

C.N.E.A. Biblioteca	
ARCHIVO PUBLICACIONES	
Nº 1	AÑO 1981

02.81.14

CAB - NT 6/81

BASE DE DATOS SOBRE ESTRUCTURA NUCLEAR

E. Browne

y

C. Espinel

Mayo 1981

RESUMEN

En el presente informe se describe la implementación de una base de datos sobre estructura nuclear en la computadora IBM-360/44 del Centro de Cómputo, Centro Atómico Bariloche. Esta base de datos contiene información experimental sobre desintegraciones radioactivas y reacciones nucleares.

Las subrutinas correspondientes al manejo de los archivos, subrutinas útiles para el manejo de caracteres, y un ejemplo de uso de esta base de datos han sido incluidos.

I. INTRODUCCION.

Toda la información sobre esquemas de desintegraciones nucleares (niveles energéticos, números cuánticos, vidas medias, radiación α , β , y γ , emisión de neutrones y protones retardados, etc) publicados en la 7^a edición de la "Table of Isotopes"¹ fue originalmente almacenada en un archivo computarizado. Posteriormente ese archivo fue reestructurado² a fin de hacerlo más eficiente para el manejo, búsqueda, y extracción de datos, convirtiéndose en lo que actualmente se conoce como una "base (banco o biblioteca) de datos". Este archivo contiene hasta la fecha los datos sobre estructura nuclear más uniformemente puestos al día.

Una base de datos como la mencionada tiene usos que van desde simples búsquedas de datos nucleares correspondientes a un isótopo particular hasta extracción ordenada y masiva de datos para estudios sistemáticos de propiedades nucleares. Su utilización en física nuclear aplicada puede ser variada y no menos importante. Como ser: búsqueda y clasificación de isótopos con propiedades físicas (o químicas) específicas requeridas para usos particulares; uso de datos sobre radiaciones emitidas por isótopos radioactivos dentro de un reactor nuclear (por ejemplo: para el cálculo del calor emitido en el núcleo del mismo), etc.

La estructura de esta base de datos ha hecho posible su implementación en computadoras relativamente pequeñas actualmente de mayor disponibilidad en diversos centros de enseñanza, investigación, y desarrollo.

En el presente informe se describe la implementación de esta base de datos (originalmente utilizada en computadoras de las series CDC-6600/7600 de gran capacidad de memoria rápida y de disco) a la computadora IBM-360/44 del Centro de Cómputos, Centro Atómico Bariloche. En la parte final del mismo se da un ejemplo simple de búsqueda de datos a fin de describir su manejo.

II. ORGANIZACION GENERAL DE LA BASE DE DATOS.

La estructura original² de la base de datos ha sido mantenida, por lo cual sugerimos a los lectores que deseen tener mayor información consulten la referencia 2.

La base de datos implementada en la computadora IBM-360/44 consiste de tres archivos guardados en cintas magnéticas y cuyos contenidos se describen a continuación:

1. Niveles energéticos poblados en desintegraciones de isótopos radioactivos (17.813 registros de 117 caracteres c/u).

2. Radiaciones emitidas en la desintegración de isótopos radioactivos (36.790 registros de 117 caracteres c/u).

3. Niveles energéticos poblados en reacciones nucleares (35.780 registros de 117 caracteres c/u).

Esta organización de la base de datos ha sido hecha de acuerdo a la disponibilidad de memoria (rápida: 128 Kbytes, disco: 1 Megabyte) y al sistema operativo de la computadora IBM-360/44 utilizada.

Cada archivo está compuesto de vectores bidimensionales de no más de 16.000 elementos enteros de media palabra IBM (2 "bytes") cada uno, y contiene toda la información sobre la estructura nuclear de cada isótopo. Los vectores son de dos tipos: "vectores L", que contienen información sobre niveles energéticos, y "vectores D", con las propiedades de las radiaciones emitidas en la desexcitación de niveles energéticos. Los vectores están almacenados en orden creciente con respecto al número de masa A, y dentro de un mismo A, en orden creciente con respecto al número atómico Z. Los grupos de vectores correspondientes a una cadena de masa (isótopos con el mismo valor de A) están separados por registros que contienen la secuencia de caracteres \$\$\$\$\$. Esto está esquematizado en la Figura 1.

III. ESTRUCTURA Y CONTENIDO DE LOS VECTORES L(K,I) CORRESPONDIENTES A LA DESINTEGRACION DE ISOTOPOS RADIOACTIVOS.

Los vectores L son enteros ("integers") y de media palabra IBM (2 "bytes"). Por lo tanto cada elemento puede contener un máximo de 2 caracteres. La dimensión es L(150,58). El subíndice I identifica las diferentes propiedades asociadas con el K-ésimo elemento del vector. Los formatos correspondientes están dados en la Figura 2. Cuando alguna propiedad requiere más de media palabra IBM (2 caracteres) se utilizan dos o más elementos para almacenarla. Todos los datos son "números generalizados", definidos como números decimales o exponenciales que pueden ser precedidos por uno de los siguientes símbolos:

<u>Símbolo</u>	<u>Interpretación</u>	<u>Cód.Hexdecimal</u>	<u>Cód.de Perf.</u>
>	>	AE	11-0-6-8
<	<	4C	12-4-8
#	≈	7D	5-8
➤	➤	E0	0-2-8
◀	◀	6C	0-4-8

Por ejemplo: > 0.580, # 352, < 200

1. El primer elemento ($K=1$) del vector contiene la siguiente información que define el tamaño y contenido del mismo:

$L(1,1) = A$ (número de masa)

$L(1,2) = Z$ (número atómico)

$L(1,3) = NLEV$ (número total de niveles)

$L(1,4) = NPAR$ (número total de nucleidos "padres")

$L(1,5+58) =$ no son utilizados.

2. Toda la información referente a los nucleidos padres está contenida en los elementos $2 \leq K \leq NPAR+1$. Las propiedades nucleares son las siguientes:

2.1 $L(K,I) (I=1,10)$ - Nucleidos Padres (10 elementos)

Los posibles nombres de los nucleidos padres están dados en la referencia 2.

2.2 $L(K,I) (I=11,15)$ - Modo de Desintegración (5 elementos)

Los posibles modos de desintegración están dados en la referencia 2.

2.3 $L(K,I) (I=16,21)$ - Vida Media (5 elementos)

Las vidas medias son "números generalizados" seguidos de "unidades", estas últimas descriptas en la referencia 2. Es posible la existencia de un carácter "blanco" entre el número y la unidad. Los números precedidos de una W se refieren a ancho de nivel ("level width") en Mev.

2.4 $L(K,I) (I=21,30)$ - Spin (10 elementos)

Estos números cuánticos pueden estar entre paréntesis (por ejemplo $(1/2+)$), o entre corchetes (por ejemplo $[3/2+]$). Los códigos para los corchetes son los siguientes:

<u>Símbolo</u>	<u>Código Hexadecimal</u>	<u>Código de Perforación</u>
----------------	---------------------------	------------------------------

[	4A	12-2-8
]	5A	11-2-8

Todos los posibles "spines" y números cuánticos están dados en la referencia 2.

2.5 L(K,I)(I=31,39) - Energía Disponible para la Transición ("Q-value")(9 elementos)

Puede ser un "número generalizado" o un número con su error experimental. Ejemplos:

4.18	(aproximadamente 4.18 Mev)
5.015+-30	(5.015 ± 0.030 Mev)

2.6 L(K,I)(I=40) - Identificador del Nucleido Padre
(1 elemento)

En caso de que hubiese más de un nucleido padre, es utilizado para identificar al correspondiente que alimenta ese nivel.

En ciertos casos este identificador puede ser un carácter blanco.

3. La información sobre los niveles energéticos está contenida en los elementos del vector con $K > NPAR+2$. Las propiedades nucleares son las siguientes:

3.1 $L(K,I)(I=1,5)$ - Energía del Nivel. (5 elementos)

La energía es un "número generalizado" ordenado en forma creciente con respecto a K , pudiendo ser seguido de los siguientes modificadores:

<u>Modificador</u>	<u>Interpretación</u>	<u>Código Hexadecimal</u>	<u>Código de Perforación</u>
I	Isómero		
?	Nivel incierto	AF	11-0-7-8
C	Nivel complejo (varios niveles)		
(C)	Nivel complejo probable.		

Las energías están dadas en la gran mayoría de los casos en Mev. En muy pocos casos están en KeV.

3.2 $L(K,I)(I=6,20)$ - Spin. (15 elementos)

La descripción de esta propiedad está dada bajo "nucleidos padres en la sección 2.4.

3.3 $L(K,I)(I=21,25)$ - Vida media (5 elementos)

La descripción de esta propiedad está dada bajo "nucleido padre" en la sección 2.3.

3.4 L(K,I)(I=26,29; 34,37; 42,45; 50,53) - Identificación de Nucleidos Padres/Modo de Desintegración (16 elementos)

L(K,I)(I=30,33; 38,41; 46,49; 54,57) - Abundancia

(16 elementos)

Estos grupos de 4 elementos contienen los diferentes identificadores del nucleido padre (ver sección 2.6) y las abundancias de alimentación al nivel considerado asociadas al nucleido padre en cuestión. El primer carácter contiene el identificador, y los restantes describen el modo de desintegración. Por ejemplo, CB⁺ significa un nivel alimentado por β^+ a partir del nucleido padre C. Si el modo de desintegración está en blanco la información correspondiente está dada en los elementos del vector del nucleido padre (ver sección 2.2). Cuando el nucleido padre decae por EC + β^+ , pero la abundancia de alimentación al nivel es sólo EC (captura electrónica), el modo debe estar especificado. Por ejemplo: DEC. (D identifica al nucleido padre y EC a la abundancia). Las abundancias de alimentación pueden ser números generalizados que representan la alimentación en %, o si están precedidas por el carácter R, en valores relativos.

3.5 L(K,I) (I=58) - Identificador de Transición (1 elemento)

Este elemento es sumamente importante porque vincula a los vectores L (niveles) con los D (radiaciones). El mismo contiene el valor del subíndice K del elemento del vector D correspondiente a la última transición que desexcita el nivel (ver sección V). Si el nivel no es desexcitado por ninguna transición el valor de este elemento es $L(K,58) = 0$, excepto para el elemento correspondiente al estado fundamental (energía cero), que tiene un valor $L(NPAR+2,58)=1$. El número de transiciones que desexcitan un nivel K está dado por: $(L(K,58) - L(K',58))$ donde $K' (<K)$ corresponde al nivel más cercano que está desexcitado por transiciones.

IV. ESTRUCTURA Y CONTENIDO DE LOS VECTORES $L(K,I)$ CORRESPONDIENTES A REACCIONES NUCLEARES

Los vectores L son enteros ("integers") y de media palabra IBM (2"bytes"). Por lo tanto cada elemento puede contener un máximo de 2 caracteres. La dimensión es $L(200,56)$.

Así como en el caso de desintegración radioactiva, el subíndice K identifica los diferentes niveles, y el subíndice I las correspondientes propiedades nucleares. Los formatos correspondientes están dados en la figura 2. Muchas de las propiedades nucleares son idénticas a las descriptas para el caso de desintegración radioactiva, por lo tanto sólo se

hará referencia a aquellas que son diferentes.

1. El primer elemento ($K=1$) del vector contiene la siguiente información que define el tamaño y contenido del mismo:

$L(1,1) = A$ (número de masa)

$L(1,2) = Z$ (número atómico)

$L(1,3) = NLEV$ (número total de niveles)

$L(1,4) = NREACT$ (número total de reacciones nucleares)

$L(1,5+56) =$ no son utilizados

2. Toda la información sobre las reacciones nucleares que pueblan estos niveles está contenida en los elementos con $2 < K < NREACT + 1$. Las propiedades nucleares son las siguientes:

2.1 $L(K,I) (I=1,20)$ - Lista de Reacciones Nucleares
(20 elementos)

Nombres típicos de reacciones nucleares y una lista de posibles proyectiles y partículas emitidas están dados en la referencia 2.

2.2 $L(K,I) (I=21)$ - Identificador de la Reacción(1 elemento)
Lo mismo que en desintegración radioactiva, el "identificador" es un carácter (que puede ser blanco) que identifica a la reacción nuclear.

3. La información correspondiente a niveles energéticos está contenida en los elementos del vector con $K > NREACT + 2$. La información es idéntica a la de desintegración radioactiva, con las siguientes excepciones:

3.1 $L(K,I)(I=26,55)$ - Identificación de Reacciones
(30 elementos)

Estos elementos del vector contienen los identificadores de las reacciones descriptos en la sección 2.2. La alimentación de grupos de niveles no resueltos energéticamente está descrita en la referencia 2.

3.2 $L(K,I)(I=56)$ - Identificador de Transición (1 elemento)

Este elemento es idéntico al elemento correspondiente $L(K,58)$ en desintegración radioactiva.

