada del usuario y, por lo tanto, se muestra antes de que se muestre la lista. Entonces, si emite una declaración de escritura en la inicialización evento y si se muestra la pantalla de selección, la pantalla de selección sobrescribe la salida de la escritura y esa salida se pierde. Luego elige Ejecutar en la pantalla de selección y, si se emiten declaraciones de escritura, la lista sobrescribe la pantalla de selección y verá el resultado. Sin embargo, si no tiene una pantalla de selección, las escrituras que ocurren en la inicialización aún se pueden ver.[[5]](#footnote-4)

1. **Averiguar si la variación write to de la sentencia write dispara o no el evento top-of-page.**

No, no dispara el evento.

1. **¿En qué eventos no se debería usar stop y por qué?**

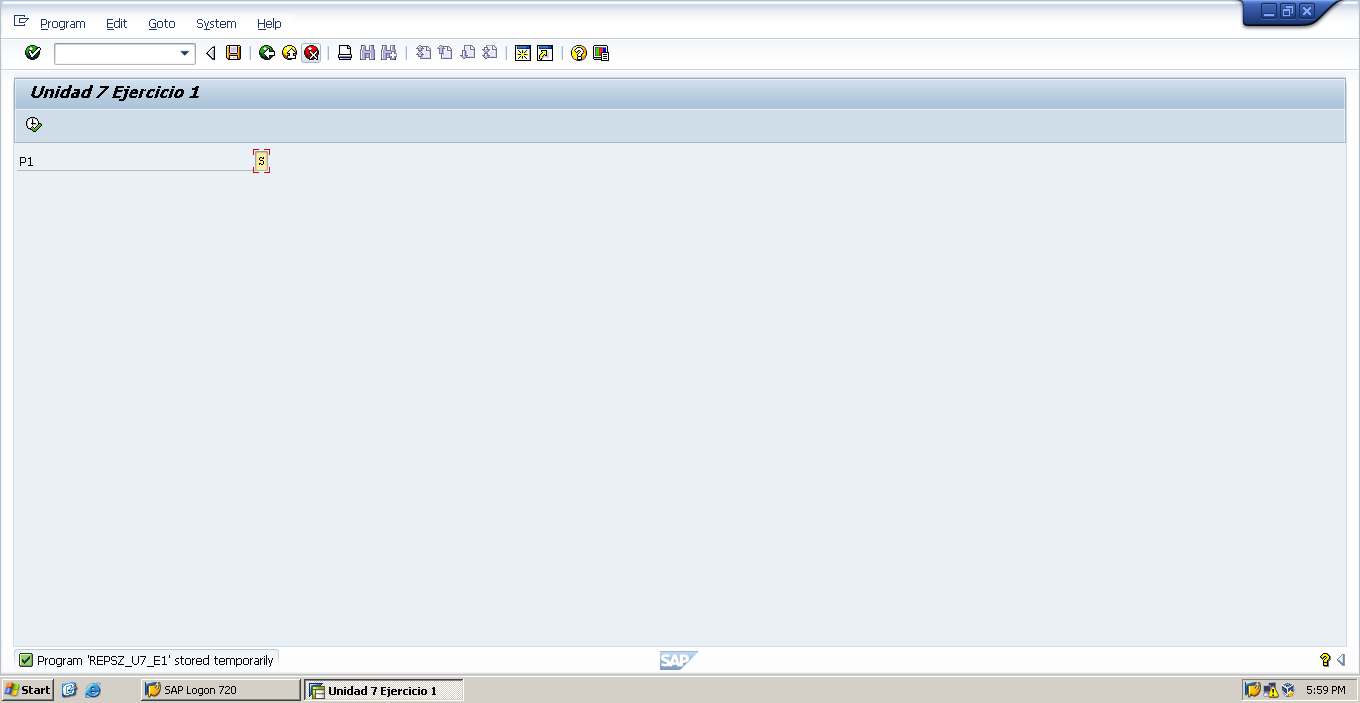
No use stop en los siguientes eventos: initialization, en selection-screen output, top-of-page, and end-of-page. Técnicamente, puede funcionar con top-of-page Y end-of-page si se abstiene de emitir declaraciones de escritura dentro del end-of-selectio. Esta recomendación es para evitar errores. Se podrían causar dumps, o se podría perder la salida de algunos procesos.[[6]](#footnote-5)

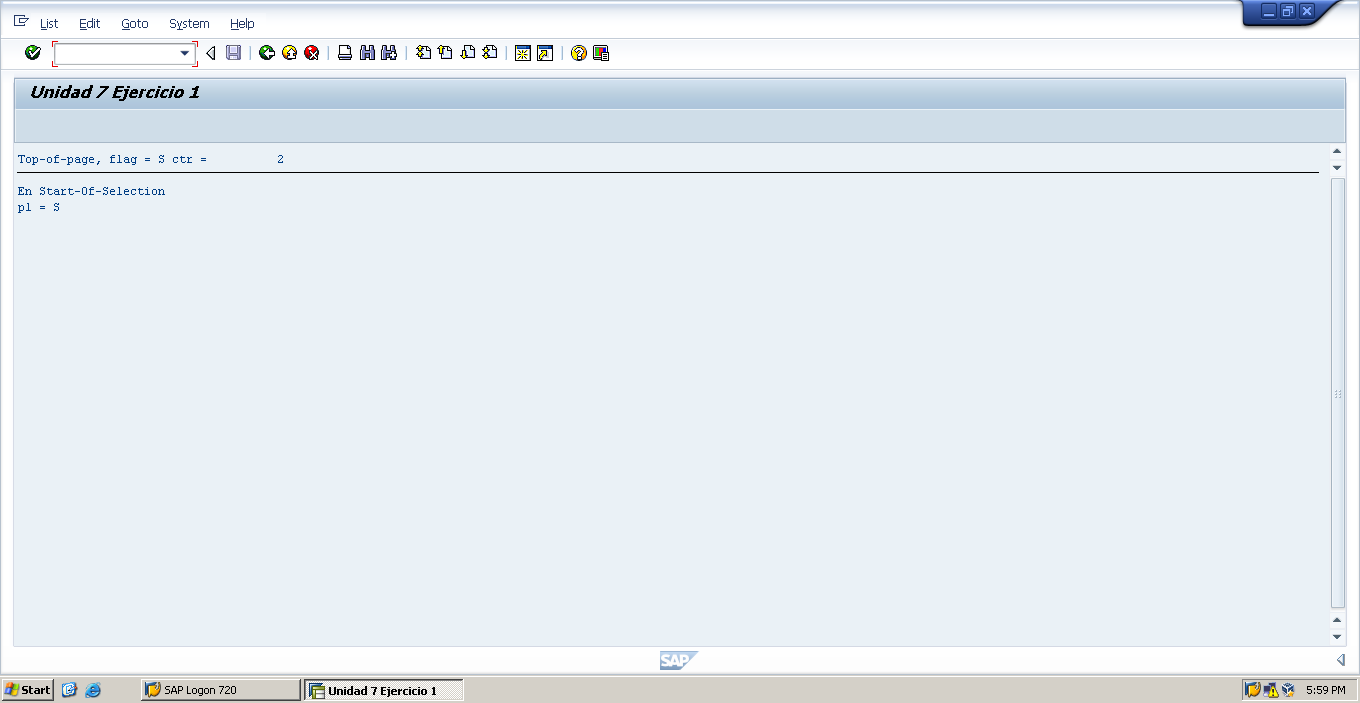
1. **¿A partir de qué momento es una variable conocida dentro de un programa?**

Para ser precisos, una variable solo se conoce dentro de un programa después del punto en el que se define. Por ejemplo, si define una variable en la línea 10, podrá acceder a ella en las líneas 11 y posteriores, pero no en las líneas 1-9. En el caso de una definición local, puede acceder a la versión global de la variable en cualquier punto antes de la definición local.

Ejercicio 2 Tarea 1

**Ejecute el programa que se muestra. Comente la instrucción parámetros y ejecútelo de nuevo. Describa la secuencia de eventos que se producen en cada caso**.



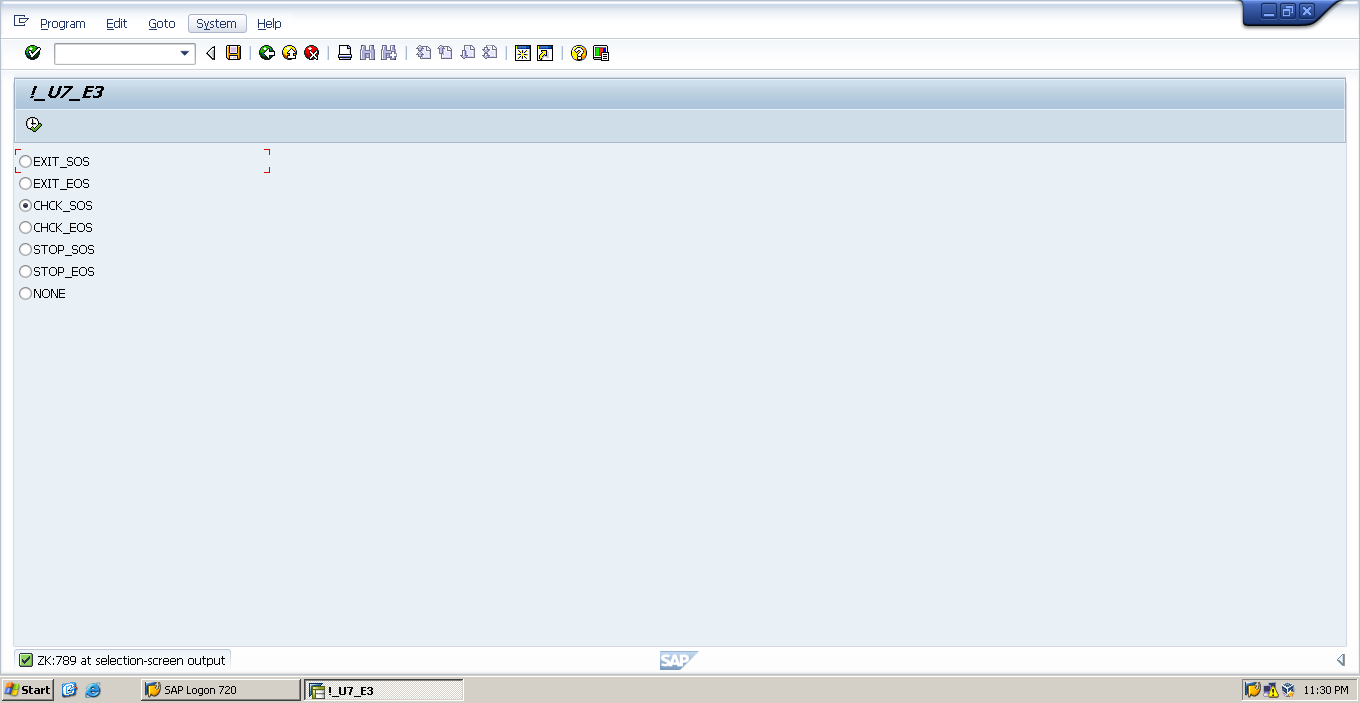
Luego de ejecutarlo por primera vez, pide el parámetro p1. En este caso elegí ingresarle el valor de S.

