



1. The alignment of a type 'c' field in a report is left Aligned.
2. In the statement `Write:/15(10) Ofal-lifnr.` what do the number 15 and 10 stand for

15 stand for the offset on the screen and 10 stands for the field length displayed.
3. Specify the default alignment for the following field types:

'D' – Left, 'F'-Right, 'N'-Left, 'I'-Right, 'T'-Left.
4. If `s_time` has the value '123456' how would you get an output of 12:34:56 with a single 'Write:' statement.

Write:`s_time` using edit mask'`--:--:--'`.

5. In order to suppress the leading zeroes of a number field the keywords used are NO-ZERO.
6. The total no of date formats that can be used to display a date during output is MM/DD/YY, DD/MM/YY, DD/MM/YYYY, MM/DD/YYYY, MMDDYY, DDMMYY, YYMMDD.
7. The UNDER Command allows for vertical alignment of fields one below the other.
8. In order to concatenate strings only for output purposes the command NO-GAP can be used in conjunction with the 'Write' statement.
9. The no of decimal places for output can be defines within a write statement. (T/F).

TRUE. Write:`<F>` decimals 2.

10. Data can be moved from one field to another using a 'Write:' Statement and stored in the desired format. (T/F).

TRUE. Write: `Date_1` to `Date_2` format `DD/MM/YY`.

11. In the statement `Write:/15(10) lfa1-lifnr.` The values 15 and 11 can also be defined by variables (T/F). False.
12. Differentiate between the following two statements if any.

ULINE.

Write: `sy-uline`.

No-difference. Except that `uline` is used outside the 'Write' Statement.

13. In order to skip a single line the number of lines need not be given as an assignment (T/F)

TRUE.



14. The "SKIP TO LINE line number" is dependent on the LINE-COUNT statement included in the report statement of the program.
15. In order to skip columns the command used is POSITION <n>.
16. In order to have boldfaced text as output the command used is Write:<f>INTENSIFIED.
17. Background and foreground colors can be interchanged using the command Format Inverse.
18. In order to restore the system defaults for all changes made with the format statement is Format Reset.
19. Like ULINE the statement VLINE is used to insert vertical lines. (T/F).

False.

20. Suppressing the number signs (+/-) is carried out using the addition NO-SIGNS to the Write statement. (T/F). False.

21. If SY-UZEIT has the value 6:34:45 it can be displayed as 063445 using No Edit Mask.
22. If the variable "Text" has the value 'ABCDEF' the output for the statement "Write:/Text+2(3)" will be "CDE".
23. The fields specified by select-options and parameters statement cannot be grouped together in the selection screen. (T/F). False.
24. When calling an external report the parameters or select-options specified in the external report cannot be called. (T/F)

FALSE.

25. Selection Texts in the text elements of the program helps in changing the displayed names of variables in the parameters statement.
26. Type F datatype cannot be used to define parameters.

27. Rounding off of values can be carried out using the write statement. (T/F). TRUE

28. How would you define the exponents for a type 'f' field?

Exponent <e>.

29. How would you format the output as left, centered or right-justified using the write statement.

Left-justified, Centered, Right-justified.

30. If the same formatting options were used for a WRITE statement that follows the FORMAT statement, which settings would take precedence.

The settings in the Write Statement.



31. For each new event, the system resets all formatting options to their default values (T/F)

TRUE.

32. All formatting options have the default value OFF. (T/F).

TRUE.

33. How would you set the formatting options statically and dynamically within a report? Statically: `FORMAT <option1>[ON|OFF]....`

Dynamically: `FORMAT <option1> = <var1><option2>=<var2>....`

34. The page footer is defined using the statement END-OF-PAGE.

35. The processing block following END-OF-PAGE is processed only if you reserve lines for the footer in the LINE-COUNT option of the REPORT statement. (T/F)

TRUE.

36. To execute a page break under the condition that less than a certain number of lines is left on a page is achieved by RESERVE n lines.

37. The RESERVE statement only takes effect if output is written to the subsequent page. No blank pages are created and it defines a block of lines that must be output as a whole. (T/F). TRUE.

38. To set the next output line to the first line of a block of lines defined with the RESERVE statement the statement BACK is used.

39. What is the limit for the length of a page if the page length is not specified in the report statement. 60,000 lines.

40. How would you start the printing process from within the program while creating a list?

NEW-PAGE PRINT ON.

41. You can change the width of pages within list levels triggered by page breaks. (T/F).

FALSE.

42. Hotspots are special areas of an output list used to trigger events. (T/F) TRUE.

43. To designate fields as hotspots at runtime, use FORMAT HOTSPOT = <h>.

44. Horizontal lines created with ULINE and blank lines created with SKIP can be formatted as hotspots. (T/F). FALSE.

45. How would you suppress the display of a parameter on the selection screen?

Parameters <p>No-Display.



46. Can you assign a matchcode object to a parameter? If so how?

Yes. PARAMETERS <p>.....MATCHCODE OBJECT <obj>.....

47. For each SELECT-OPTIONS statement, the system creates a selection table. (T/F)

TRUE.