V. ESTRUCTURA Y CONTENIDO DE LOS VECTORES $D(K,I)$ DE RADIACIONES CORRESPONDIENTES A DESEXCITACIONES DE NIVELES.

Los vectores D son enteros ("integers") y de media palabra IBM (2 "bytes"). Por lo tanto cada elemento puede contener un máximo de 2 caracteres. La dimensión es $D(400,39)$. El subíndice K identifica las diferentes transiciones, y el subíndice I las correspondientes propiedades nucleares. Los formatos correspondientes están dados en la Figura 2.

1. El primer elemento ($K=1$) del vector contiene la siguiente información que describe el tamaño y contenido del mismo:

$D(1,1) = A$ (número de masa)

$D(1,2) = Z$ (número atómico)

$D(1,3) = NTRANS$ (número total de transiciones)

$D(1,4+39) =$ no son utilizados

2. Los elementos restantes ($K>2$) del vector contienen información sobre las transiciones. Estas pueden ser rayos gamma o partículas (incluyendo captura electrónica). Si no ha sido observada ninguna transición $D(1,3)=0$. Las propiedades nucleares son las siguientes:

2.1 $D(K,I)(I=1,5)$ - Energía de la Transición (5 elementos)

Para rayos gamma la energía es un número generalizado que puede estar seguido de un asterisco "*" (significa que la transición ha sido ubicada más de una vez en el esquema de niveles) o de un signo de interrogación "?" (transición incierta). Para partículas la energía no está dada. Consultar referencia 2 para más información.

2.2 $D(K,I)(I=6,12)$ - Intensidad Absoluta (7 elementos)

Las intensidades pueden ser "números generalizados", números con errores experimentales (por ejemplo 3.1 ± 2), o el carácter "w" (transición débil). Para rayos gamma la intensidad se refiere a fotones. Intensidades de electrones de conversión interna están descriptas en la referencia 2.

2.3 $D(K,I)(I=13,19)$ - Intensidad Relativa (7 elementos)

Lo dicho para intensidades absolutas (sección 2.2) se aplica también a intensidades relativas.

2.4 $D(K,I)(I=20,33)$ - Multipolaridad (14 elementos)

Consultar la referencia 2 para una descripción de las posibles formas de multipolaridad almacenadas en estos elementos.

2.5 $D(K,I)(I=34,38)$ - Energía del Nivel Final (5 elementos)

Este "número generalizado" identifica al nivel poblado por el rayo gamma (o la partícula). Consultar referencia 2 para más detalles.

2.6 $D(K,I)(I=39)$ - Identificador de Nivel (1 elemento)

Este elemento es sumamente importante pues vincula a los vectores de niveles $L(K,I)$ con los de transiciones $D(K,I)$. El "identificador de nivel" es el subíndice K del vector de niveles $L(K,I)$ correspondiente al nivel inicial de la transición.

VI. ORGANIZACION Y MANEJO DE ARCHIVOS.

Como ya fuera mencionado anteriormente (sección II) los vectores de la base de datos están almacenados en archivos de cinta magnética. Los archivos contienen los vectores correspondientes a un número de masa A y están separados entre sí por registros que contienen la secuencia de caracteres \$\$\$\$.

Si no existe información sobre una cadena particular de masa A, ésta viene representada por un registro con la marca de fin-de-archivo \$\$\$\$\$. (ver Figura 1)

Los archivos que constituyen la base de datos nucleares están almacenados en 5 cintas magnéticas de 2400 pies de longitud cada una. El código de grabación es EBCDIC, la longitud de registro es de 117 caracteres, el factor de bloqueo es 1 (no está bloqueada), y la densidad de grabación es de 1600 "bpi". Todas las cintas tienen marcas de fin-de-información al final de las mismas. Estas marcas son detectables con el parámetro END de la instrucción READ del lenguaje FORTRAN.

La distribución de la información e identificación de las cintas son las siguientes:

<u>Cinta</u> <u>Número</u>	<u>Rótulo</u>	<u>Unidad de Cinta</u> <u>Unidad de entrada</u>	<u>Contenido</u>
22	U00333	081 (2)	L(K,I). Niveles. Desintegración radioactiva. A=1 - 256-
15	U00334	081 (2)	D(K,I).Radiación.Desintegración radioactiva. A=1 - 200-
58	U00337	080 (3)	D(K,I).Radiación.Desintegración radioactiva. A=201-256-
28	U00335	081 (2)	L(K,I). Niveles. Reacciones Nucleares A=1 - 200
51	U00338	080 (3)	L(K,I). Niveles. Reacciones Nucleares A=201 - 251

El manejo de archivos de esta base de datos ha sido organizado de acuerdo a las posibilidades de la computadora IBM-360/44 y los equipos periféricos disponibles. Una descripción esquemática está dada en la Figura 3.

La búsqueda del archivo deseado se realiza en la cinta magnética utilizando una de las siguientes subrutinas de manejo de archivos:

- SKPLLD - Para los archivos de vectores $L(K,I)$ de Desintegración Radioactiva.
- SKPLLR - Para los archivos de vectores $L(K,I)$ de Reacciones Nucleares.
- SKPLG - Para los archivos de vectores $D(K,I)$ de Desintegración Radioactiva.

Estas subrutinas después de localizar el archivo buscado lo almacenan en registros de 360 "bytes" de disco. El sector del disco utilizado está preparado para guardar hasta 1200 registros de la longitud mencionada. (Disco EICREA, Sector CHARLY). El procedimiento para crear un sector de este tipo en disco está descrito en el Apéndice IV. Las subrutinas de manejo de archivos "llaman" a dos subrutinas de "traducción de formato" escritas en lenguaje ASSEMBLER:

TRAD - Esta subrutina cambia el formato alfanumérico (A) de los primeros 4 elementos del vector L o D ($K=1$; $I=1,4$) a formato entero (I).

TRAD1 - Efectúa el mismo cambio que TRAD pero para los elementos $L(K,58)$ y $D(K,39)$ de desintegración radioactiva, y $L(K,56)$ de reacciones nucleares. Una vez hecho el cambio estos elementos son escritos en el disco. Esto es necesario pues los valores almacenados en estos elementos del vector deben ser números con los cuales se puedan efectuar operaciones aritméticas.

Una vez que la información correspondiente a una cadena de masa A ha sido almacenada en disco, se utilizan las siguientes subrutinas para buscar y almacenar en memoria rápida el vector correspondiente al isótopo deseado:

LOADLD - Para archivos de vectores $L(K,I)$ de desintegración radioactiva.

LOADLR - Para archivos de vectores $L(K,I)$ de reacciones nucleares.

LOADG - Para archivos de vectores $D(K,I)$ de desintegración, radioactiva.

La búsqueda final de las propiedades nucleares deseadas se realiza en memoria rápida ("core memory") y los resultados pueden ser almacenados en disco, cinta magnética, tarjetas perforadas, o impresos en papel.

Todo programa principal ("main program") debe contener las siguientes instrucciones que definen los vectores de "media palabra IBM", el sector del disco utilizado, y el sector de memoria rápida en el que están almacenados los vectores:

```
INTEGER * 2  L  (o D, según el caso)
DEFINE FILE 1 (600, 360, L, IDREC)
COMMON/LARRAY/L  }
COMMON/DARRAY/D  }  (Según el caso)
```

VII. SUBROUTINAS PARA EL MANEJO DE ARCHIVOS DE LA BASE DE DATOS.

Las siguientes subrutinas (ver listados en apéndice I) pueden ser utilizadas para manejar los archivos de la base de datos.

1. SKPLLD (IFS, IFL, KOUT) Esta subrutina efectúa el manejo de archivos en la cinta magnética. Transfiere a disco un archivo de vectores correspondiente a niveles poblados en la desintegración radioactiva de isótopos pertenecientes a una cadena de masa "A".

IFS(parámetro de entrada)--Número de archivos (registros con \$\$\$\$) que deben ser "saltados" en la cinta magnética.

IFL(parámetro de entrada)--Indicador. Si IFL=1 el archivo de vectores debe ser almacenado en disco. Si IFL=0, no.

KOUT(parámetro de salida)--Cantidad de registros de cinta magnética almacenados en disco. Si IFL=0 resulta KOUT=0.

Si IFL =1, la subrutina carga el archivo de vectores en el disco y deja posicionada la cinta después de la "marca" \$\$\$\$ de fin-de-archivo correspondiente al archivo almacenado en disco.

Si IFL = 0, la subrutina "salta" las "marcas" de fin-de-archivo \$\$\$\$ especificadas por IFS dejando a la cinta magnética posicionada después de la última "marca" "saltada". El sector utilizado en el disco debe estar definido en el programa principal por la instrucción: DEFINE FILE 1 (600,360, L, IDREC).

Mensajes de error

"IFL SHOULD BE EQUAL TO 1 OR 0, IFL X" - El valor de IFL es diferente de 1 y de 0.

"AFTER 100000 ATTEMPTS IT COULD NOT FIND AN END-OF-FILE \$\$\$\$ MARK"-La "marca" de fin-de-archivo \$\$\$\$ no pudo ser detectada.

Sólo podría ocurrir en caso de un funcionamiento anormal del sistema operativo o algún defecto de la cinta magnética.

"END OF INFORMATION ON TAPE" - Se ha tratado de leer un registro pasando la "marca" de fin-de-información en la cinta.

Se recuerda que ésta subrutina utiliza las subrutinas TRAD Y TRAD1 y por lo tanto deben ser suministradas.

Ejemplo 1. Después de haber sido leído el primer archivo (A=1) de la cinta, se desea "saltar" 10 archivos y transferir el archivo de vectores correspondientes a A=11 a disco.

CALL SKPLLD (10, 1, KOUT)

Ejemplo 2. Se ha leído el archivo número 28. Se desea "saltar" 15 archivos y dejar la cinta posicionada para leer el archivo siguiente.

CALL SKPLLD (15, 0, KOUT)

La cinta ha quedado posicionada para leer el archivo número 44. El valor del parámetro KOUT es 0.

2. SKPLLR (IFS, IFL, KOUT, IEND) Esta subrutina es similar a la descrita en la sección anterior (VII-1) pero debe ser utilizada para niveles poblados en reacciones nucleares. El significado de los parámetros IFS, IFL, y KOUT es el mismo que para la subrutina SKPLLD descrito en la sección anterior. El parámetro IEND está relacionado con el hecho de que los datos correspondientes a niveles poblados en reacciones nucleares están contenidos en 2 cintas magnéticas. Su significado es el siguiente:

IEND (parámetro de entrada)--Indica la sección de la base de datos que se desea utilizar.
IEND=2 corresponde a las cadenas de masa A=1 - 200 almacenadas en la cinta U00335 que debe ser montada en la unidad 081.
IEND=3 corresponde a las cadenas de masa A=201 - 251 almacenadas en la cinta U00338 que debe ser montada en la unidad 080.

Mensaje de error.

"IFL SHOULD BE EQUAL TO 1 OR 0" - Ver sección VII-1, SKPLLD.
"AFTER 100000 ATTEMPTS IT COULD NOT FIND AN END-OF-FILE \$\$\$\$ MARK" - Ver sección VII-1, SKPLLD.
"ERROR. IEND SHOULD BE 2 OR 3. IEND X" - El valor de IEND es diferente de 2 y de 3.
"END OF FILEX" Se ha tratado de leer un archivo pasando la "marca" de fin-de-información en la cinta (X=2 o 3).

En caso de que se desee utilizar información correspondiente a A=1-251, las cintas U00335 y U00338 deben ser montadas en las unidades respectivas. Si IEND=2 al comenzar el programa, automáticamente toma el valor 3 al llegar al final de la cinta U00335 y la ejecución continua con la cinta U00338.

Ejemplo 1. Se desea leer archivos de la cinta U00338 (unidad 080).

CALL SKPLLR (IFS, IFL, KOUT, 3)

Ejemplo 2. Se desea leer archivos de la cinta U00335. Al finalizar esta cinta se desea continuar leyendo archivos de la cinta U00338.

CALL SKPLLR (IFS, IFL, KOUT, 2)

Al terminar de leer la cinta U00335 el valor de IEND pasa automáticamente a 3, y continua leyendo la cinta U00338.

3. SKPLG (IFS, IFL, KOUT, IEND) Esta subrutina es similar a las dadas anteriormente (SKPLLD y SKPLLR) pero debe ser utilizada para radiaciones (principalmente gamma) emitidas en la desexcitación de niveles poblados en desintegraciones radioactivas.

El significado de todos los parámetros es el mismo que en el de la subrutina SKPLLR descrito en la sección VII-2. En este caso los datos también están almacenados en dos cintas magnéticas. El valor del parámetro IEND=2 corresponde a las cadenas de masas A=1 - 200 contenidas en la cinta U00334 (unidad de cinta 081). IEND=3 corresponde a las cadenas de masas A=201 - 251 contenidas en la cinta U00337 (unidad de cinta 080). Los mensajes de error son los mismos que los de la subrutina SKPLLR descritos en la sección VII-2.