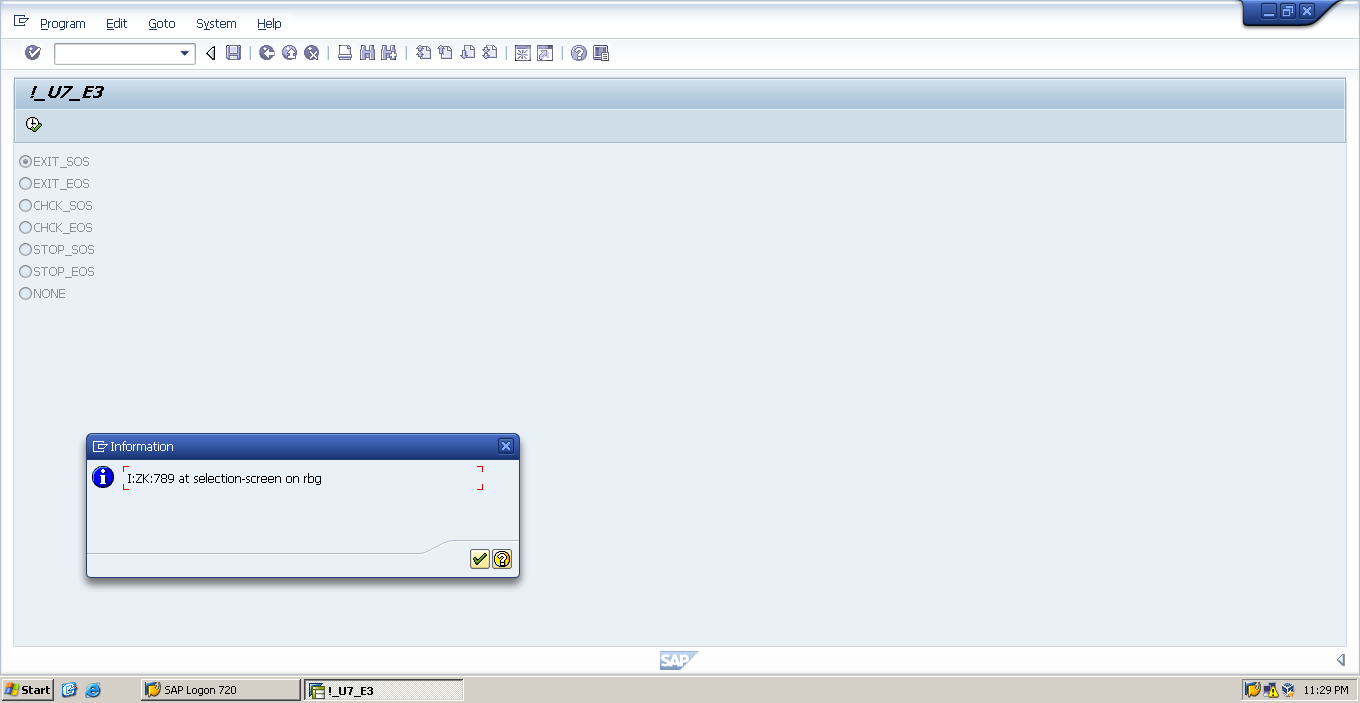
Ejecutándolo nuevamente, aparece el siguiente resultado:

Descripción de eventos:

La sección de initialization no se ejecuta porque no hay criterios de selección en el programa. Se ejecuta el evento de Start-Of-Selection. Cuando el programa llega a la primera instrucción de escritura, el control del programa pasa al evento de Top-of-page. Cuando se completa el evento de Top-of-page, el control se devuelve al evento de inicio de selección.

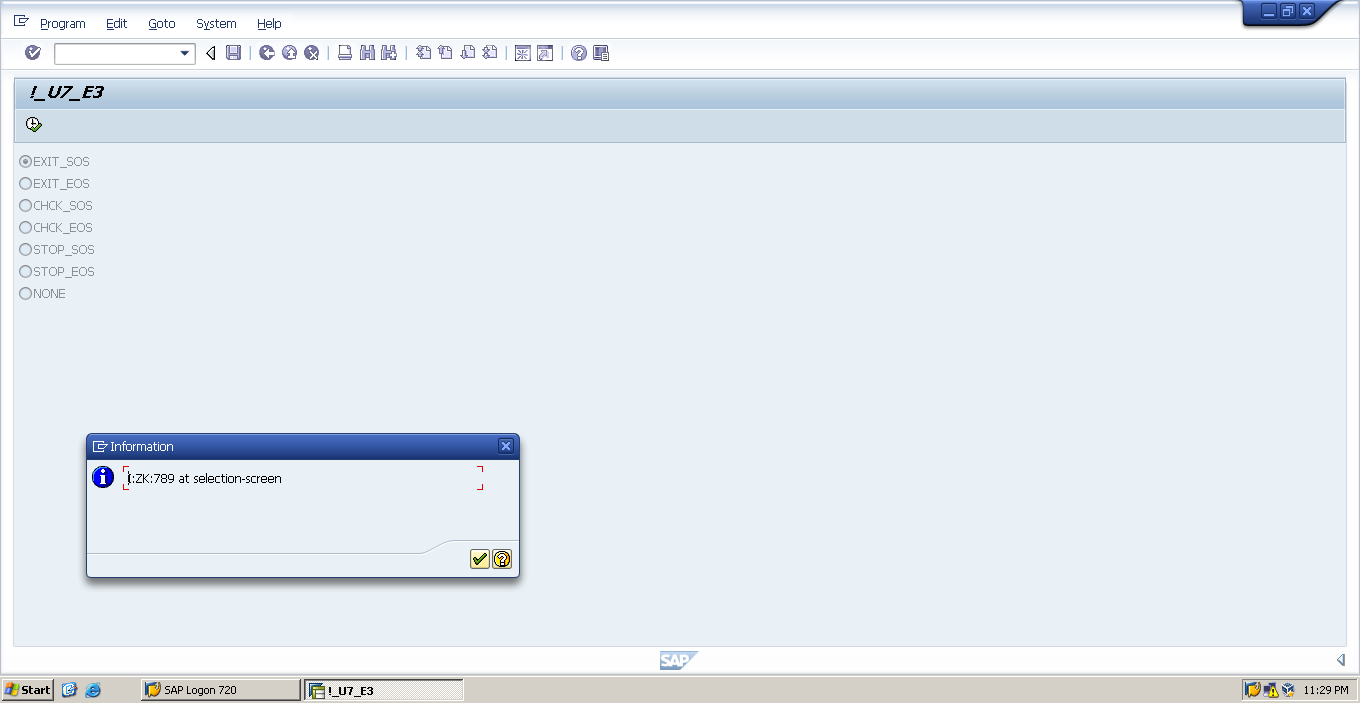
Ejercicio 3 Tarea 1

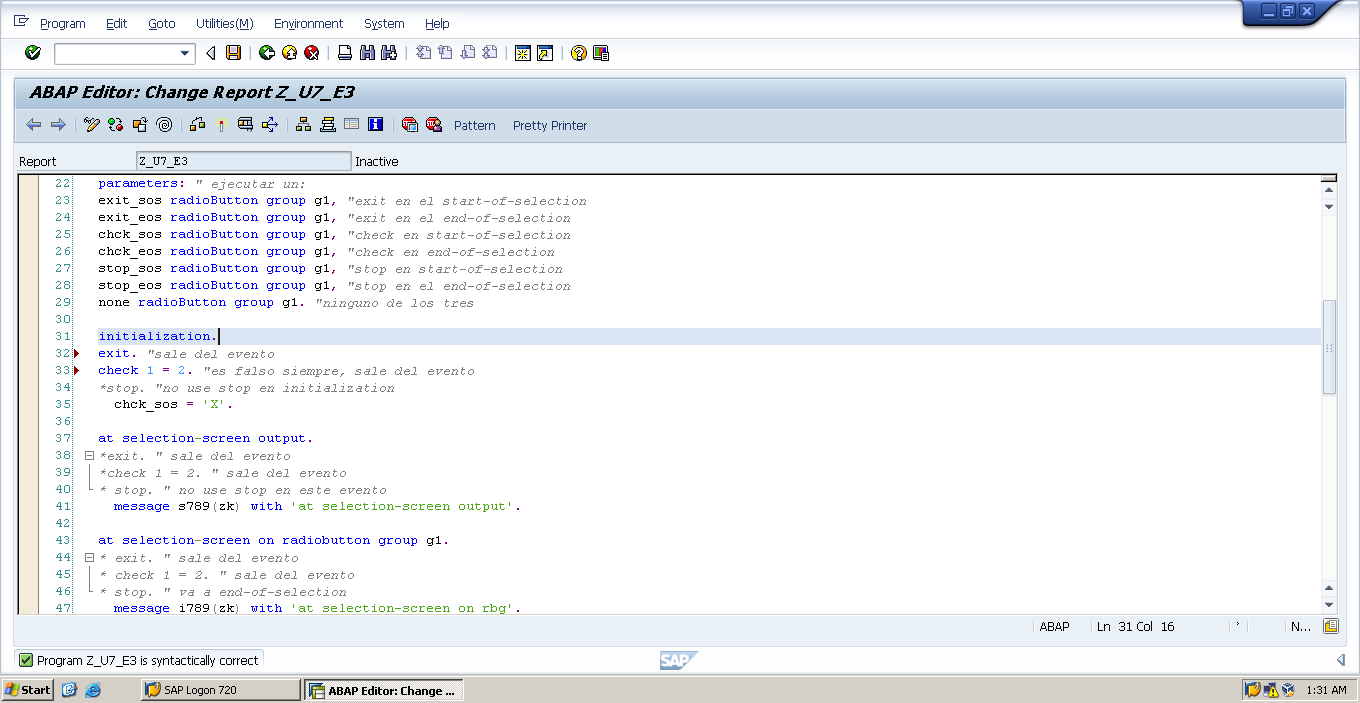
**Respecto del programa ztx0705 del material teórico, indique cuál es la salida que se obtiene (incluyendo los mensajes) en cada caso, al comentar y dejar luego de comentar cada una de las sentencias, tal cual lo pide el mencionado material teórico.**