48. To position a set of parameters or comments on a single line on the selection screen, you must declare the elements in a block enclosed by

SELECTION-SCREEN BEGIN OF LINE.

.....

SELECTION-SCREEN END OF LINE.

49. How can Symbols or R/3 icons be output on the screen?

WRITE <symbol-name>AS SYMBOL.

WRITE <icon-name> AS ICON.

50. In the standard setting, you cannot create empty lines with the WRITE statement alone. (T/F). TRUE.

REPORTING – GENERAL

1. The system field, which indicates success or failure of a SQL operation, is SY-SUBRC.
2. What is the syntax for specifying database table name at runtime in SELECT statement.

NAME = 'SPFL1'.

SELECT * FROM (NAME).

.....

.....



ENDSELECT.

3. How do you read selected lines of database table into an internal table in packages of predefined size.

```
SELECT * FROM <SPFLI> INTO TABLE <ITAB> PACKAGE SIZE <N>.
```

Where n is variable.

4. Name the WILDCARD characters which are used for comparisons with character strings & numeric strings.
'%' and '-'.
5. In SELECT statements can you specify a variable in WHERE condition or a part of the condition, if so what is the syntax.

```
SELECT * FROM <table> WHERE <var1><condition><var or const>.
```

6. Name the ABAP/4 key words, which are used to change the contents of database table.

UPDATE or MODIFY.

7. How to specify a client for database table processing.

```
TABLES SPFLI.
```

```
SELECT * FROM SPFLI CLIENT SPECIFIED WHERE MANDT BETWEEN '001' AND '003'.
```

.....

ENDSELECT.

8. How do you write a DATA object from ABAP/4 program to ABAP/4 memory and restore the same from memory to program.

```
EXPORT <f1>[FROM <g1>]<f2>[FROM <g2>].... TO MEMORY ID <key>.
```

The ID <key>, which can be up to 32 characters long, identifies the data in memory.

9. What are DATA CLUSTERS?

You can group any complex internal data objects of an ABAP/4 program together in data clusters and store them temporarily in ABAP/4 memory or for longer periods in databases. You can store data



clusters in special databases of the ABAP/4 Dictionary. These databases are known as ABAP/4 cluster databases and have a predefined structure. Storing a data cluster is specific to ABAP/4. Although you can also access cluster databases using SQL statements, only ABAP/4 statements are able to decode the structure of the stored data cluster.

10. Statements used to delete data objects in ABAP/4 memory FREE MEMORY [ID <key>].

11. How will you create a file on application server.

Open dataset <dsn> for output.

12. ABAP/4 statement for opening a file on application server for reading Open dataset <dsn> for input.

13. How will you transfer data into a file in application server?

Data fname(60) value 'mYFILE'.

Data num type i.

Open dataset fname for output.

Do 10 times.

Num = Num +1.

Transfer num to fname.

Enddo.

.....etc.

14. Name the function modules to write data from an Internal Table to the Presentation Server.

DOWNLOAD and WS_DOWNLOAD.

15. Name the function module that can be used to give information about files on Presentation Server and about its Operating System.

WS_QUERY.

16. Name the ABAP/4 key word, which is used to clear the Headerline of an Internal Table.

CLEAR<itab>.



17. Name the function modules to read data from Presentation Server into an Internal Table.

UPLOAD and WS_UPLOAD.

18. Name the ABAP/4 keywords to initialize an Internal Table with and without headerline.

REFRESH <itab>.

19. How to determine the attributes of an internal table?

DESCRIBE TABLE <itab>[LINES <lin>] [OCCURS <occ>].

20. Name the ABAP/4 key word for searching a string in an Internal Table.

SEARCH <itab> FOR <str><options>.

The different options (<options>) for the search in an internal table are:

ABBREVIATED

Searches table<itab>for a word containing the character string specified in <str>, where other characters might separate the characters. The first letter of the word and the string <str> must be the same.

STARTING AT<lin1>

Searches table<itab> for <str>, starting at line <line1>. <lin1> can be a variable.

ENDING AT<n2>

Searches table <itab>for <str>upto line<lin2>. <lin2>can be a variable.

AND MARK

If the search string is found, all the characters in the search string (and all the characters in between when using ABBREVIATED) are converted to upper case.

21. What are the different attributes that can be assigned to a variant?

The different attributes that can be assigned to a variant are....



Description

Enter a short, meaningful description of the variant. This may be upto 30 characters long.

Background only

Specify whether you want to use the variant in background processing only, or in online environment as well.

Protected variant.

Mark the field if you want to protect your variant against being changed by other users.

Do not display variant.

Mark this field if you want the variant name to be displayed in the catalog only, but not in the F4 value list.

For the selections you cover in a variant, you can enter the following attributes:

Type

The system displays whether the field is a parameter or a select option.

Protected

Mark this field for each field on the selection screen you want to protect from being overwritten. Values that you mark this way are displayed to the users, but they cannot change them, that are they are not ready to accept input.