4. LOADLD, LOADLR, LOADG (IA, IZ, KOUT, INREC, KOUTRC). Estas subrutinas pueden ser utilizadas para manejar el archivo almacenado en disco. Leen los elementos del vector correspondientes a un isótopo y los almacenan en memoria rápida ("core memory"). Las dos primeras subrutinas son para niveles poblados en desintegraciones radioactivas y reacciones nucleares respectivamente, la última es para radiaciones que desexcitan niveles poblados en desintegraciones radioactivas. Estas subrutinas deben ser utilizadas después de que el archivo de vectores ha sido transferido a memoria de disco (después de haber sido utilizadas las subrutinas SKPLLD, SKPLLR, o SKPLG).

LOADLD y LOADLR tienen el vector de niveles L en el COMMON:

COMMON/LARRAY/L

LOADLG tiene el vector de desexcitaciones D en el COMMON:

COMMON/DARRAY/D

IA (parámetro de entrada)	- Número de masa A
IZ (parámetro de entrada)	- Número atómico Z
KOUT(parámetro de salida)	- Cantidad de registros transferi <u>dos</u> a memoria rápida.

INREC(parámetro de entrada) - Número del registro a partir del cual se desea transferir la información a memoria rápida. Este parámetro es utilizado por la subrutina únicamente si $IA=0$ y/o $IZ=0$. El valor que se le debe dar al mismo debe ser tal que posicione el archivo siempre a partir del primer registro correspondiente al vector del isótopo deseado. Esto es necesario dado que la subrutina transfiere el vector completo a memoria rápida. Si $IA \neq 0$ e $IZ \neq 0$ la subrutina ignora el parámetro INREC.

KOUTRC(parámetro de salida) - Número del último registro del archivo transferido a memoria rápida.

Mensajes de error.

"ERROR. COULD NOT FIND RECORD FOR WHICH $L(1,2)=IZ$."

El isótopo requerido no está en el archivo.

"ERROR WHEN READING $L(1,K)$, $K=1,4$ ". El sistema ha detectado un error en la lectura de los cuatro primeros elementos del vector L.

"ERROR WHEN READING D(1,K), K=1,3". El mismo error que el anterior para la subrutina LOADG de radiaciones.

"ERROR. L(1,1). NE. IA. L(1,1)=X, IA= Y". El vector leído no corresponde al número de masa A deseado. Esto ocurre generalmente cuando las subrutinas SKPLLD, SKPLLR, o SKPLG han sido utilizadas incorrectamente.

"ERROR. D(1,1). NE. IA. D(1,1)=X, IA=Y". El mismo error que el anterior para la subrutina LOADG de radiaciones.

"ERROR WHEN READING L(K,I), I=1,40 (58)" El sistema ha detectado errores en la lectura de estos elementos del vector.

"ERROR WHEN READING D(K,I), I=1,39". EL mismo error que el anterior para la subrutina LOADG de radiaciones.

"ERROR. INREC. LE. O. THIS IS NOT POSSIBLE". El valor del parámetro INREC es cero o negativo. Esto no es posible pues este parámetro es la "variable asociada" de la subrutina de sistema DEFINE FILE con la cual se manejan los registros del disco. Esta variable contiene el número de registro considerado y por lo tanto debe ser siempre positiva.

El último registro del archivo de vectores almacenado en el disco contiene ceros. Por lo tanto, al llegar a este registro las subrutinas LOAD detectan $L(1,1)=0$ (o $D(1,1)=0$), devuelven KOUT=0 y no transfieren ninguna información a memoria rápida.

En el caso de los archivos de radiaciones emitidas en la desexcitación de niveles, si el isótopo considerado no emite ninguna radiación $D(1,3)=0$. La subrutina no transfiere ninguna información a memoria rápida y devuelve $KOUT=0$.

En la figura 4 se muestra la organización de tarjetas para recopilar y ejecutar un programa que utiliza la base de datos. Las unidades de entrada 1, 2, y 3 han sido asignadas al disco (EICREA), y las cintas magnéticas U00334 y U00337 respectivamente.

Ejemplo 1. Se desea transferir de cinta magnética a disco el archivo de niveles (poblados en desintegración radioactiva) correspondiente a $A=37$ y almacenar en memoria rápida el vector correspondiente a $Z=18$. (Se supone que la cinta está posicionada para leer el archivo correspondiente a $A=1$).

CALL SKPLLD (36, 1, KOUT)

CALL LOADLD (37, 18, IOU, INREC, KOUTRC)

La primera instrucción hace "saltar" 36 archivos en la cinta magnética y transfiere el archivo número 37 a memoria de disco.

La segunda instrucción transfiere el vector correspondiente a $Z=18$ a memoria rápida. Los valores de los parámetros de salida en este caso son:

KOUT = 29 (registros de 117 caracteres)

IOU = 6 (registros de 117 caracteres)

KOUTRC = 13 (número del último registro transferido a memoria rápida)

El parámetro INREC no es utilizado por LOADLD pues IA e IZ son distintos de cero.

Ejemplo 2. Se desea transferir de cinta magnética a disco el archivo de vectores de niveles poblados en reacciones nucleares correspondientes a A=33. Almacenar en memoria rápida el vector correspondiente a Z=16. Transferir de cinta magnética a disco el archivo correspondiente a A=54 y guardar en memoria rápida el primer vector del archivo.

```
CALL SKPLLR (32, 1, KOUT)
CALL LOADLR (33, 16, IOUT, INREC, KOUTRC)
CALL SKPLLR (20, 1, KOUT)
CALL LOADLR (0,0, IOUT, 1, KOUTRC)
```

La primera instrucción hace "saltar" 32 archivos en la cinta magnética, almacena el archivo número 33 en disco, y deja posicionada la cinta después de la marca de fin-de-archivo \$\$\$\$ número 33. La cinta está lista para ser leído (o "saltado") el archivo número 34. La segunda instrucción almacena el vector correspondiente a Z=16 en memoria rápida. La tercera instrucción hace "saltar" 20 archivos más en la cinta magnética a partir del archivo número 34 y transfiere a disco el archivo correspondiente a A=54. La cuarta instrucción almacena el primer vector del archivo A=54 en memoria rápida.

VIII. SUBROUTINAS UTILES PARA EL MANEJO DE CARACTERES.

Las subrutinas siguientes sirven para el manejo de caracteres (o cadenas de caracteres) y son de suma utilidad en los programas de búsqueda y extracción selectiva de datos. Un listado de las mismas está dado en el Apéndice II.

1. DECODE (IW, N, LIST) Esta subrutina "desempaqueta" una cadena de N caracteres contenida en el vector unidimensional IW(100) y almacena cada uno de los caracteres en un elemento del vector unidimensional LIST(100). Estos dos vectores son enteros con elementos de media palabra IBM(2 "bytes") cada uno.

IW (parámetro de entrada) - vector que contiene la cadena de caracteres de entrada.

N (parámetro de entrada) - Número de caracteres contenido en IW.

LIST(parámetro de salida) - Vector que contiene los caracteres de IW pero almacenados uno en cada elemento.

Ejemplo 1. Se desea separar la cadena de caracteres "CENTRO ATOMICO". En este caso IW(1)=CE, IW(2)=NT, IW(3)=RO, IW(4)=~~SA~~ IW(5)=TO, IW(6)=MI, IW(7)=CO. En este caso N=14 (número de ca racteres)

```
INTEGER*2 IW (100), LIST (100)
```

```
CALL DECODE (IW, 14, LIST)
```

La subrutina devuelve el vector LIST siguiente:

LIST(1)=C~~6~~, LIST(2)=E~~6~~, LIST(3)=M~~6~~, LIST(4)=T~~6~~, LIST(5)=R~~6~~,
LIST(6)=O~~6~~, LIST(7)=~~6~~, LIST(8)=A~~6~~, LIST(9)=T~~6~~, LIST(10)=O~~6~~,
LIST(11)=M~~6~~, LIST(12)=L~~6~~, LIST(13)=C~~6~~, LIST(14)=O~~6~~.

Ejemplo 2. Se desea saber si la cadena de caracteres dada en el ejemplo 1 contiene la letra Z por lo menos una vez.

```
INTEGER*2 Z, IW(100), LIST(100)
DATA Z/'Zb'/'
IFLAG = 0
CALL DECODE (IW, 14, LIST)
DO 10 I=1,14
  IF (LIST(I).EQ.Z) IFLAG=1
10 CONTINUE
```

Si IFLAG=1 la cadena contiene la letra Z por lo menos una vez
Si IFLAG=0, no.

2. ENCODE (IW, N, LIST) Esta subrutina efectúa la "operación inversa" a DECODE, o sea, "empaqueta" N caracteres contenidos (uno en cada elemento) en el vector IW(100) y los almacena (de a dos caracteres por elemento) en el vector LIST (100). El significado de los parámetros es el mismo que el de la subrutina DECODE dado en la sección VIII-1.

Ejemplo. El vector IW contiene los siguientes caracteres:
IW(1)= R6, IW(2)= E6, IW(3)=A6, IW(4)=C6, IW(5)=T6, IW(6)=O6,
IW(7)= R6.

Se desea almacenar estos caracteres de a dos por elemento en el vector LIST.

```
INTEGER*2 IW(100), LIST(100)
```

```
CALL ENCODE (IW, 7, LIST)
```

El vector LIST contiene: LIST(1)= RE, LIST(2)=AC, LIST(3)=TO
LIST(4)=R6.

3. TRREAL (IA, RIA) Esta subrutina transforma dentro de memoria rápida ("core memory") un número de hasta 10 dígitos (incluyendo el punto decimal) de formato alfanumérico (A) a formato real ("floating point"). TRREAL utiliza la subrutina REAL (escrita en lenguaje ASSEMBLER).

IA(5) (parámetro de entrada) - Vector unidimensional entero de media palabra IBM(2"bytes") que pueden contener hasta 10 dígitos (2 por elemento).

RIA (parámetro de salida) - Contiene el valor numérico de IA en "punto flotante". Esta variable es real y de doble precisión (REAL*8)

Ejemplo. Los vectores IA(5) e IB(5) contienen los siguientes caracteres numéricos:

IA(1)=78, IA(2)=5., IA(3)=32, IA(4)=56, IA(5)=40

IB(1)=58, IB(2)=2., IB(3)=63, IB(4)=23, IB(5)=20

Se desea sumar estos números.

```
INTEGER*2 IA(5), IB(5)
```

```
REAL*8 RIA, RIB, SUM
```

```
CALL TRREAL (IA, RIA)
```

```
CALL TRREAL (IB, RIB)
```

```
SUM=RIA+RIB
```

El vector IA contiene los caracteres 785.325640. El vector IB contiene los caracteres 582.632320. Las variables RIA y RIB contienen los números mencionados. La variable SUM contiene el número real 1367.957960.

TRREAL es indispensable para efectuar operaciones aritméticas con los números almacenados en formato alfanumérico en la base de datos.

IX. EJEMPLO DE USO DE LA BASE DE DATOS: NUMERO DE NIVELES NUCLEARES POBLADOS EN DESINTEGRACIONES RADIOACTIVAS Y REACCIONES NUCLEARES. ASIGNACION DE "SPINES"

Resulta de utilidad conocer el estado actual de ciertas propiedades físicas (o químicas) de todos los isótopos. Esta información puede ser utilizada con fines puramente

científicos (por ejemplo, para determinar la validez de modelos nucleares utilizados en ciertas zonas de la tabla de elementos) o para la administración de fondos de apoyo a proyectos científicos (o técnicos) (por ejemplo, para subsidiar proyectos de investigación sobre ciertas propiedades físicas o químicas poco estudiadas de ciertos elementos, ya sea con fines científicos básicos o aplicados).

En el ejemplo descrito en esta sección se desea obtener en función del número de masa A la siguiente información:

1. El número total de niveles energéticos.
2. El número de niveles con una asignación definitiva del spin.
3. El número de niveles con una asignación incierta del spin.
4. El número de niveles sin asignación de spin.
5. El número de niveles con una vida media determinada experimentalmente.

El programa de búsqueda y extracción de estos datos es simple y está dado en el Apéndice III. Consta de un programa principal y dos subrutinas: SPINC y DECODE. Además son utilizadas las subrutinas de manejo de archivos mencionadas en secciones anteriores.

El número total de niveles para cada número de masa A está de terminado en el programa principal sumando los valores de los elementos $L(1,3)$ de los vectores correspondientes a cada isótopo.

La subrutina SPINC determina:

1. El número de niveles con asignación definitiva de spin. Esto es hecho verificando que los elementos $(L(K,I), I=6,20)$ no contengan los caracteres $(_ o _)$. Estos caracteres aparecen únicamente en los casos de asignaciones inciertas. Por ejemplo: $(1/2+)$ o $3/2,5/2-$.

2. El número de niveles con asignación incierta de spin. Esto es hecho verificando que los mencionados elementos de los vectores contengan los caracteres $(_ o _)$.

3. El número de niveles sin asignación de spin. Esto es hecho verificando que los mencionados elementos de los vectores no contengan ningún carácter, sólo "blancos".