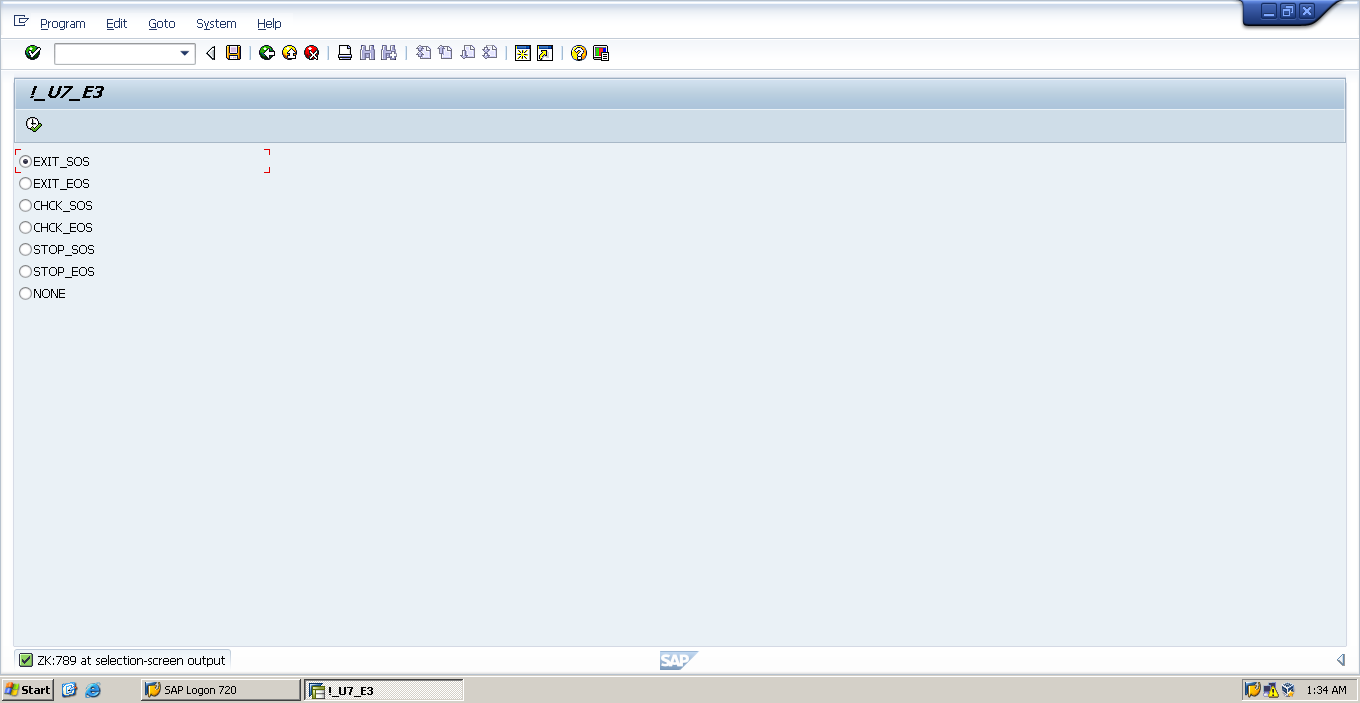
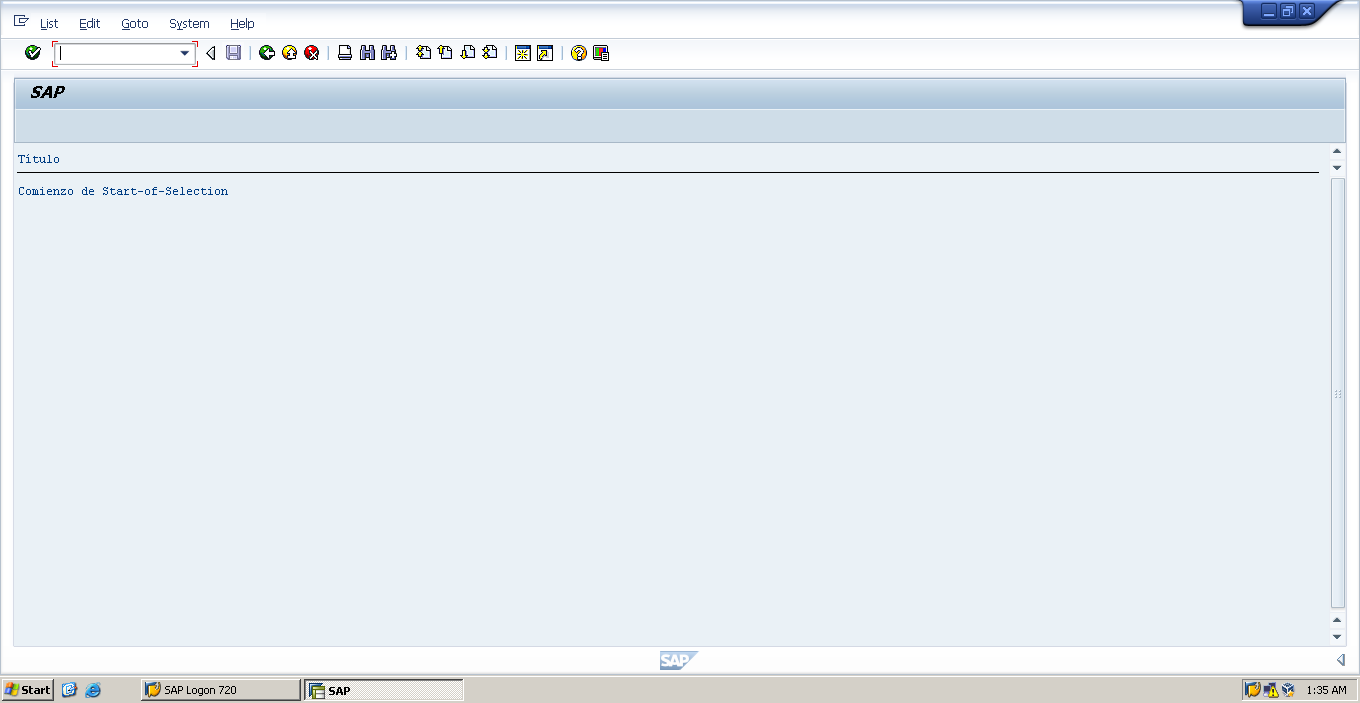
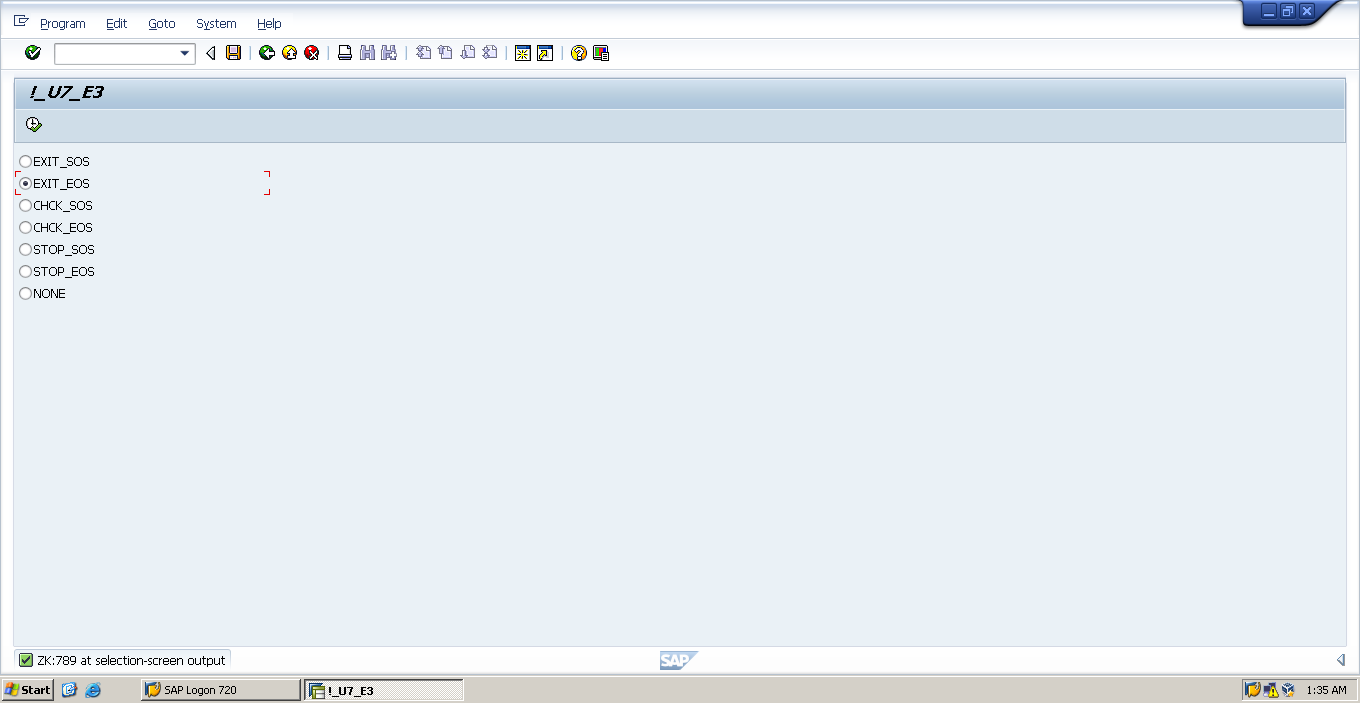
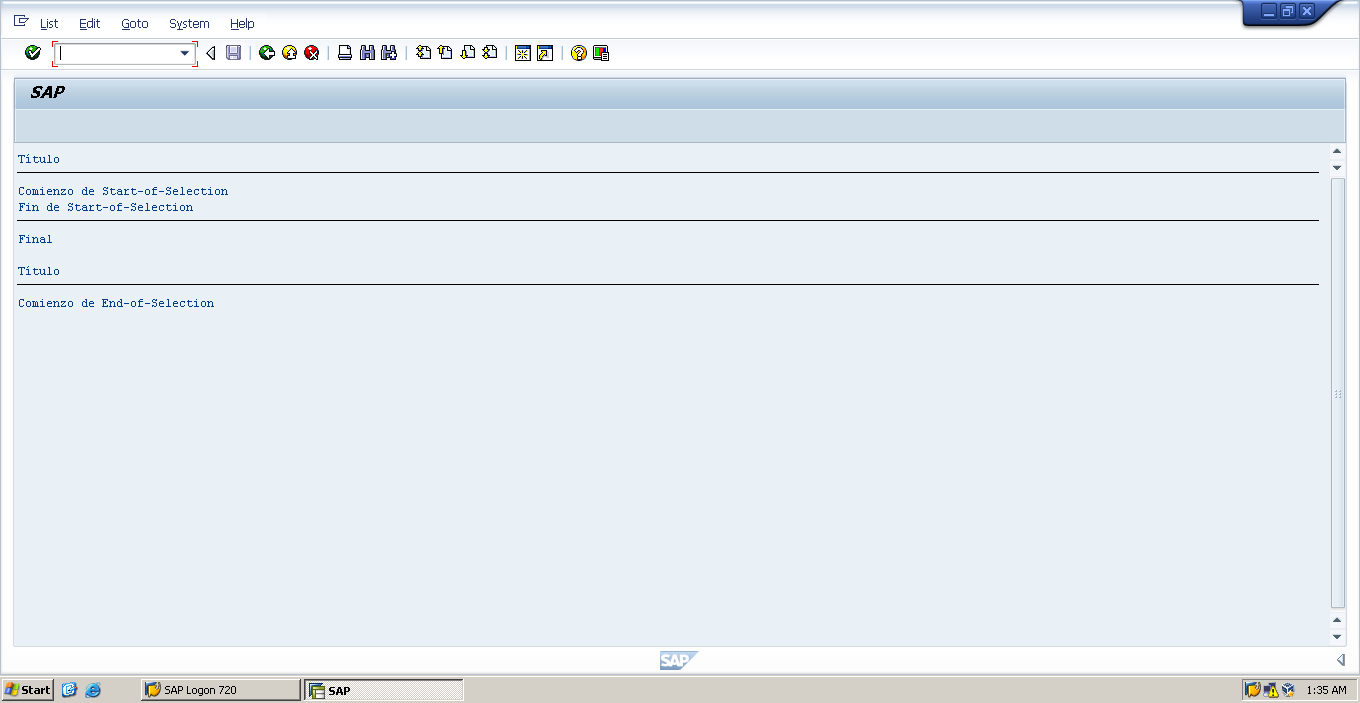