4. El número de niveles con vidas medias determinadas experimentalmente. Esto es hecho verificando que los elementos $(L(K,I), I=21,25)$ no contengan caracteres "blancos".

Los resultados están mostrados en las figuras 5 y 6 para los casos de desintegraciones radioactivas y reacciones nucleares respectivamente. El número de niveles con vidas medias determinadas experimentalmente en función del número de masa A no ha sido incluido en las figuras por razones de claridad. Una muestra de los resultados impresos está graficada en la Figura 7.

AGRADECIMIENTOS

Al Señor Neil Callwood por el desarrollo de las subrutinas ENCODE y DECODE. A los Señores Ricardo Germi y Luis Angel Goro Guerra por la cuidadosa operación de la computadora. En especial a este último, por las largas horas de operación durante los fines de semana. Al Señor Horacio Cingolani, jefe del Centro de Computo, por haber puesto a nuestra disposición lo necesario para la implementación de esta base de datos. Especial reconocimiento va por haber puesto a punto las unidades de cinta después de períodos intermitentes de mal funcionamiento.

REFERENCIAS

1. Table of Isotopes, 7th edition: C.M. Lederer and V.S. Shirley, editors; E. Browne, J.M. Dairiki, and R.E. Doebler, principal authors; A.A. Shihab-Eldin, L.J. Jardine, J.K. Tuli, and A.B. Buyrn, authors, John Wiley and Sons, Inc., New York (1978).
2. "Nuclear Structure Data base", R.B. Firestone and E.Browne, LBL - 11089 (1980).

APENDICE I

Listado de subrutinas para manejo de archivos. (Ver sección VII).

1. Desintegración radioactiva.

```

SUBROUTINE SKPLLD(IFS,IFL,KOUT)
  INTEGER * 2 LA(57),*1
  LOGICAL * 1 LL(11),IA(4),IB(117),IC(3)
  EQUIVALENCE (IB(1),LA(1)),(IB(115),IC(1))
  EQUIVALENCE (IA(1),IAA)
  EQUIVALENCE (LL(1),IAA)
  DIMENSION L(4)
  DATA IPESO/'$$$$$'/
  IF (IFL.EQ.0.OR.IFL.EQ.1) GO TO 3
  GO TO 260
3  ICOJNT = 0
  KOUT = 0
  KL1=1
  IF (IFS.EQ.0) GO TO 40
  DO 10 I= 1, 100000
  READ(2,END=990) IA
  IF (IAA.EQ.IPEO) GO TO 5
  GO TO 10
5  ICOJNT = ICOJNT + 1
  IF (ICOJNT.EQ.IFS) GO TO 40
10 CONTINUE
  PRINT 350
  GO TO 1000
40 IF (IFL.EQ.0) GO TO 1000
45 READ(2,END=990) LL
  IF (IAA.EQ.IPEO) GO TO 980
  CALL TRAC(LL,L(1),L(2),L(3),L(4))
  IF (L(4).EQ.0) L(4)=1
  WRITE(1*KL1,100) L
  KL1=KL1+1
  IF (L(1).EQ.0) GO TO 950
  KMAX1=L(4)
  KMAX2=L(3)
  KOUT=L(3)+L(4)+1+KOUT
  DO 150 I3=1,KMAX1
  READ(2,END=990) IB
  WRITE(1*KL1,200) (LA(K),K=1,40)
  KL1=KL1+1
150 CONTINUE
  DO 151 I4=1,KMAX2
  READ(2,END=990) IB
  CALL TRAC(IC,M)
  M1=M
  WRITE(1*KL1,300) LA,*1
  KL1=KL1+1
151 CONTINUE
  GO TO 45
950 READ(2,END=990) IA
  IF (IAA.EQ.IPEO) GO TO 1000

```

```

      GO TO 950
980  L(1) = 0
      L(2) = 0
      L(3) = 0
      L(4) = 0
      WRITE(1*KL1,100) L
      GO TO 1000
990  PRINT 250
      GO TO 1000
260  WRITE(6,270) IFL
100  FORMAT(3I3,12)
200  FORMAT(40A2)
250  FORMAT('END OF INFORMATION ON TAPE')
270  FORMAT('IFL SHOULD BE I OR C. IFL=',I4)
300  FORMAT(57A2,I3)
350  FORMAT('AFTER 100000 ATTEMPTS IT COULD NOT FIND AN END-CF-FILE MAR
      *K $$$$')
1000 RETURN
      END
      *****
SUBROUTINE SKPLG(IFS,IFL,KOUT,IEND)
      INTEGER*2 LA(38), M1
      LOGICAL*1 LL(11), IA(4), IB(117), IC(3), IBB(117)
      EQUIVALENCE (IB(1),LA(1)), (IB(77),IC(1))
      EQUIVALENCE (IA(1),IAA)
      EQUIVALENCE (LL(1),IAA)
      DIMENSION L(4)
      DATA IPES0/'$$$$'/
      IF(IFL.EQ.0.OR.IFL.EQ.1) GO TO 3
      GO TO 260
3    ICOJNT = 0
      KOJT = 0
      KL1 = 1
      IF(IEND.EQ.2) GO TO 58
      IF(IEND.EQ.3) GO TO 57
      GO TO 960
57   IEND = 3
58   IF(IFS.EQ.0) GO TO 40
      DO 10 I=1,100000
      READ(IEND,END=970) IA
      IF(IAA.EQ.IPES0) GO TO 5
      GO TO 10
5    ICOJNT = ICOJNT + 1
      IF(ICOJNT.EQ.IFS) GO TO 40
10   CONTINUE
      PRINT 350
      GO TO 1000
40   IF(IFL.EQ.0) GO TO 1000
45   READ(IEND,END=970) LL
      IF(IAA.EQ.IPES0) GO TO 990
      CALL TRAC(LL,L(1),L(2),L(3),L(4))
      WRITE(1*KL1,100) L
      KL1 = KL1 + 1
      IF(L(1).EQ.0) GO TO 950
      KOUT = L(3) + 1 + KOUT
      IF(L(3).EQ.0) GO TO 45
      KMAX2=L(3)

```

```
CO 151 I4=1,KMAX2
READ( IEND,END=97C) I88
CO 111 K= 1, 24
111 IP(K) = IEB(K)
CO 121 K= 26, 39
121 IB(K-1) = I88(K)
CO 131 K= 41, 68
131 IR(K-2) = I88(K)
CO 141 K= 70, 82
141 IB(K-3) = I88(K)
CALL TRAC1( IC,M)
M1 = M
WRITE(1'KL1,300) LA,M1
KL1 = KL1 + 1
151 CONTINUE
GO TO 45
950 READ( IEND,END=97C) IA
IF( IAA.EQ.IPES0) GO TO 1000
GO TO 950
960 WRITE(6,155) IEND
GO TO 1000
970 IF( IEND.EQ.3) GO TO 980
IEND = 3
GO TO 58
980 WRITE(6,158) IEND
990 L(1) = 0
L(2)=0
L(3)=0
L(4)=0
WRITE(1'KL1,100) L
GO TO 1000
260 WRITE(6,270) IFL
100 FORMAT(3I3,I2)
155 FORMAT('ERROR. IEND SHOULD BE 2 OR 3. IEND=',I3)
158 FORMAT('END OF INFORMATION ON FILE ',I1)
270 FORMAT(' IFL SHOULD BE 1 OR 0. IFL=',I4)
300 FORMAT(38A2,I3)
350 FORMAT('AFTER 100000 TRIES IT COULD NOT FIND AN END-OF-FILE MARK
* $$$$ ')
1000 RETURN
END
```

```

SUBROUTINE LOADLD(IA,IZ,KOUT,INREC,KOUTRC)
  INTEGER*2 L(150,58)
  COMMON/LARRAY/L
  IFLAG = 0
  KOUT = 0
  KOUTRC = 0
  IF(IA.EQ.0.OR.IZ.EQ.0) GO TO 25
  IDREC = 1
10 READ(1,IDREC,1000,ERR=40) L(1,1), L(1,2), L(1,3), L(1,4)
  IDREC = IDREC + 1
  IF(L(1,1).EQ.0) GO TO 18
  IF(IFLAG.EQ.1) GO TO 13
  IF(L(1,1).NE.1A) GO TO 50
  IF(L(1,2).NE.IZ) GO TO 20
13 KMAXP = L(1,4) + 1
  KMAXP1 = KMAXP + 1
  KOUT = KMAXP + L(1,3)
  DO 15 K = 2, KMAXP
    READ(1,IDREC,1010,ERR=70) (L(K,I),I=1,40)
15 IDREC = IDREC + 1
    DO 17 K = KMAXP1, KOUT
      READ(1,IDREC,1020,ERR=80) (L(K,I),I=1,58)
17 IDREC = IDREC + 1
18 KOUTRC = IDREC - 1
  IF(1A.NE.0.AND.IZ.NE.0) GO TO 30
  GO TO 100
20 IDREC = IDREC + L(1,3) + L(1,4)
  GO TO 10
25 IFLAG = 1
  IF(INREC.LE.0) GO TO 65
  IDREC = INREC
  GO TO 10
30 PRINT 2000
  GO TO 100
40 PRINT 2010
  GO TO 100
50 WRITE(6,2020) L(1,1), 1A
  GO TO 100
65 PRINT 2060
  GO TO 100
70 PRINT 2040
  GO TO 100
80 PRINT 2050
1000 FORMAT(3I3,I2)
1010 FORMAT(39A2,A1)
1020 FORMAT(57A2,I3)
2000 FORMAT('ERROR. COULD NOT FIND RECORD FOR WHICH L(1,2)=IZ')
2010 FORMAT('ERROR WHEN READING L(1,K),K=1,4')
2020 FORMAT('ERROR. L(1,1).NE.1A. L(1,1)='',I3,2X,'1A='',I3)
2040 FORMAT('ERROR WHEN READING L(K,I),I=1,40')
2050 FORMAT('ERROR WHEN READING L(K,I),I=1,58')
2060 FORMAT('ERROR. INREC. LE. 0. THIS IS NOT POSSIBLE')
100 RETURN
  END

```

```
SUBROUTINE LOADG(IA,IZ,KOUT,INREC,KOUTRC)
  INTEGER*2 D(400,39)
  COMMON/CARRAY/D
  IFLAG = 0
  KOUT = 0
  KOUTRC = 0
  IF(IA.EQ.0.OR.IZ.EQ.0) GO TO 25
  ICREC = 1
10 READ(1,ICREC,1000,ERR=40) D(1,1), D(1,2), D(1,3)
  ICREC = ICREC + 1
  IF(D(1,1).EQ.0.OR.D(1,3).EQ.0) GO TO 18
  IF(IFLAG.EQ.1) GO TO 13
  IF(D(1,1).NE.1A) GO TO 50
  IF(D(1,2).NE.IZ) GO TO 20
13 KOUT = D(1,3) + 1
  DO 15 K=2, KOUT
  READ(1,ICREC,1020,ERR=70) (D(K,I),I=1,39)
15 ICREC = ICREC + 1
18 KOUTRC = ICREC - 1
  IF(1A.NE.0.AND.IZ.NE.0) GO TO 30
  GO TO 100
20 ICREC = ICREC + D(1,3)
  GO TO 10
25 IFLAG = 1
  IF(INREC.LE.0) GO TO 65
  ICREC = INREC
  GO TO 10
30 PRINT 2000
  GO TO 100
40 PRINT 2010
  GO TO 100
50 WRITE(6,2020) D(1,1), 1A
  GO TO 100
65 PRINT 2060
  GO TO 100
70 PRINT 2040
  GO TO 100
1000 FORMAT(3I3)
1020 FORMAT(38A2,13)
2000 FORMAT('ERROR. COULD NOT FIND RECORD FOR WHICH L(1,2)=IZ')
2010 FORMAT('ERROR WHEN READING D(1,K),K=1,3')
2020 FORMAT('ERROR. D(1,1).NE.1A. D(1,1)=' ,I3,2> , '1A=' ,I3)
2040 FORMAT('ERROR WHEN READING D(K,I),I=1,39')
2060 FORMAT('ERROR. INREC. LE. 0. THIS IS NOT POSSIBLE')
100 RETURN
END
```

2. Reacciones nucleares.

```

SUBROUTINE SKPLLR( IFS, IFL, KOUT, IEND)
  INTEGER*2 LA(55), M1
  LOGICAL*1 LL(11), IA(4), IB(117), IC(3)
  EQUIVALENCE (IB(1),LA(1)), (IB(111),IC(1))
  EQUIVALENCE (IA(1),IAA)
  EQUIVALENCE (LL(1),IAA)
  DIMENSION L(4)
  DATA IPESO/'$$$$'/
  IF(IFL.EQ.0.OR.IFL.EQ.1) GO TO 3
  GO TO 260
3  ICOUNT = 0
  KOUT = 0
  KL1 = 1
  IF(IEND.EQ.2) GO TO 58
  IF(IEND.EQ.3) GO TO 57
  GO TO 960
57 IEND = 3
58 IF(IFS.EQ.0) GO TO 40
  1  DO 10 I=1, 100000
    READ(IEND,END=970) IA
    IF(IAA.EQ.IPEO) GO TO 5
    GO TO 10
  5  ICOUNT = ICOUNT + 1
    IF(ICOUNT.EQ.IFS) GO TO 40
10  CONTINUE
    PRINT 350
    GO TO 1000
40  IF(IFL.EQ.0) GO TO 1000
45  READ(IEND,END=970) LL
    IF(IAA.EQ.IPEO) GO TO 990
    CALL TRAD(LL,L(1),L(2),L(3),L(4))
    WRITE(1*KL1,100) L
    KL1 = KL1 + 1
    IF(L(1).EQ.0) GO TO 990
    KOUT = KOUT + L(3) + L(4) + 1
    KMAX1 = L(4)
    KMAX2 = L(3)
    DO 150 I3 = 1, KMAX1
      READ(IEND,END=970) IB
      WRITE(1*KL1,200) (LA(K),K=1,21)
      KL1 = KL1 + 1
150  CONTINUE
      DO 151 I4 = 1, KMAX2
        READ(IEND,END=970) IB
        CALL TRAD1(IC,M)
        M1 = M
        WRITE(1*KL1,300) LA, M1
        KL1 = KL1 + 1
151  CONTINUE

```

```

GO TO 45
950 READ(IEND,END=970) IA
    IF(IAA.EQ.IPES7) GO TO 1000
    GO TO 950
960 WRITE(6,155) IEND
    GO TO 1000
970 IF(IEND.EQ.3) GO TO 980
    IEND = 3
    GO TO 58
980 WRITE(6,158) IEND
990 L(1) = 0
    L(2) = 0
    L(3) = 0
    L(4) = 0
    WRITE(1*KL1,100) L
    GO TO 1000
260 WRITE(6,270) IFL
100 FORMAT(3I3,I2)
155 FORMAT('ERROR. IEND SHOULD BE 2 OR 3. IEND=',I3)
158 FORMAT('END OF INFORMATION ON FILE',I1)
200 FORMAT(20A2,A1)
270 FORMAT('IFL SHOULD BE 1 OR 0. IFL=',I4)
300 FORMAT(55A2,I3)
350 FORMAT('AFTER 100000 ATTEMPTS IT COULD NOT FIND AN END-OF-FILE MAR
    *K $$$$')
1000 RETURN
END
*****
SUBROUTINE LOADLR(IA,IZ,KOUT,INREC,KOUTRC)
    INTEGER*2 L(200,56)
    COMMON/LARRAY/L
    IFLAG = 0
    KOUT = 0
    KOUTRC = 0
    IF(IA.EQ.0.OR.IZ.EQ.0) GO TO 25
    IDREC = 1
10 READ(1,IDREC,1000,ERR=40) L(1,1), L(1,2), L(1,3), L(1,4)
    IDREC = IDREC + 1
    IF(L(1,1).EQ.0) GO TO 18
    IF(IFLAG.EQ.1) GO TO 13
    IF(L(1,1).NE.IA) GO TO 50
    IF(L(1,2).NE.IZ) GO TO 20
13 KMAXP = L(1,4) + 1
    KMAXP1 = KMAXP + 1
    KOUT = KMAXP + L(1,3)
    DO 15 K=2, KMAXP
    READ(1,IDREC,1010,ERR=70) (L(K,I),I=1,21)
15 IDREC = IDREC + 1
    DO 17 K= KMAXP1, KOUT
    READ(1,IDREC,1020,ERR=80) (L(K,I),I=1,56)
17 IDREC = IDREC + 1
18 KOUTRC = IDREC - 1
    IF(IA.NE.0.AND.IZ.NE.0) GO TO 30
    GO TO 100
20 IDREC = IDREC + L(1,3) + L(1,4)
    GO TO 10
25 IFLAG = 1
    IF(INREC.LE.0) GO TO 65

```

```

IDREC = INREC
GO TO 10
30 PRINT 2000
GO TO 100
40 PRINT 2010
GO TO 100
50 WRITE(6,2020) L(1,1), IA
GO TO 100
65 PRINT 2060
GO TO 100
70 PRINT 2040
GO TO 100
80 PRINT 2050
1000 FORMAT(3I3,I2)
1010 FORMAT(20A2,A1)
1020 FORMAT(55A2,I3)
2000 FORMAT('ERROR. COULD NOT FIND RECORD FOR WHICH L(1,2)=I2')
2010 FORMAT('ERROR WHEN READING L(1,K),K=1,4')
2020 FORMAT('ERROR. L(1,1).NE.IA. L(1,1)='',I3,2X,'IA='',I3)
2040 FORMAT('ERROR WHEN READING L(K,I), I=1,21')
2050 FORMAT('ERROR WHEN READING L(K,I),I=1,56')
2060 FORMAT('ERROR. INREC. LE. C. THIS IS NOT POSSIBLE')
100 RETURN

```

END

1	TRAD	C SECT	1	TRADI	C SECT
2		BALR 8,0	2		BALR 8,0
3		USING *,8	3		USING *,8
4	*		4		L 2,0(1)
5		L 2,C(1)	5		L 9,4(1)
6		L 9,4(1)	6		LA 3,3
7		LA 6,3	7		SR 5,5
8		LA 3,3	8	DJS	IC 4,C(2)
9	UNO	SR 5,5	9		V 4,=F'15'
10	DJS	IC 4,C(2)	10		MH 5,=H'10'
11		N 4,=F'15'	11		AR 5,4
12		MH 5,=H'10'	12		LA 2,1(2)
13		AR 5,4	13		BCT 3,DOS
14		LA 2,1(2)	14		ST 5,C(9)
15		BCT 3,DOS	15		MVI 12(13),X'FF'
16		ST 5,C(9)	16		BCR 15,14
17		LA 9,4(9)	17	LTJRG	
18		LA 3,3	18	DC F'15'	
19		BCT 6,UNO	19	DC H'10'	
20		LA 3,2	20		END
21		SR 5,5			
22	TRES	IC 4,C(2)			
23		N 4,=F'15'			
24		MH 5,=H'10'			
25		AR 5,4			
26		LA 2,1(2)			
27		BCT 3,TRES			
28		ST 5,C(9)			
29	FIN	MVI 12(13),X'FF'			
30		BCR 15,14			
31	LTJRG				
32	DC F'15'				
33	DC H'10'				
34		E'ID			

APENDICE II

Listado de subrutinas útiles para el manejo de caracteres(Ver sección VIII).

```

SUBROUTINE DECODE(IW,N,LIST)
  INTEGER*2 LIST(100), IW(100), IA
  DATA IA/' '/
  DO 5 I=1,100
5 LIST(I) = IA
  M = N/2
  NOOD = (N/2)*2
  IF(NOOD.NE.N) M=(N+1)/2
  K = 1
  DO 10 I = 1, M
  L = IW(I)
  IF(IW(I).LT.0) L=65536 + IW(I)
  LA= L/256 * 256
  LB = (L-LA)*256 + 64
  LA = LA + 64
  IF(LA.GE.32768) LIST(K) = LA - 65536
  IF(LA.LT.32768) LIST(K) = LA
  IF(LB.GE.32768) LIST(K+1) = LB - 65536
  IF(LB.LT.32768) LIST(K+1) = LB
  K = K + 2
10 CONTINUE
  RETURN
  END

```

```

SUBROUTINE ENCODE(IW,N,LIST)
  INTEGER*2 LIST(100), IW(100), IBLANK
  INTEGER*4 LA, LB
  DATA IBLANK/' '/
  K = 1
  DO 10 I=1,N,2
  LA = IW(I)
  IF(IW(I).LT.0) LA = 65536 + IW(I)
  II = I + 1
  LB = IW(II)
  IF(IW(II).LT.0) LB = 65536 + IW(II)
  IF(I.EQ.N) LB = IBLANK
  LA = LA/256*256 + LB/256
  IF(LA.GE.32768) LIST(K) = LA - 65536
  IF(LA.LT.32768) LIST(K)= LA
  K = K + 1
10 CONTINUE
  RETURN
  END

```

```

SUBROUTINE TRREAL(IA,RIA)
  INTEGER*2 IA(5)
  REAL*8 RIA, R1, R2
  CALL REAL(IA,I1,I2,I3)
  R1 = DFLOAT(I1)
  R2= DFLOAT(I2)
  RIA= R1 + (R2/(10.**I3))
  RETURN
END

```

```

1  REAL      C SECT
2              BALR    8,C
3              USING   *,8
4  *
5              L        2,0(1)
6              L        5,4(1)
7              L        10,8(1)
8              L        7,12(1)
9              LA       3,10
10             SR       5,5
11  UNO       IC        4,C(2)
12             CLI     C(2),X'48'
13             BNE     DEC
14             ST       5,C(9)
15             SR       5,5
16             SR       6,6
17             B        DDS
18  DEC       CLI     C(2),X'40'
19             BE      TRES
20             N        4,=F'15'
21             MH       5,=H'10'
22             AR       5,4
23             LA       6,1(6)
24  DDS       LA       2,1(2)
25  TRES      BCT      3,UNO
26             ST       5,C(10)
27             ST       6,C(7)
28             MVI     12(13),X'FF'
29             BCR     15,14
30  L TORG
31  DC  F'15'
32  DC  H'10'
33             END

```

APENDICE III

Listado de un programa de ejemplo de uso de la base de datos. (Ver sección IX).

```
C.....MAIN PROGRAM.
C.....
  INTEGER*2 L(150,58)
  DEFINE FILE 1(600,360,L,IDREC)
  COMMON/LARRAY/L
C.....
C.....PRINT HEADINGS.
C.....
  PRINT 90
  90 FORMAT(/,22X,'NUMBER OF LEVELS POPULATED IN RADIOACTIVE DECAY')
  PRINT 100
  100 FORMAT(/,21X,'A',3X,'UNKNOWN',2X,'UNCERTAIN',2X,'DEFINITE',4X,'KNOWN',
    *'TOTAL',/27X,'SPIN',6X,'SPIN',6X,'SPIN',4X,'HALF-LIFE',2X,
    *'LEVELS')
  PRINT 110
  110 FORMAT(20X,52(' '),/)
C.....
C.....INITIALIZATION OF VARIABLES.
C.....
  ISJM = 0
  ISUM1 = 0
  ISUM2 = 0
  ISJM3 = 0
  ISJM4 = 0
  ISUM5 = 0
  KA = 0
  1 KSJM1 = 0
  KSJM2 = 0
  KSUM3 = 0
  KSJM4 = 0
  KSJM5 = 0
C.....
C.....LOAD NEXT FILE ON TO DISK.
C.....
  IFS = 0
  IFL = 1
  CALL SKPLLD(IFS,IFL,KOUT)
  IF(KOUT.EQ.0) GO TO 30
C.....
C.....LOAD ARRAY FOR FIRST ISOTOPE ON TO CORE MEMORY.
C.....
  INREC = 1
  10 IA = 0
  IZ = 0
  CALL LOADLD(IA,IZ,IOUT,INREC,KOUTRC)
  IF(IOUT.EQ.0) GO TO 20
  KA= L(1,1)
C.....
C.....RETRIEVE LEVEL DATA.
C.....
  CALL SPINC( IBLANK, ISPINU, ISPINC, IHALF)
C.....
C.....IBLANK = NUMBER OF LEVELS WITH NO SPIN ASSIGN.
C.....ISPINU = NUMBER OF LEVELS WITH UNCERTAIN SPIN ASSIGN.
C.....ISPINC = NUMBER OF LEVELS WITH DEFINITE SPIN ASSIGN.
C.....IHALF = NUMBER OF LEVELS WITH MEASURED HALF-LIFE.
```

```

C.....
  KSUM1 = KSUM1 + IBLANK
  KSUM2 = KSUM2 + ISPINU
  KSUM3 = KSUM3 + ISPINC
  KSUM4 = KSUM4 + IHALF
  KSUM5 = KSUM5 + L(1,3)
  INREC = KOUTRC + 1
  GO TO 10
C.....
C.....PRINT INFORMATION.
C.....
  20 WRITE(6,120) (A,KSUM1,KSUM2,KSUM3,KSUM4,KSUM5
120 FORMAT(20X,I3,4X,I3,7X,I3,7X,I3,8X,I3,6X,I3)
  ISUM = ISUM + KSUM1 + KSUM2 + KSUM3
  ISUM1 = ISUM1 + KSUM1
  ISUM2 = ISUM2 + KSUM2
  ISUM3 = ISUM3 + KSUM3
  ISUM4 = ISUM4 + KSUM4
  ISUM5 = ISUM5 + KSUM5
  30 IF(KA.GE.256) GO TO 1000
  GO TO 1
1000 PRINT 110
C.....
C.....PRINT CUMULATIVE INFORMATION FOR ALL MASS CHAINS.
C.....
  WRITE(6,250) ISUM,ISUM1,ISUM2,ISUM3,ISUM4,ISUM5
250 FORMAT(20X,'TOTAL NUMBER OF LEVELS=',I5,/20X,'TOTAL NUMBER OF LEVE
*LS WITH UNKNOWN SPIN=',I5,/20X,'TOTAL NUMBER OF LEVELS WITH UNCERT
*AIN SPIN=',I5,/20X,'TOTAL NUMBER OF LEVELS WITH DEFINITE SPIN =',I
*5,/20X,'TOTAL NUMBER OF LEVELS WITH MEASURED HALF-LIFE=',I5,/20X,'
*TOTAL NUMBER OF LEVELS=',I5,/'END OF JOB')
  STOP
  END

SUBROUTINE SPINC(NBLANK,NSPINU,NSPINC,NHALF)
  INTEGER*2 L(150,58),KBLANK,IW(100),LIST(100),ICOMA,IPAREN
  INTEGER*2 KPROJ, <SPEC,KLMM,KT,IPLUS,IMINUS
  COMMON/LARRAY/L
  DATA KBLANK/' '/
  DATA ICOMA/' '/
  DATA IPAREN/'(' '/
  DATA KPROJ/'K '/
  DATA KLMM/'L '/
  DATA KSPEC/'S '/
  DATA KT/'T '/
  DATA IPLUS/'+' '/
  DATA IMINUS/'-' '/
  NSPINU = 0
  NSPINC = 0
  NBLANK = 0
  NHALF = 0
  KMIN = L(1,4) + 2
  KMAX = L(1,3) + L(1,4) + 1
  2 K = KMIN
C.....
C.....CHECK FOR SPIN ASSIGN.
C.....
  GO TO 1 = 6, 20
  IF(L(K,I).NE.KBLANK) GO TO 20