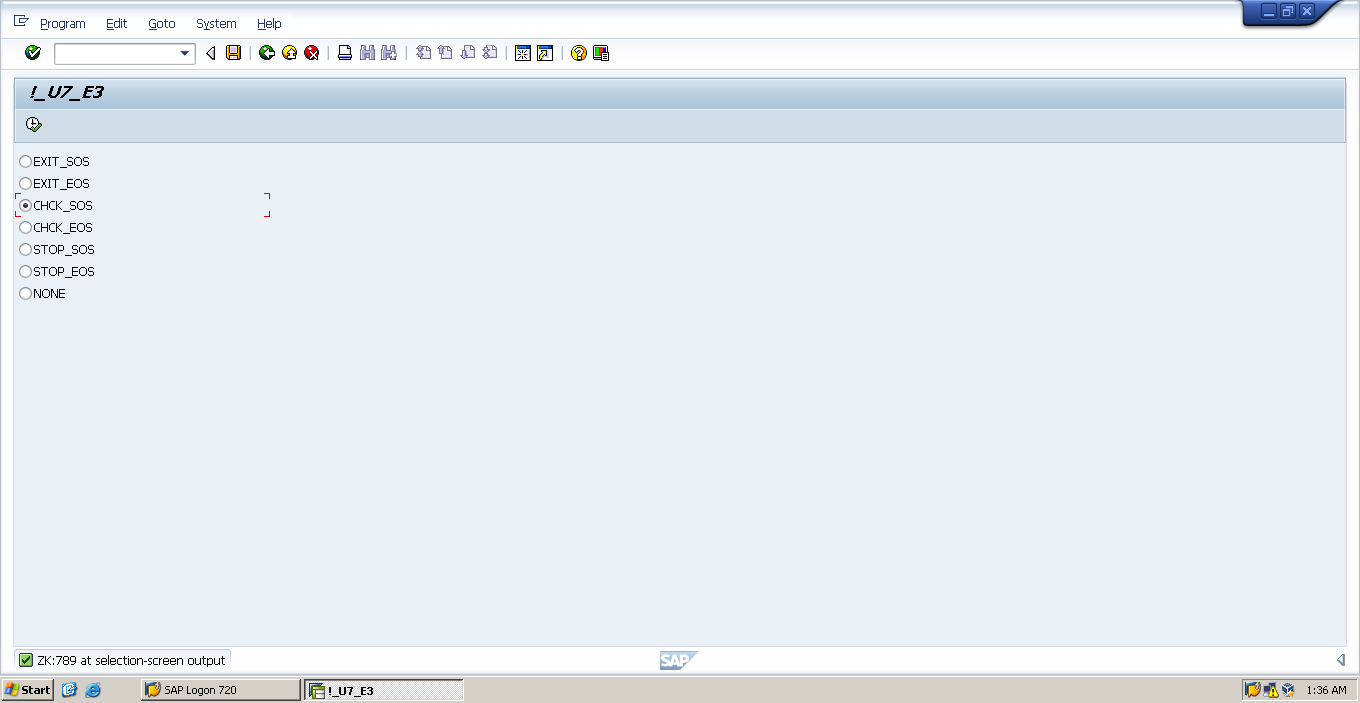
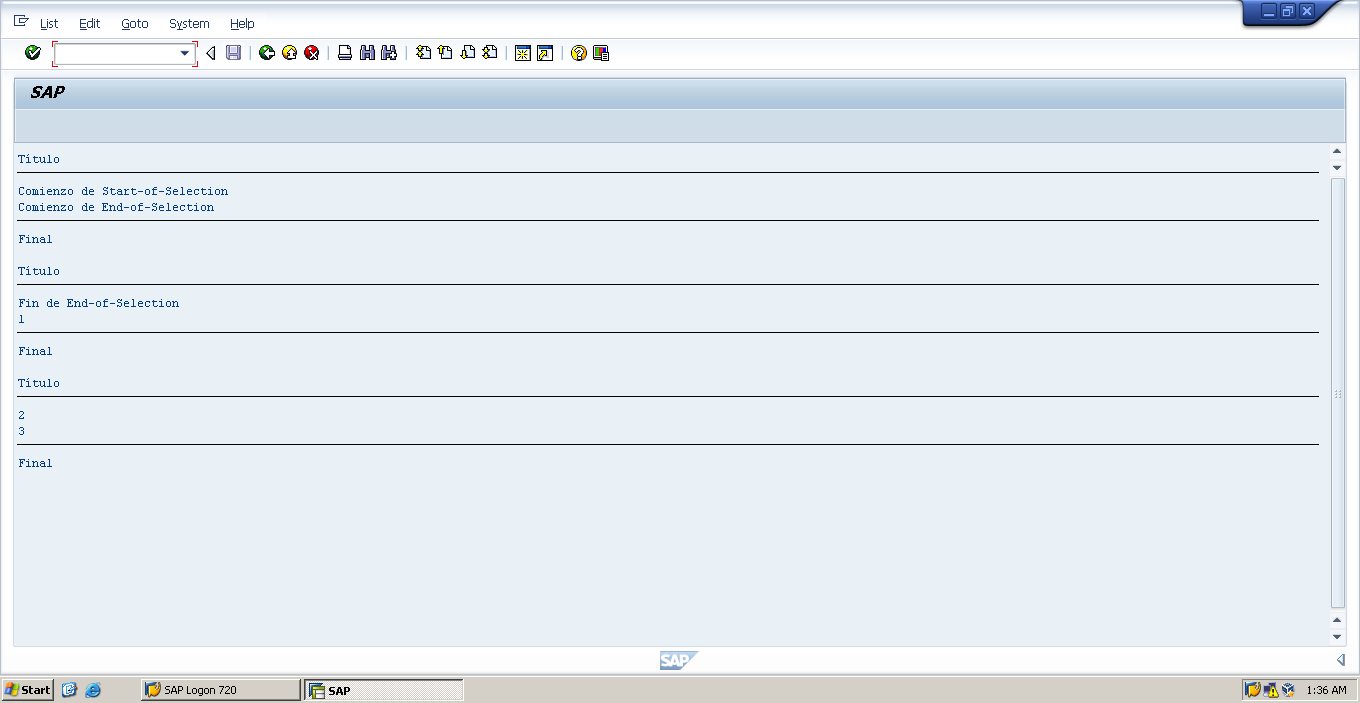
En cada caso se mostraron

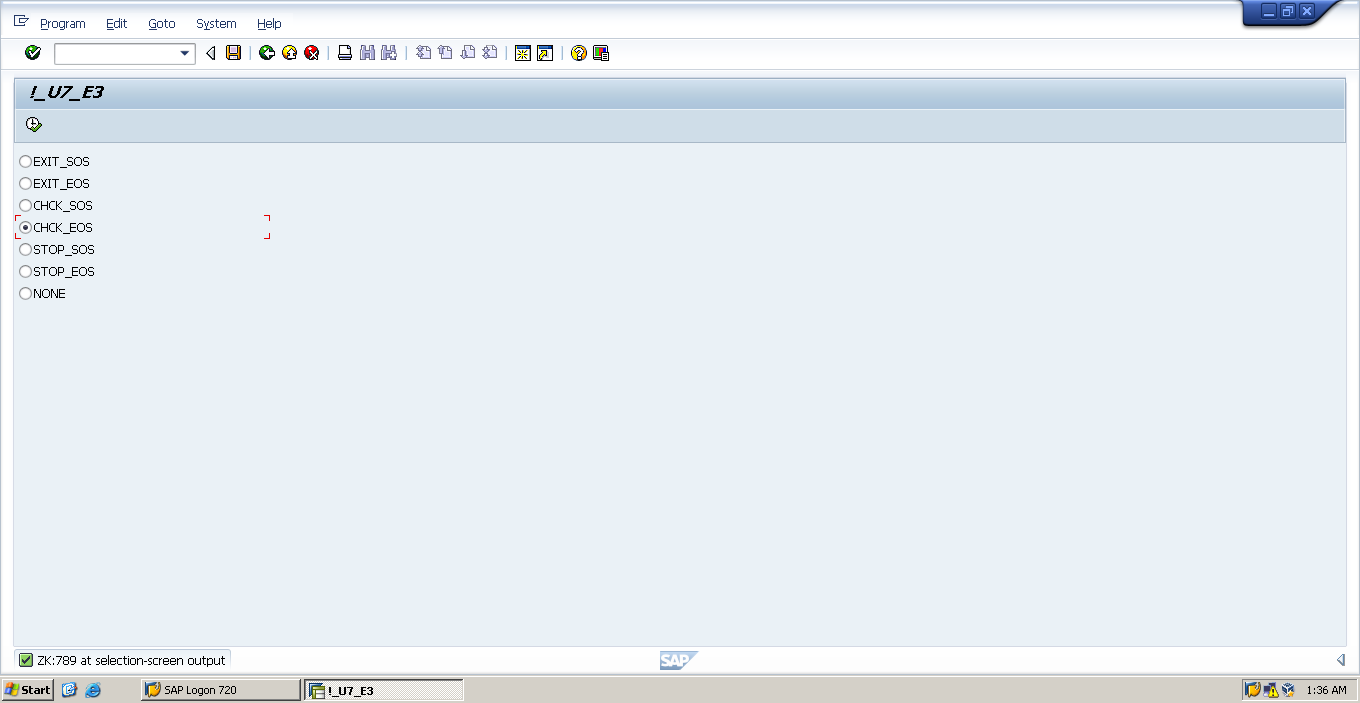
los siguientes mensajes:

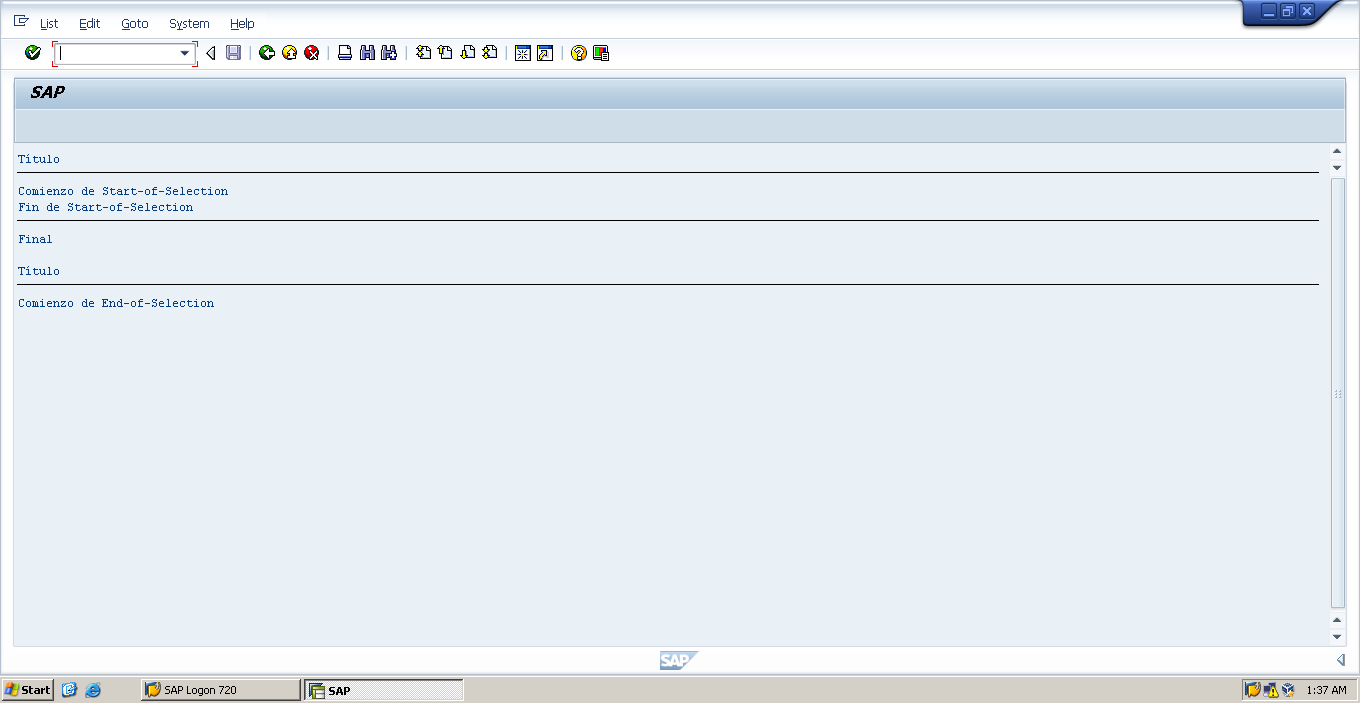


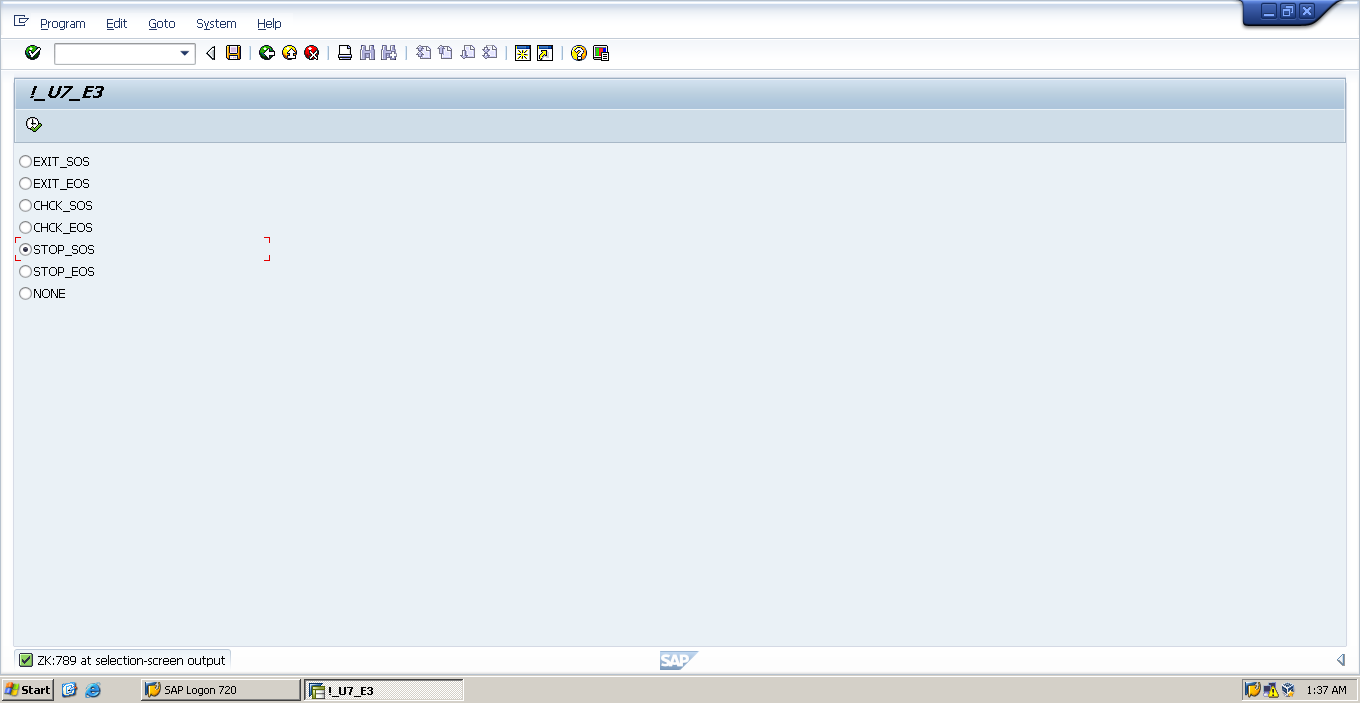
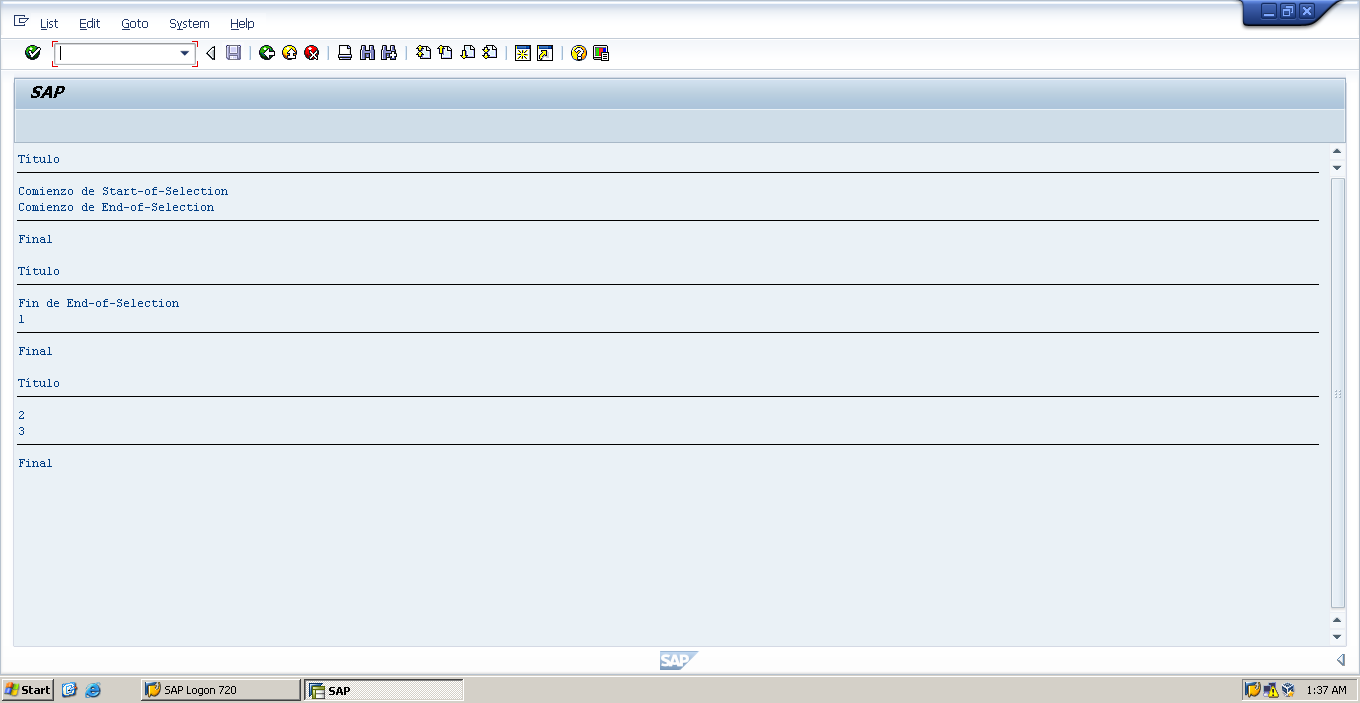
Solo activando check y exit en initialization:

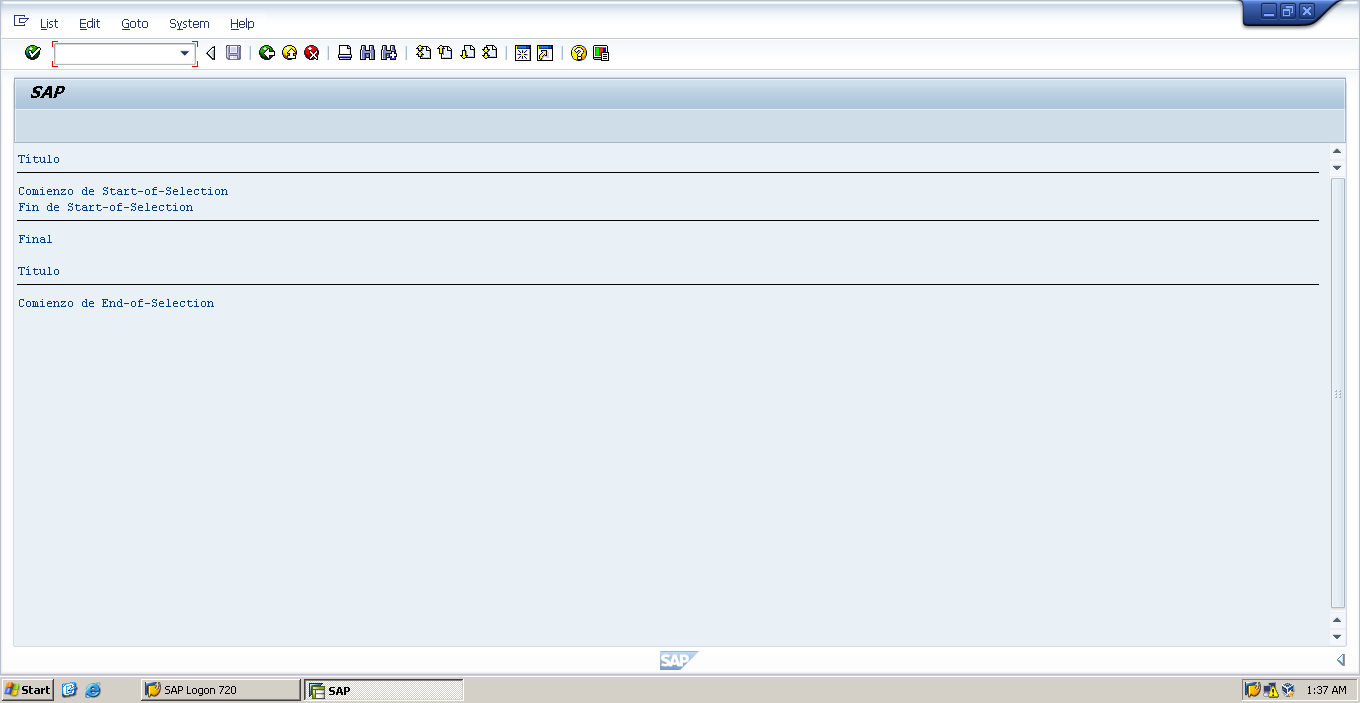
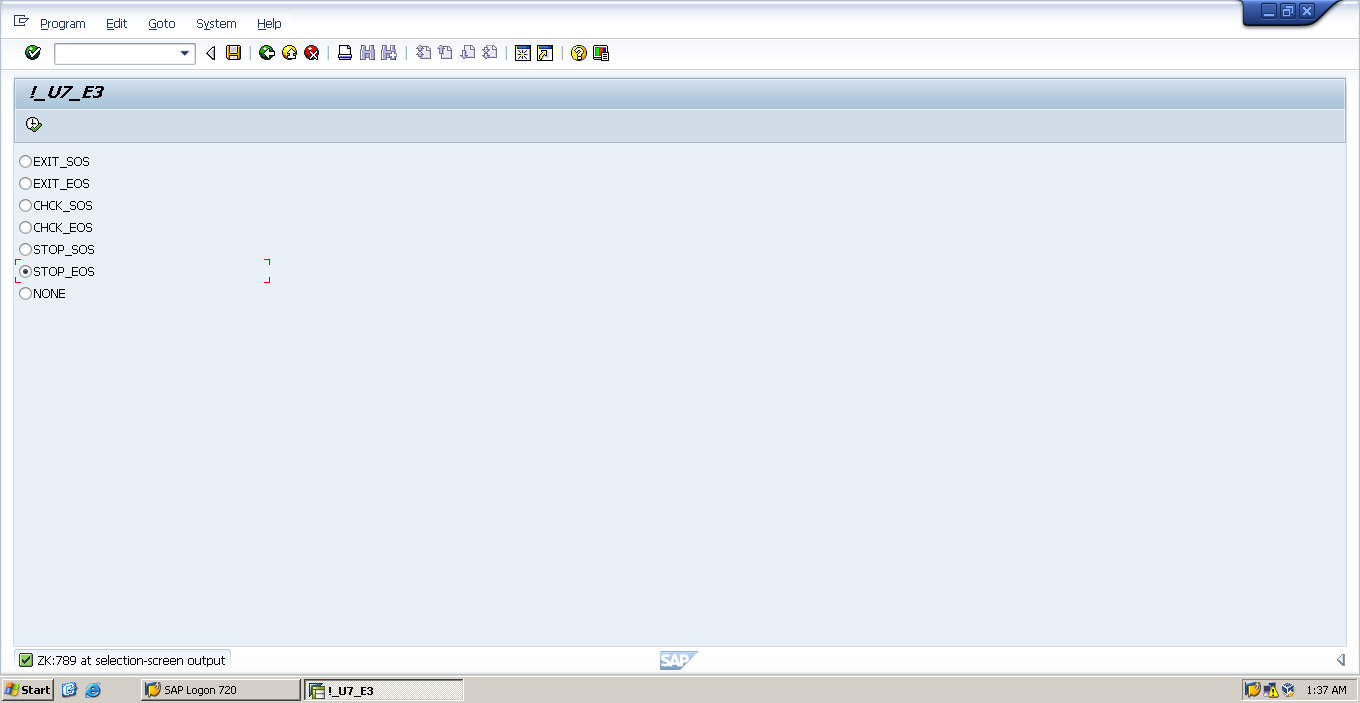