```

```

10 CONTINUE
   NBLANK = NBLANK + 1
   GO TO 100
20 DO 30 I= 6, 20
30 IW(I-5) = L(K,I)
   N = 30
   CALL DECODE(IW,N,-IST)
   DO 35 KM = 1, 30
   IF(LIST(KM).NE.<BLANK) GO TO 36
35 CONTINUE
   PRINT 300
300 FORMAT('ERROR. SPIN FIELD IS BLANK')
   STOP

C.....
C..... CHECK FOR K, S, T, L, OR PARITY ASSIGN. ONLY.
C.....
36 IF(LIST(KM).EQ.<PROJ.OR.LIST(KM).EQ.KSPEC) GO TO 50
   IF(LIST(KM).EQ.<LMCM.OR.LIST(KM).EQ.KT) GO TO 50
   IF(LIST(KM).EQ.IPLUS.OR.LIST(KM).EQ.IMINUS) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.KPROJ) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.KSPEC) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.KLMCM) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.KT) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.IPLUS) GO TO 50
   IF(LIST(KM).EQ.IPAREN.AND.LIST(KM+1).EQ.IMINUS) GO TO 50

C.....
C.....CHECK FOR UNCERTAIN SPIN ASSIGN.
C.....
   DO 40 KL = KM, 30
   IF(LIST(KL).EQ.ICOM) GO TO 90
   IF(LIST(KL).EQ.IPAREN) GO TO 90
   IF(LIST(KL).EQ.<BLANK) GO TO 45
40 CONTINUE
45 NSPINC = NSPINC + 1
   GO TO 100
50 NBLANK = NBLANK + 1
   GO TO 100
90 NSPINU = NSPINU + 1

C.....
C.....CHECK FOR HAL-LIFE.
C.....
100 DO 105 I=21, 25
   IF(L(K,I).NE.<BLANK) GO TO 107
105 CONTINUE
   GO TO 108
107 NHALF = NHALF + 1
108 IF(K.EQ.KMAX) GO TO 110
   KMIN = KMIN + 1
   GO TO 2
110 KSUM = NBLANK + NSPINU + NSPINC
   IF(KSUM.NE..(1,3)) GO TO 120
   GO TO 1000
120 WRITE(6,200) KSUM, L(1,3)
200 FORMAT('ERROR. <SUM NE L(1,3). KSUM=',I4,'L(1,3)=',I4)
   STOP
1000 RETURN
   END

```

APENDICE IV

Inicialización del disco EICREA y generación del archivo CHARLY.

```
//ESKEMPES JOB
// EXEC UTILS
// INITIAL TYPE=SDSD, DVADR=199, VOL ID='EICREA', VTDC=36
/*
/&
//ESKEMPES JOB
// ALLOC CHARLY, SDSU='EICREA', 1600, FMT
// LABEL 360, RECLN=360, WRCHK
// ALLOC SDSREL, SDSU='EICREA', 1500, 238
// LABEL 360, RECLN=72, WRCHK
/*
//SYSD02 ACCESS SDSREL, SDSU='RES091'
//SYSD03 ACCESS SDSREL, SDSU='EICREA'
// EXEC UTILS
// COPYMEM SIZE=360, SIZEUT=360, INCL=(UNITABN-PDUMP)
/&
```

Figura 1. Estructura de la base de datos en la cinta magnética.
Los vectores correspondientes a cada isótopo están agrupados de acuerdo al número de masa A y almacenados en archivos separados entre sí por la cadena de caracteres \$\$\$\$.

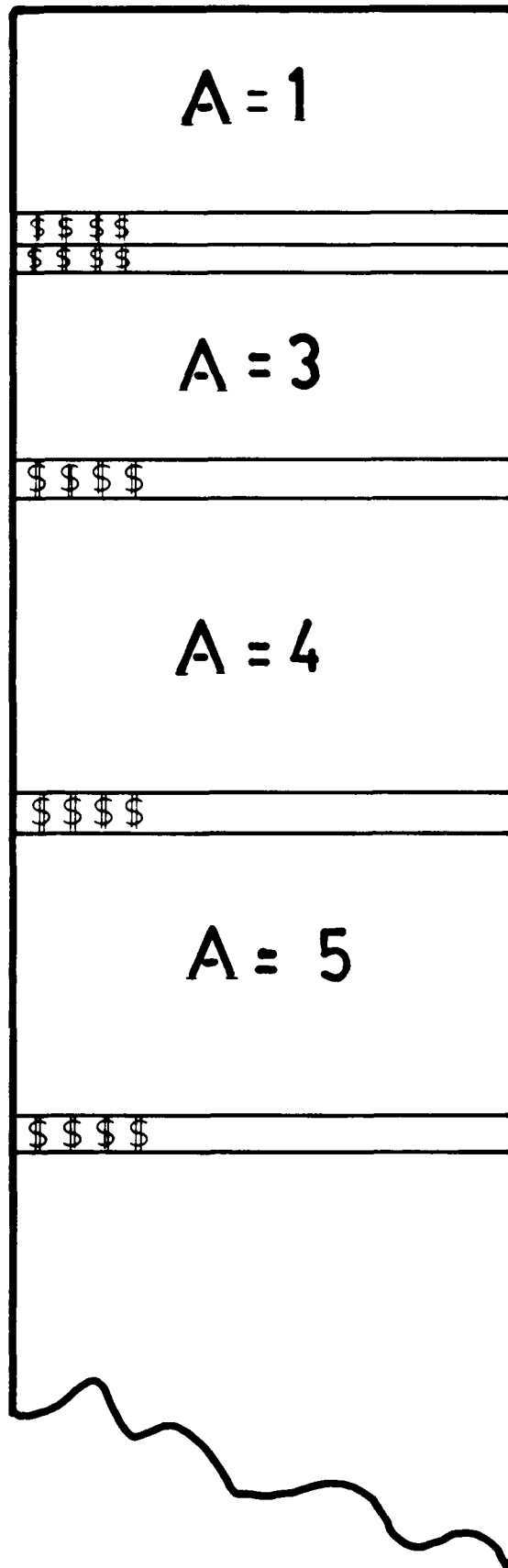


Figura 2. Formatos de los vectores correspondientes a desinte_gración radioactiva y a reacciones nucleares. La definición de las variables está dada en el texto.

a) Vectores L(150,58) de decaimiento radioactivo

<u>FORMATO</u>	<u>ELEMENTOS</u>
3I3,I2	L(1,1)=A, L(1,2)=Z, L(1,3)=NLEV, L(1,4)=NPAR
39A2,A1	L(K,I), K=2, NPAR+1; I=1,40
57A2,I3	L(K,I), K=NPAR+2, NPAR+NLEV+1; I=1,58

b) Vectores L(200,56) de reacciones nucleares.

<u>FORMATO</u>	<u>ELEMENTOS</u>
3I3,I2	L(1,1)=A, L(1,2)=Z, L(1,3)=NLEV, L(1,4)=NREACT
20A2,A1	L(K,I), K=2, NREACT+1; I=1,21
55A2,I3	L(K,I), K=NREACT+2, NREACT+NLEV+1; I=1,56

c) Vectores D(400,39) de decaimiento radioactivo.

<u>FORMATO</u>	<u>ELEMENTOS</u>
3I3	D(1,1)=A, D(1,2)=Z, D(1,3)=NTRANS
38A2,I3	D(K,I), K=2, NTRANS+1; I=1,39.

Figura 3. ORGANIZACION Y MANEJO DE ARCHIVOS.

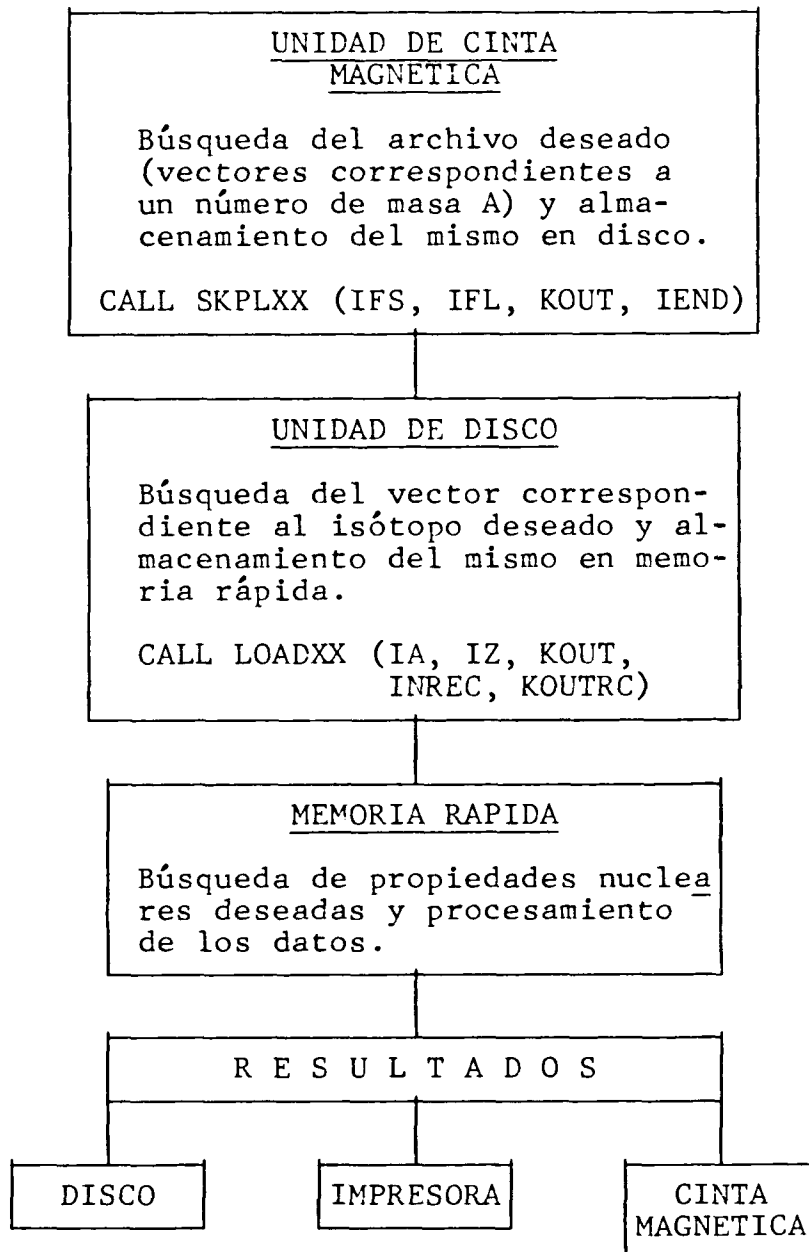
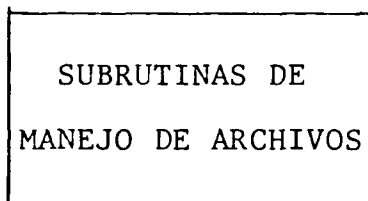
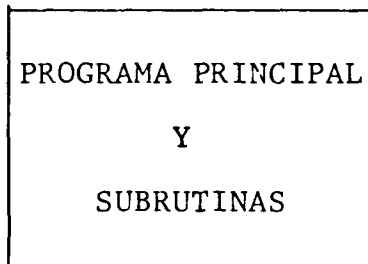


Figura 4. Organización de un "Job". Tarjetas de control.

```
//ESKEMPES JOB
//SYSOPT ACCESS SDSOPT   }
//SYSLST ACCESS SDSLST   } (Anula el control de carro)
//EXEC FORTRAN
```



```
/*
//EXEC LNKEDT
/*
//SYS001 ACCESS CHARLY, SDSD= 'EICREA' (Unidad de disco)
//SYS002 ACCESS TAPE 2, 081= 'U00334'   }
//SYS003 ACCESS TAPE 3, 080= 'U00337'   } (Unidades de cinta)
/*
//EXEC
/*
/&
```

Figura 5a. Histograma de niveles nucleares poblados en desintegraciones radioactivas. Niveles con asignación definitiva de "spin".

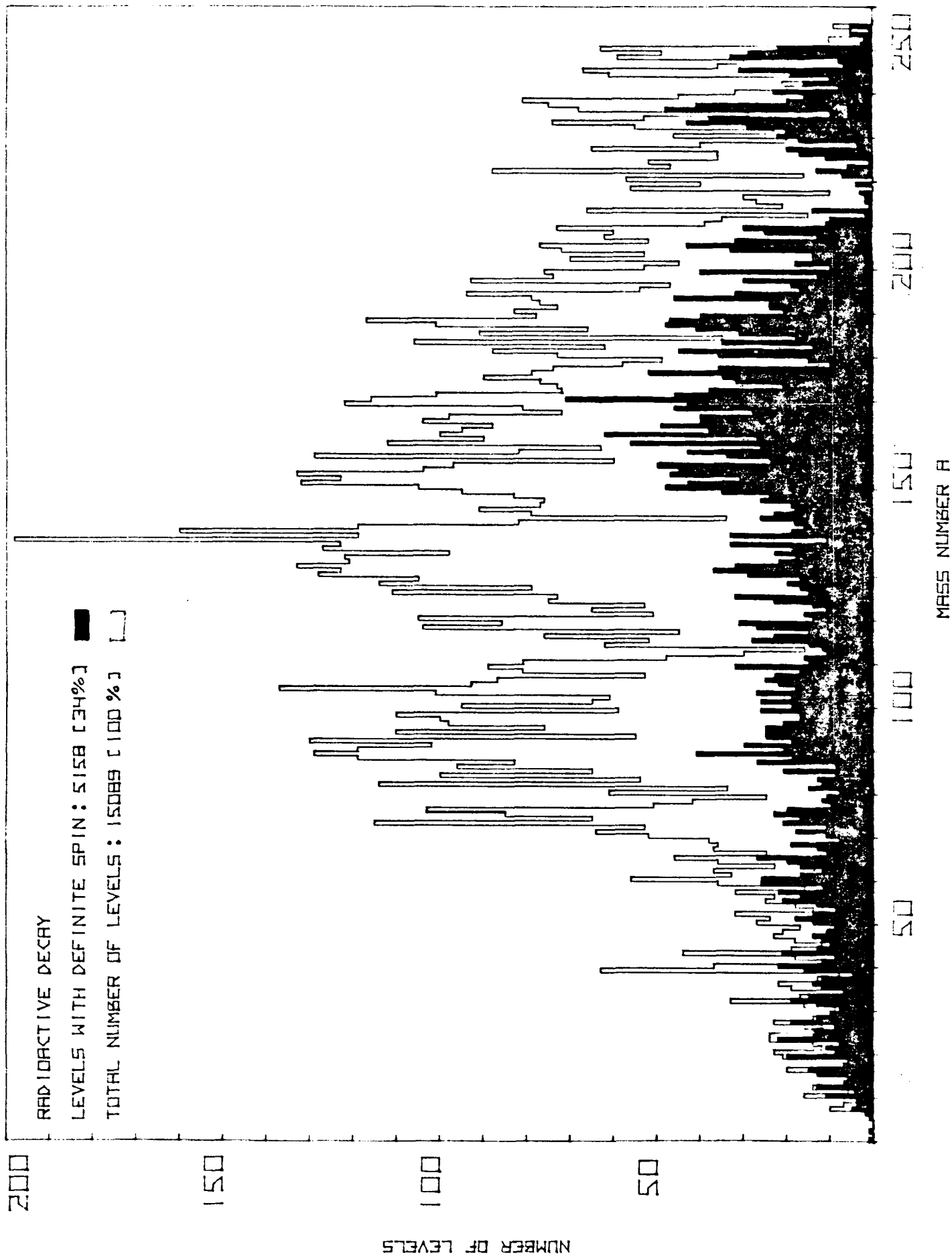


Figura 5b. Histograma de niveles nucleares poblados en desintegraciones radioactivas. Niveles con asignación incierta de "spin".

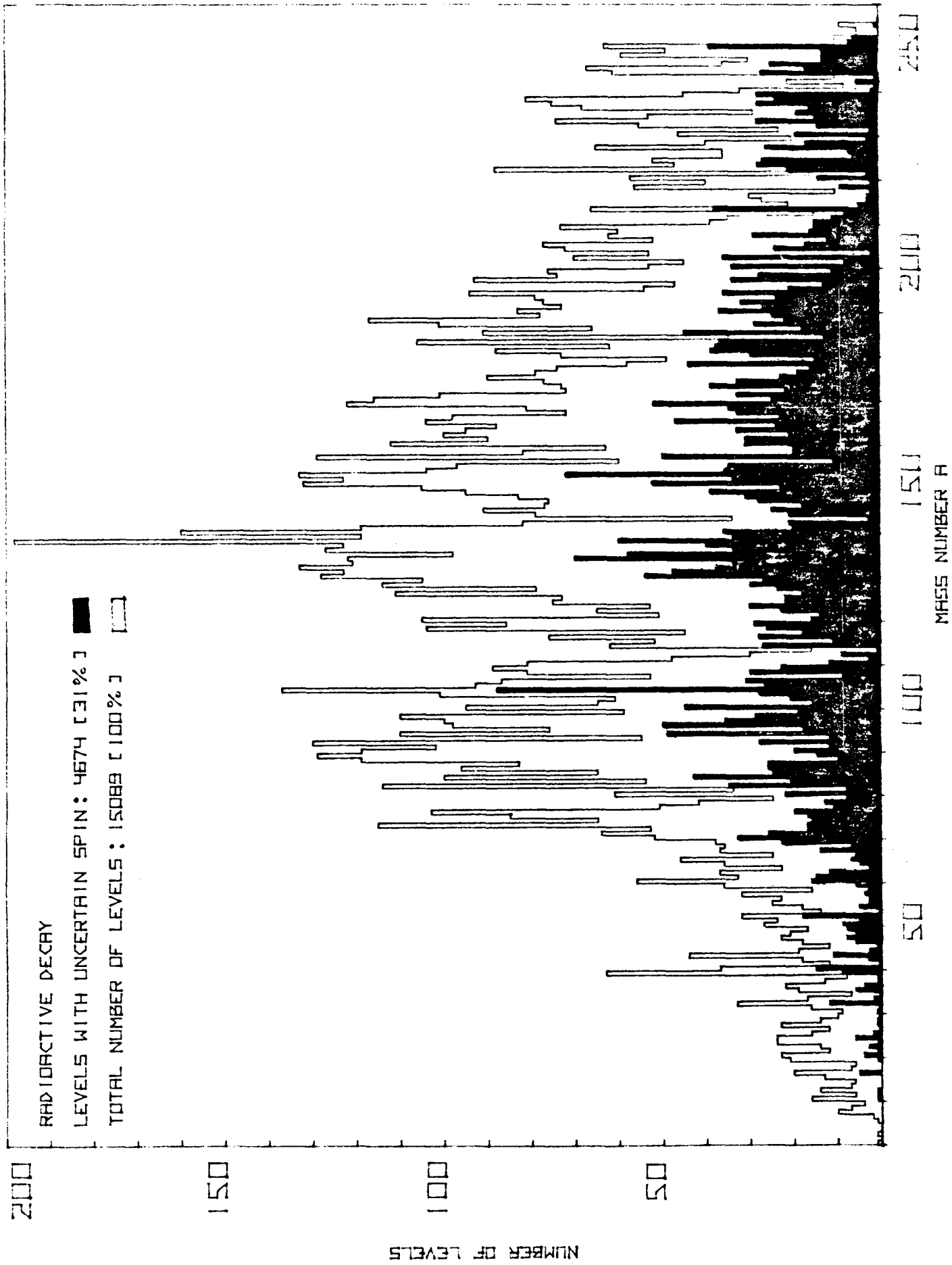


Figura 5c. Histograma de niveles nucleares poblados en desintegraciones radioactivas. Niveles sin asignación de "spin".

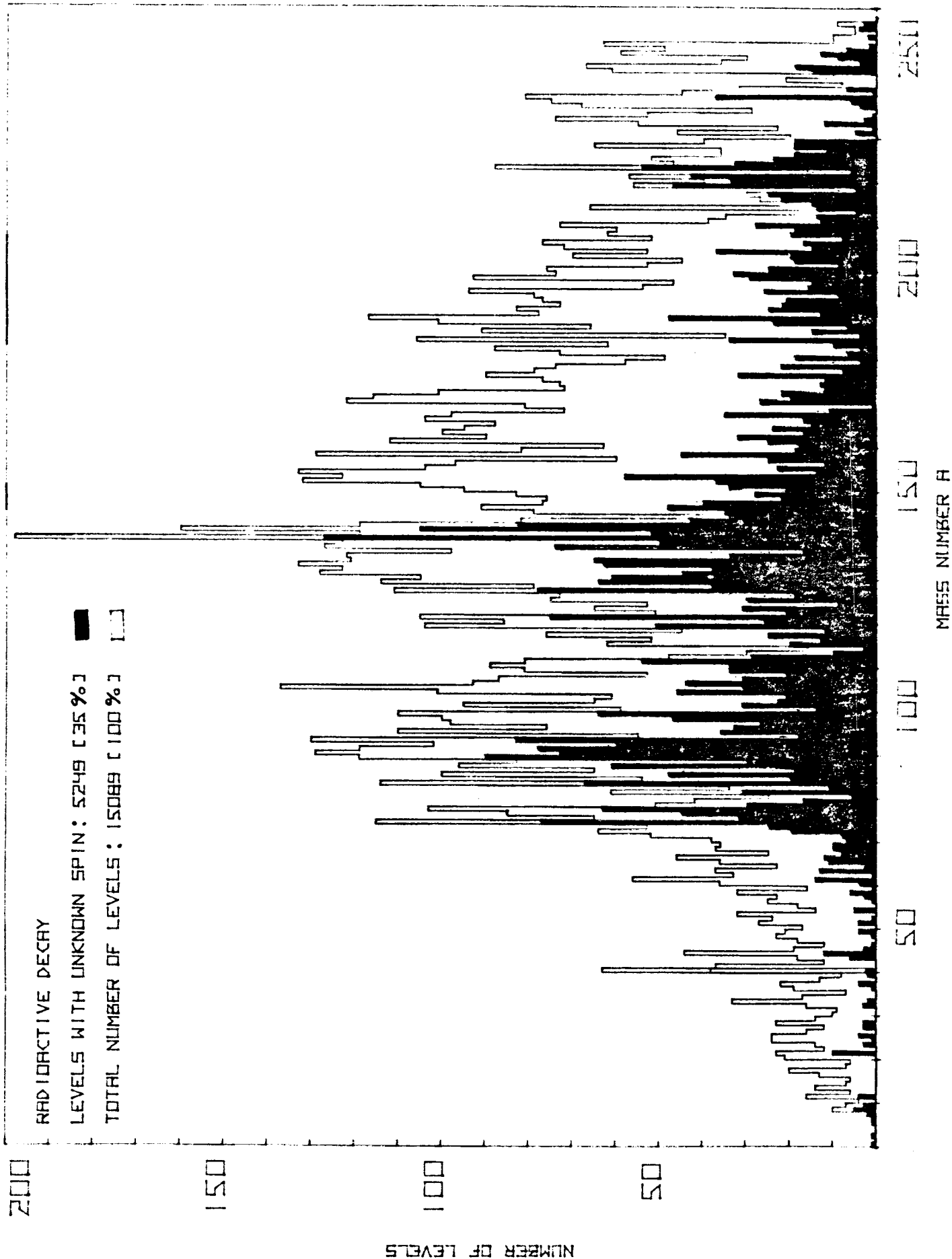


Figura 6a. Histograma de niveles nucleares poblados en reacciones nucleares. Niveles con asignación definitiva de "spin".