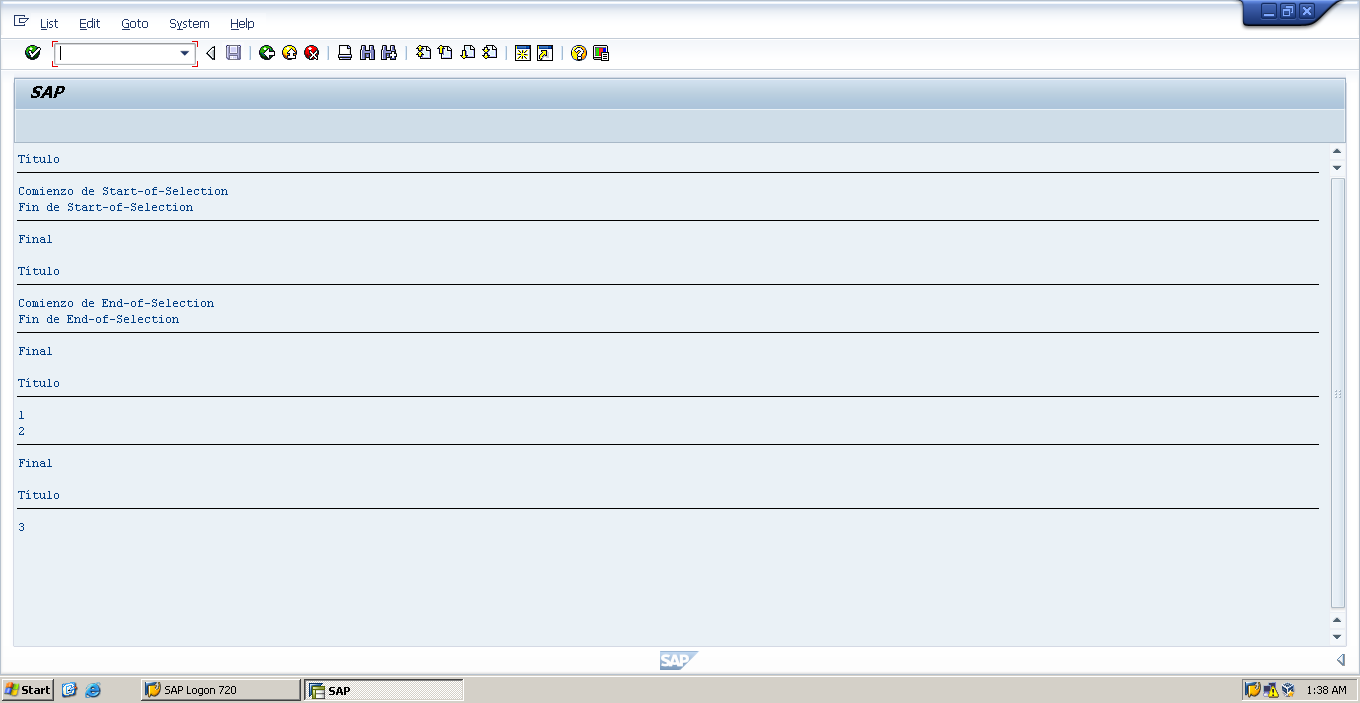
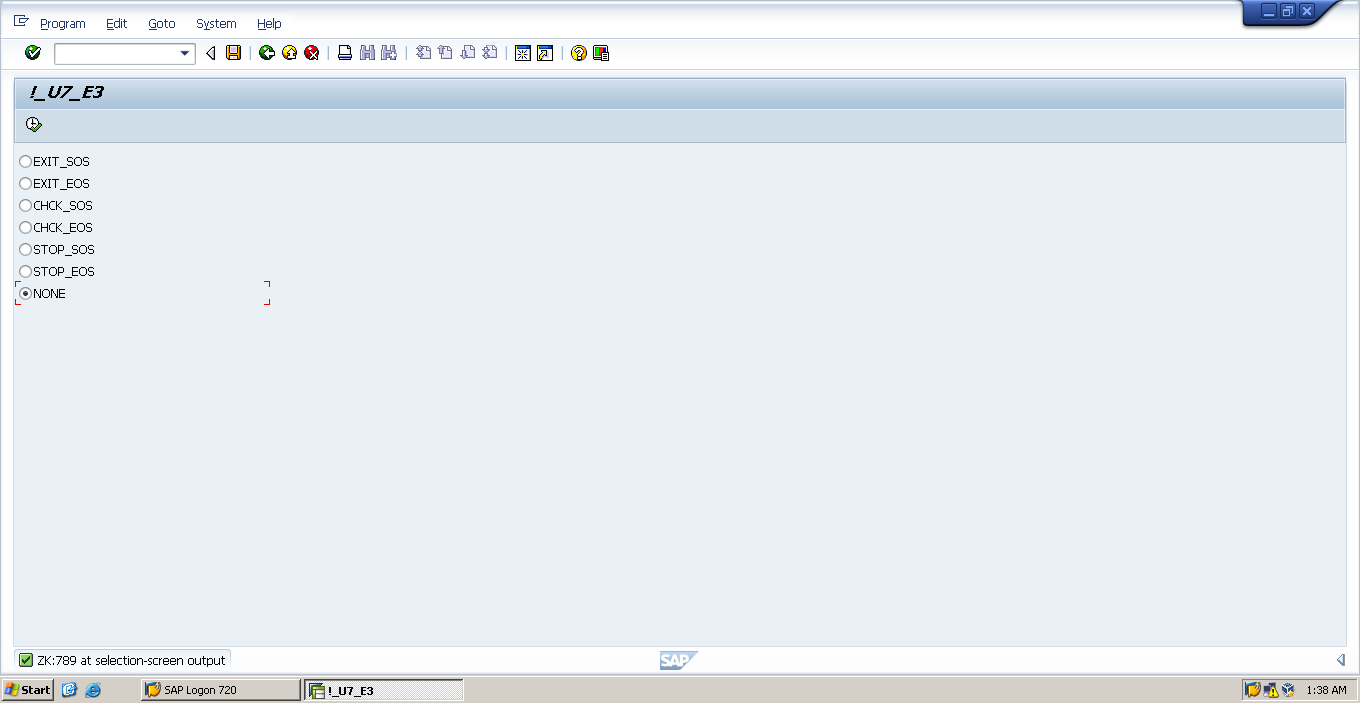