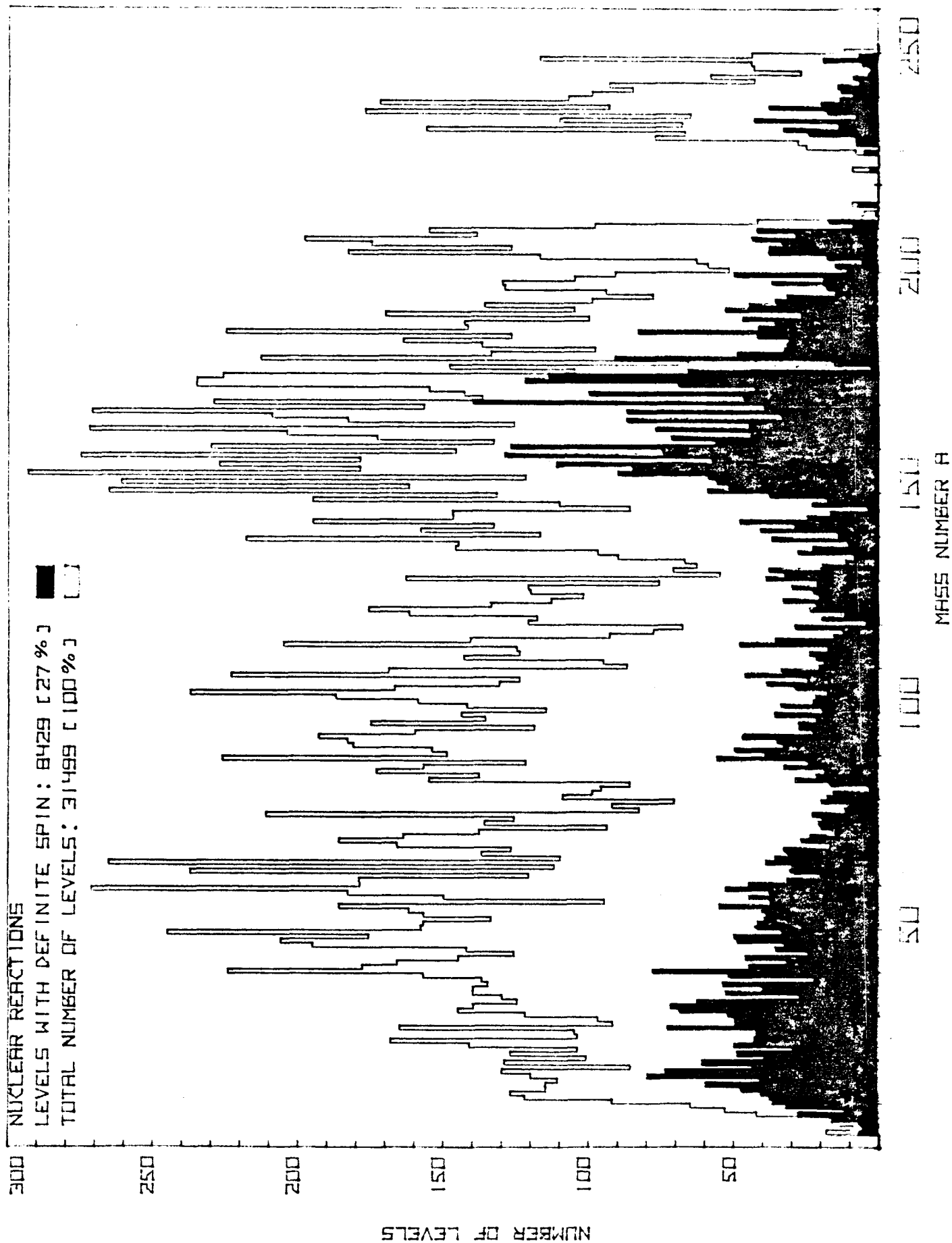


Figura 6b. Histograma de niveles nucleares poblados en reacciones nucleares. Niveles con asignación incierta de "spin".

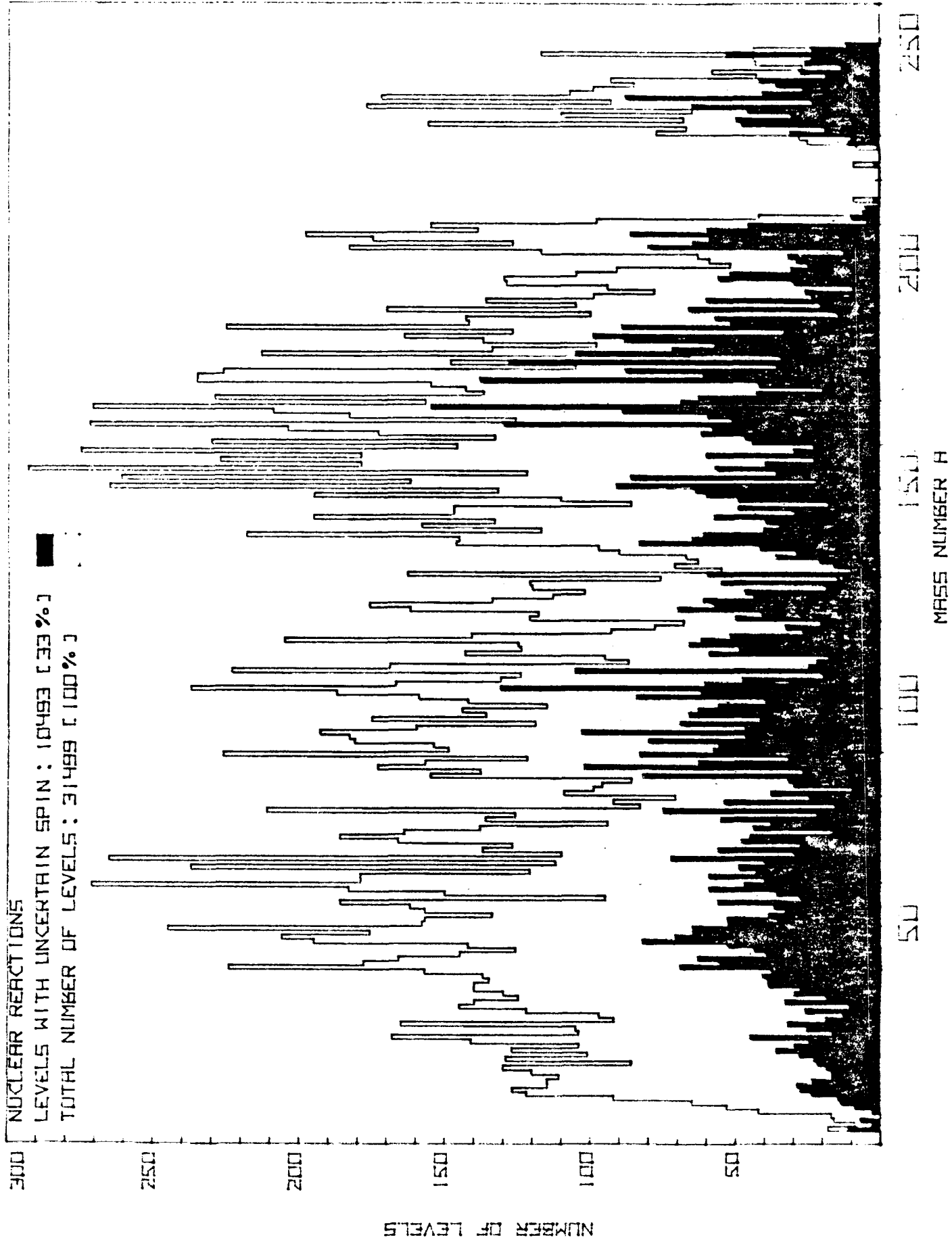
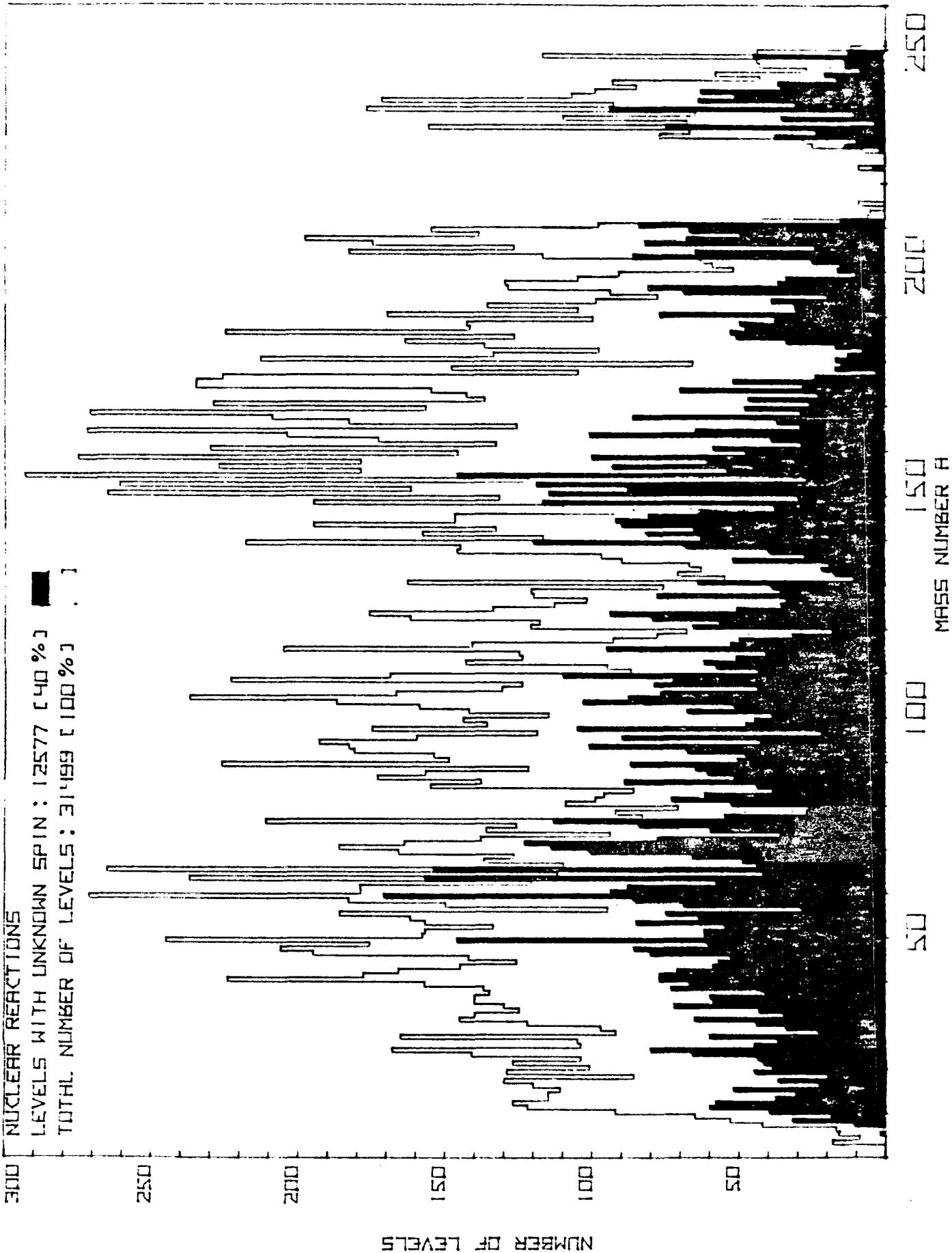


Figura 6c. Histograma de niveles nucleares poblados en reacciones nucleares. Niveles sin asignación de "spin".



NUMBER OF LEVELS POPULATED IN RADIOACTIVE DECAY

A	UNKNOWN SPIN	UNCERTAIN SPIN	DEFINITE SPIN	KNOWN HALF-LIFE	TOTAL LEVELS
*****	*****	*****	*****	*****	*****
1	0	0	1	0	1
3	0	0	1	0	1
6	0	0	1	0	1
7	0	0	2	1	2
8	5	0	5	9	10
9	3	0	4	4	7
10	0	0	4	1	4
11	4	1	11	9	16
12	0	1	5	5	6
13	0	1	13	11	14
14	0	0	7	5	7
15	0	0	6	2	6
16	0	0	13	9	13
17	0	5	15	7	20
18	0	0	7	6	7
19	0	0	6	4	6
20	0	1	20	11	21
21	10	4	9	9	23
22	0	1	11	8	12
23	3	3	8	6	14
24	1	1	22	14	24
25	4	6	14	10	24
26	0	2	14	12	16
27	3	0	9	8	12
28	3	1	19	18	23
29	0	1	13	12	14
30	0	0	10	8	10
31	0	1	6	8	9
32	3	0	13	7	16
33	2	12	19	9	33
34	0	2	15	12	17
35	0	0	7	6	7
36	1	6	12	7	19
37	4	4	14	7	22
38	0	1	12	9	13
39	2	1	5	2	8
40	38	9	16	11	63
41	0	15	22	8	37
42	0	0	12	11	12
43	6	3	9	8	18
44	12	11	21	12	44
45	3	3	13	9	19
46	1	1	10	6	12
47	0	6	12	10	18
48	1	8	14	13	23
49	4	6	11	15	21
50	0	8	9	11	17
51	4	9	14	12	27
52	1	5	16	17	24
53	0	15	17	21	32
54	5	0	9	3	14
55	0	5	13	16	18
56	1	3	21	15	25
57	3	3	17	15	23
58	6	4	22	13	32

Figura . Una muestra de los resultados impresos por el programa descrito en la sección IX.