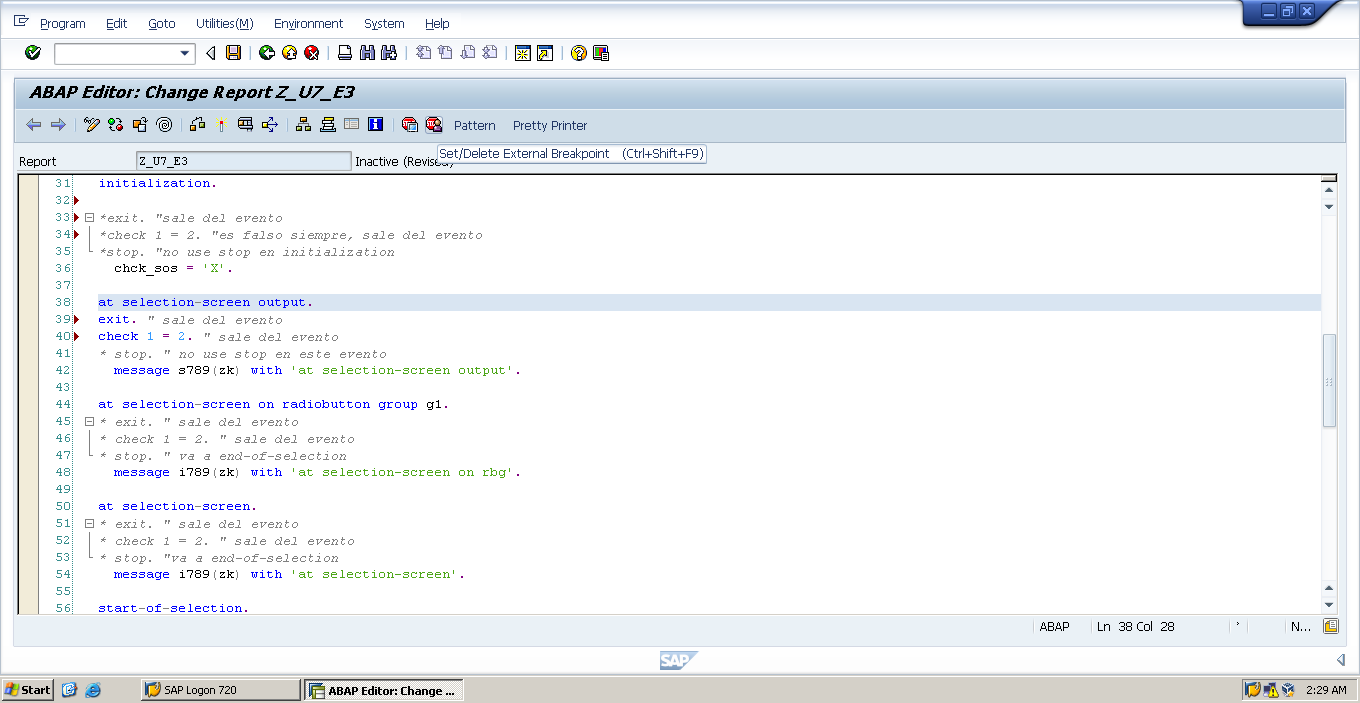




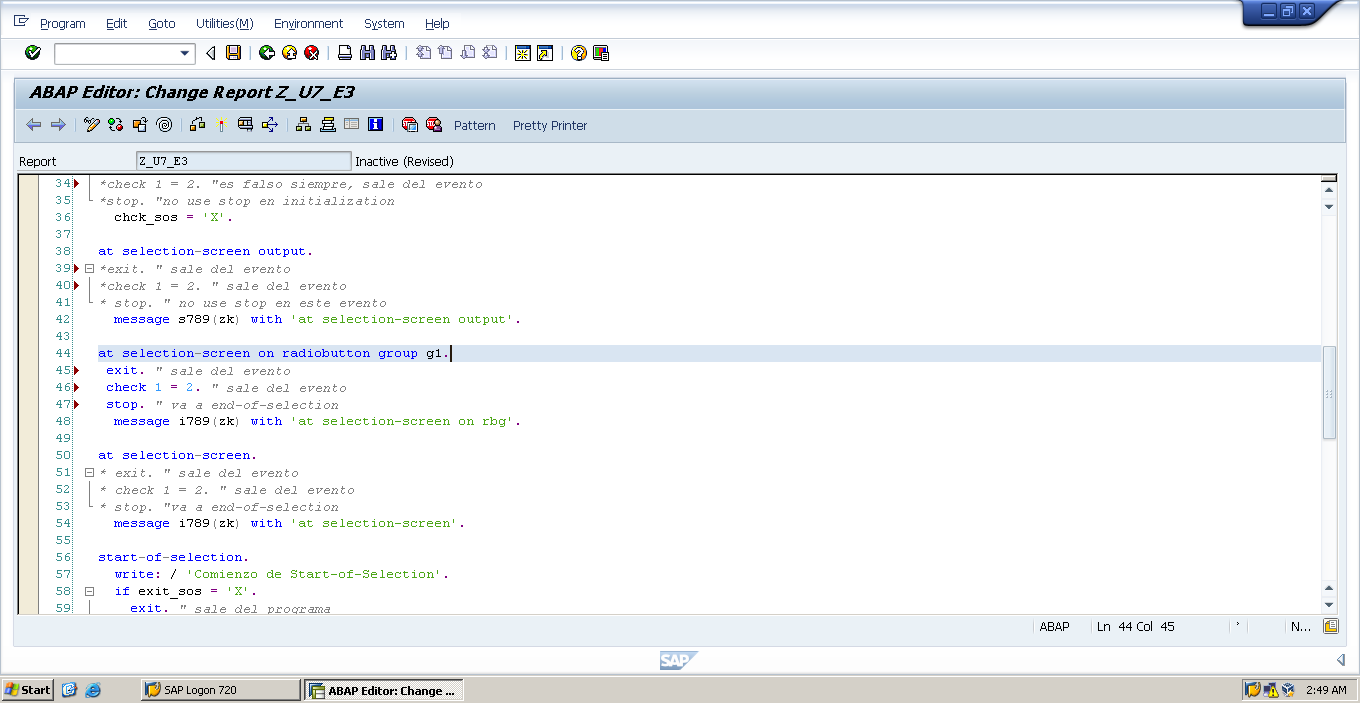


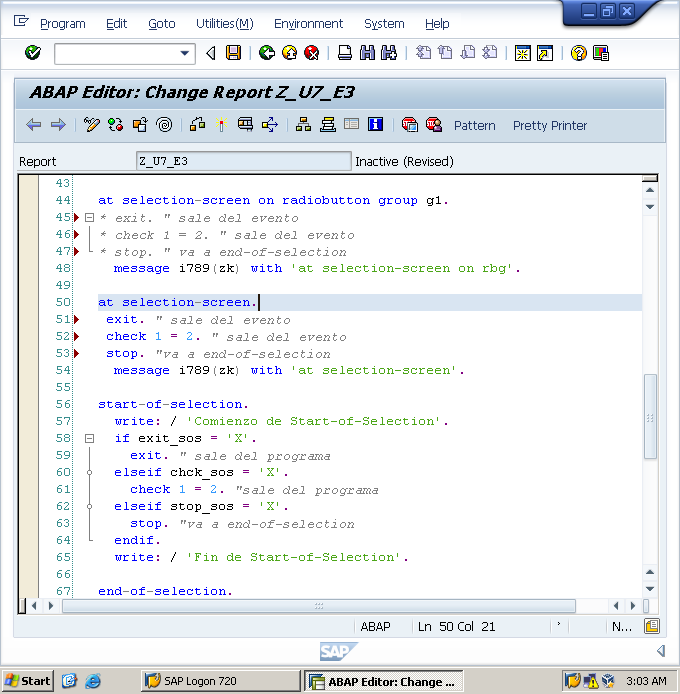




Check y exit en selection-screen output: Los resultados son iguales a los de Check y exit en initialization.

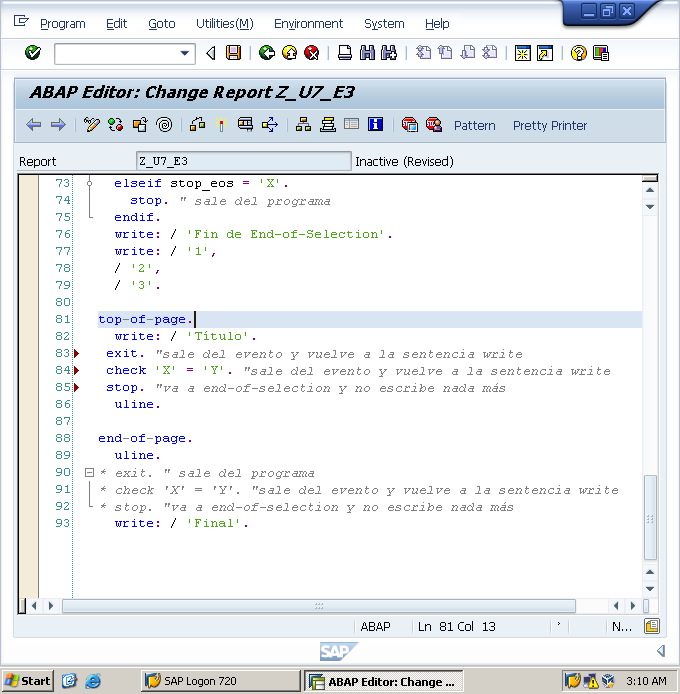
At selection on radiobutton group g1.

Los resultados son iguales a los de Check y exit en initialization y en selection-screen output

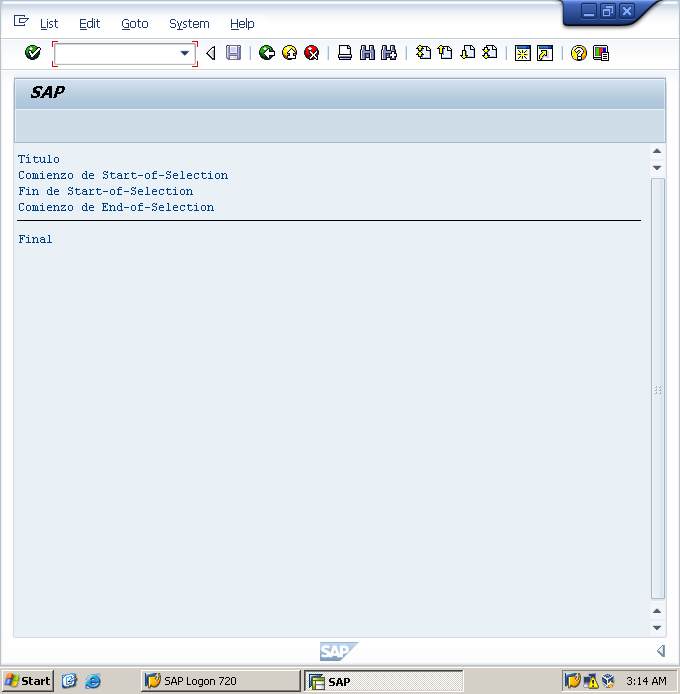
At selection-screen.

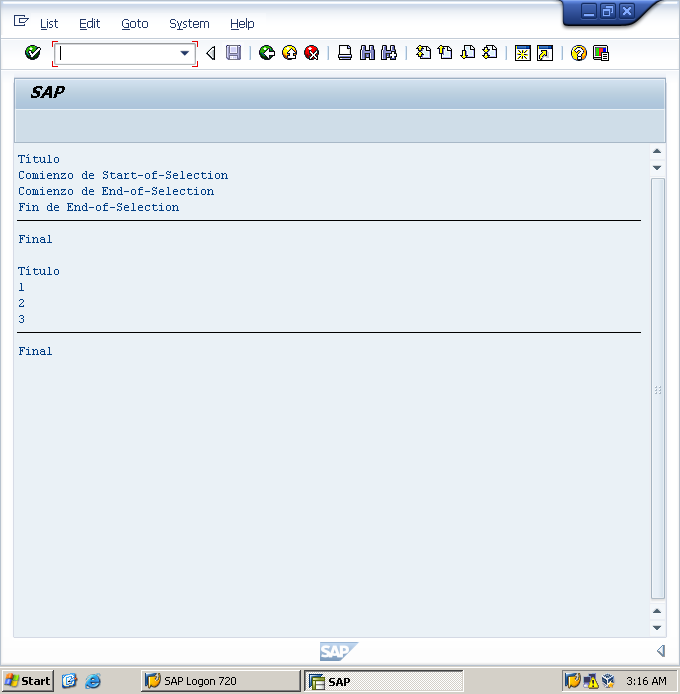
Los resultados son iguales a los anteriores.

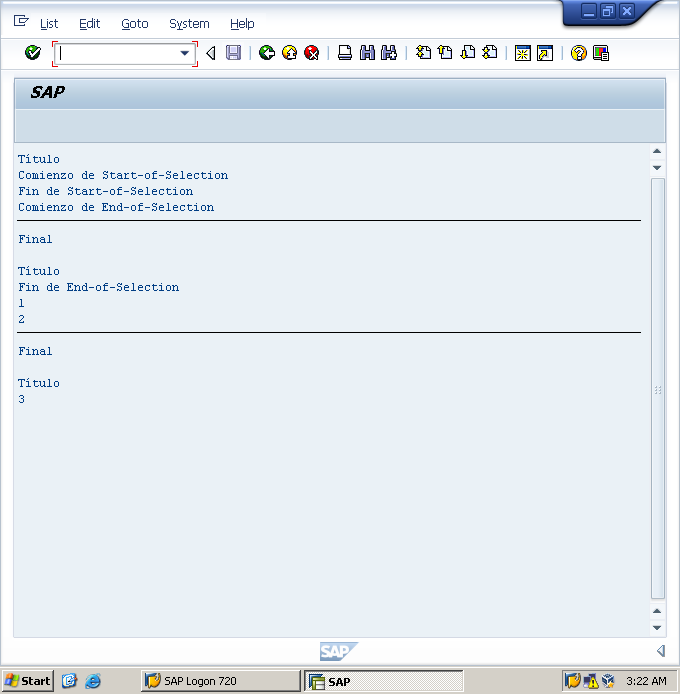
Top-of-page.



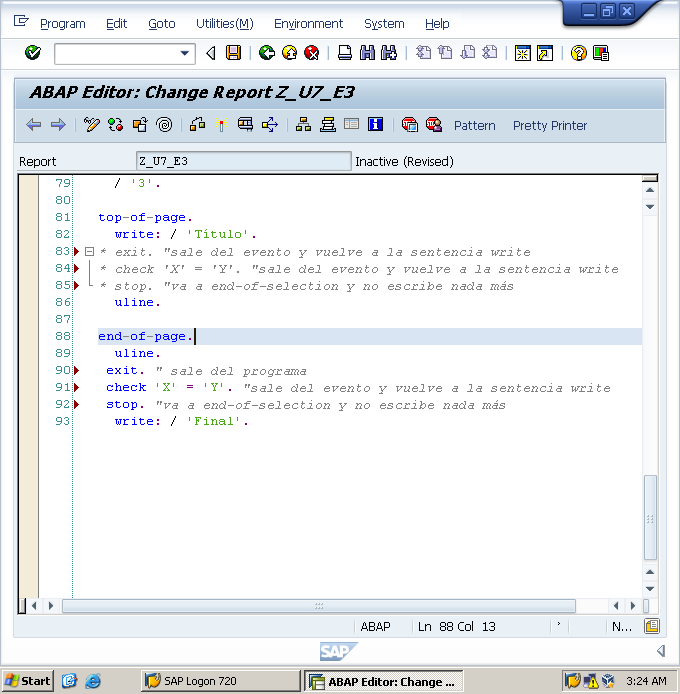
EXIT\_SOS: Igual a los anteriores.

EXIT\_EOS, CHEK\_EOS y STOP\_EOS: 

CHEK\_SOS y STOP\_SOS: 

NONE: 

End-of-page:



EXIT\_SOS: igual a los anteriores.

EXIT\_EOS, CHEK\_EOS y STOP\_EOS: igual al anterior.

CHEK\_SOS y STOP\_SOS: igual al anterior.

NONE: igual al anterior.

Tarea 2

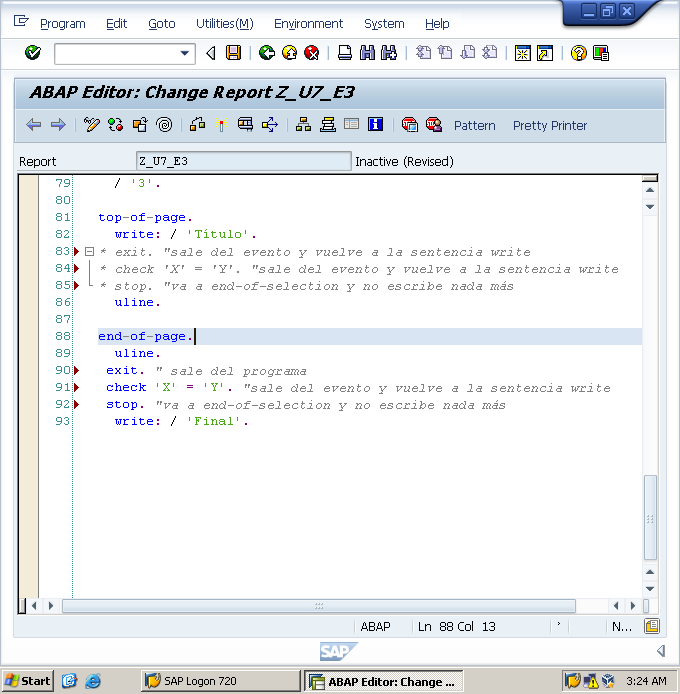
**Explique en particular, con respecto al programa anterior, con el mayor detalle posible el efecto de la instrucción check 1 = 2**

En esta instrucción se genera una condición falsa, debido a que uno no es igual a dos. La utilidad de esta es la desviación del programa a la siguiente acción o evento.

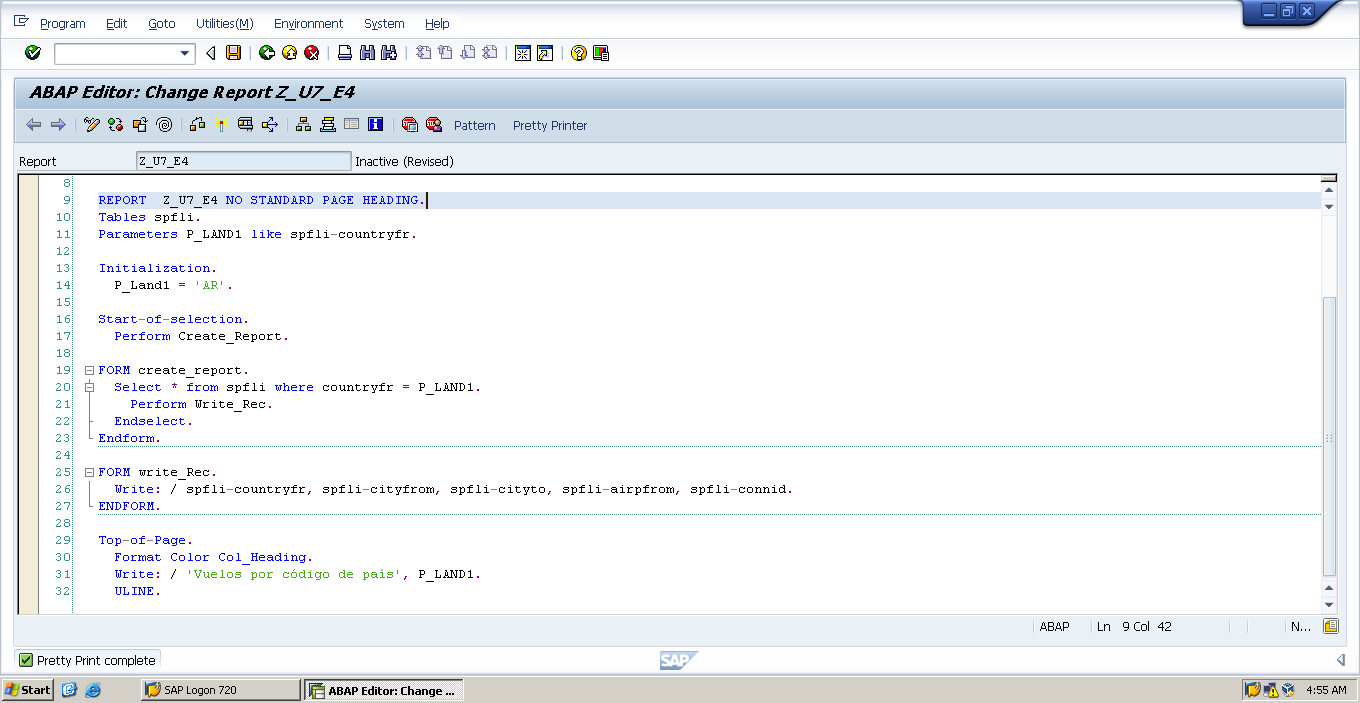
Tarea 3

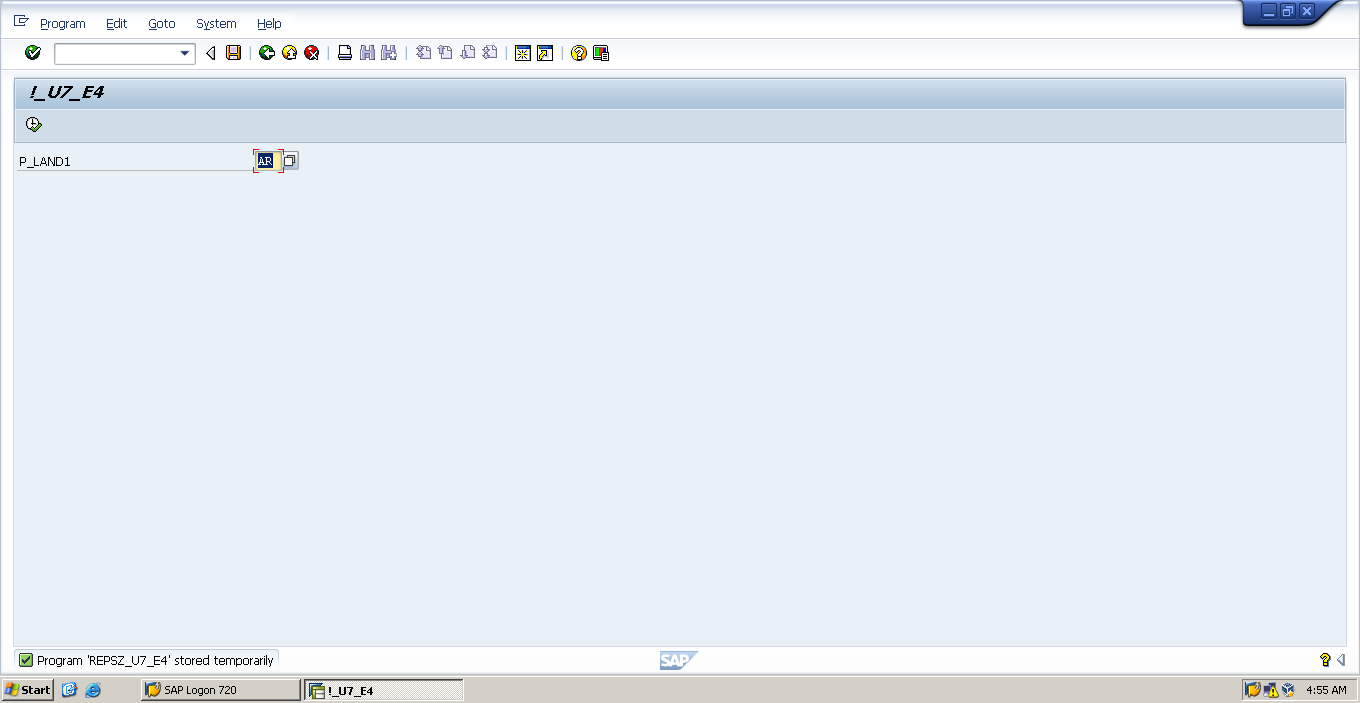
**Ídem anterior para el rol que cumple en la salida presentada al usuario el código que se incluye en el evento end-of-page.**

El código del ídem anterior (check) que se incluye en el evento end-of-page en la salida presentada es: check ‘X’ = ‘Y’. A pesar de que esta igualación no sea correcta, la sentencia write se cumple. Si fuera verdadera, se cumpliría la sentencia write igualmente.



Ejercicio 4 Tarea 1

Crear un informe de que acepte un parámetro de entrada denominado p\_land1. Definirlo como ztxlfa1-land1. Rellene este campo con el valor por defecto ARG, en tiempo de ejecución, mediante el uso del evento initialization. Su programa principal debe tener una sola declaración que será usada para llamar a una subrutina que seleccione registros de ztxlfa1 donde el campo land1 sea igual al parámetro p\_land1. Dentro del select, llamar a otra subrutina para escribir el registro. Utilice el evento top-of-page para crear el título del informe, y escriba el valor de p\_land1 en ese título..



Ejercicio 5 Tarea 1

1. **¿Cómo puedo decidir si voy a utilizar un include, una subrutina interna, una subrutina externo, un evento o un módulo de función para ejecutar mi código?**

Cuando usaría un/a;

Include: para organizar el código fuente ABAP en pequeñas unidades editables que se pueden insertar en cualquier lugar en otros programas ABAP

Subrutina interna: para obtener una mejor visión general del programa principal y utilizar la secuencia correspondiente de enunciados

Subrutina externa: para hacer el programa principal más legible

Evento; En general: para organizar el programa en etiquetas. Particularmente:

INITIALIZATION: para inicializar los campos de entrada de la pantalla de selección o cambiar los valores predeterminados de estos antes de que el usuario ingrese datos en ellos.

AT SELECTION SCREEN OUTPUT: para manipular los atributos reales de la pantalla de selección utilizando el bucle en la funcionalidad de la pantalla. Esto le permite hacer cosas como ocultar campos, atenuarlos para que solo se impriman o intensificarlos, etc.

AT SELECTION-SCREEN: para agregar procesamiento relacionado con la validación de pantalla. Manipular los valores contenidos dentro de los parámetros de la pantalla sel o detener el programa y devolver al usuario a la pantalla de selección mostrando un mensaje de error.

TOP-OF-PAGE: para mostrar un encabezado para la lista.

Módulo de función:

Para encapsular y reutilizar funciones globales en el sistema SAP, para que no se puedan realizar cambios incompatibles en ningún módulo de función que ya se haya lanzado

1. http://www.sapnet.ru/abap21day/ch17/ch17.htm [↑](#footnote-ref-0)
2. http://www.sapnet.ru/abap21day/ch17/ch17.htm [↑](#footnote-ref-1)
3. http://www.sapnet.ru/abap21day/ch17/ch17.htm [↑](#footnote-ref-2)
4. http://www.sapnet.ru/abap21day/ch17/ch17.htm [↑](#footnote-ref-3)
5. http://www.sapnet.ru/abap21day/ch17/ch17.htm [↑](#footnote-ref-4)
6. Material del curso. [↑](#footnote-ref-5)