Invisible

If you mark this column, the system will not display the corresponding field on the selection screen the user sees when starting the report program.

Variable

Mark this column if you want to set the value for this field at runtime.

22. Is it possible to create new dynamic programs during runtime of an ABAP/4 program? If so how?



To create new dynamic programs during the runtime of an ABAP/4 program, you must use an internal table. For this purpose, you should create this internal table with one character type column and a line width of 72. You can use any method you like from Filling Internal Tables to write the code of your new program into the internal table. Especially, you can use internal fields in which contents are dependent on the flow of the program that you use to create a new one, to influence the coding of the new program dynamically. The following example shows how to proceed in principal:

```
DATA CODE (72) OCCURS 10.
```

```
APPEND 'REPORT ZDYN1.'
```

```
TO CODE.
```

```
APPEND 'WRITE/"Hello, I am dynamically created!"/.'
```

```
TO CODE.
```

Two lines of a very simple program are written into the internal table CODE.

In the next step you have to put the new module, in the above example it is a report, into the library. For this purpose you can use the following statement:

Syntax

```
INSERT REPORT <prog>FROM <itab>.
```

The program <prog> is inserted in your present development class in the R/3 Repository. If a program with this name does not already exists, it is newly created with the following attributes:

Title: none,

Type: 1 (Reporting),

Application: S (Basis).

You can specify the name of the program <prog> explicitly within single quotation marks or you can write the name of a character field, which contains the program name. The name of the program must not necessarily be the same as given in the coding, but it is recommended to do so. <itab> is the internal table containing the source code. For the above example you could write:

```
INSERT REPORT 'ZDYN1' FROM CODE.
```

Or

DATA REP (8).

REP = 'ZDYN1'

INSERT REPORT REP FROM CODE.

23. Data types can be elementary or structured (T/F).

TRUE.

24. The amount of memory associated with a data type is ZERO.

25. Data objects are the physical units a program uses at runtime. (T/F).

TRUE.

26. The data object does not occupy any space in memory. (T/F)

FALSE.

27. What are the three hierarchical levels of data types and objects?

Program-independent data, defined in the ABAP/4 Dictionary.

Internal data used globally in one program.

Data used locally in a procedure (subroutine, function module)

28. How would you find the attributes of a data type or data object?

DESCRIBE FIELD <f> [LENGTH <l>] [TYPE <t> [COMPONENTS <n>]]

[OUTPUT-LENGTH <o>] [DECIMALS <d>]

[EDIT MASK <m>].

29. The components of a field string cannot have different data types. (T/F).

FALSE.

30. Field strings are also called as Record or Structures.



31. If a field string is aligned (Left, centered, right justified etc.), the filler fields are also added to the length of the type C field. (T/F).

TRUE.

32. You cannot assign a local data object defined in a subroutine or function module to a field group. (T/F)

TRUE.

33. Field group reserves storage space for the fields, and does not contain pointers to existing fields (T/F).

False.

34. Defining a field group as 'HEADER' is optional (T/F)

FALSE.

35. How would you define a field symbol?

FIELD-SYMBOLS<FS>.

36. Which function module would you use to check the user's authorization to access files before opening a file?

AUTHORITY_CHECK_DATASET

37. Name the function module used to convert logical file names to physical file names in ABAP/4 programs.

FILE_GET_NAME.

38. Parameters, which are defined during the definition of a subroutine with the FORM statement, are called Formal Parameters.

39. Parameters which are specified during the call of a subroutine with the PERFORM statement are called Actual Parameters.

40. In subroutines internal tables that are passed by TABLES, are always called by value and result. (T/F)

FALSE. They are called by reference.



1. What is interactive reporting?

It helps you to create easy-to-read lists. You can display an overview list first that contains general information and provide the user with the possibility of choosing detailed information that you display on further lists.

2. What are the uses of interactive reporting?

The user can actively control data retrieval and display during the session. Instead of an extensive and detailed list, you create a basic list with condensed information from which the user can switch to detailed displays by positioning the cursor and entering commands. The detailed information appears in secondary lists.

3. What are the event key words in interactive reporting?

Event Keyword	Event
AT LINE-SELECTION	Moment at which the user selects a line by double clicking on it or by positioning the cursor on it and pressing F2.
AT USER-COMMAND	Moment at which the user presses a function key.
TOP-OF-PAGE DURING LINE-SELECTION	Moment during list processing of a secondary list at which a new page starts.

4. What is secondary list?

It allows you to enhance the information presented in the basic list. The user can, for example, select a line of the basic list for which he wants to see more detailed information. You display these details on a secondary list. Secondary lists may either overlay the basic list completely or you can display them in an extra window on the screen. The secondary lists can themselves be interactive again.

5. How to select valid lines for secondary list?

To prevent the user from selecting invalid lines, ABAP/4 offers several possibilities. At the end of the processing block END-OF-SELECTION, delete the contents of one or more fields you previously stored for valid lines using the HIDE statement. At the event AT LINE-SELECTION, check whether the work area is initial or whether the HIDE statement stored field contents there. After processing the secondary



list, clear the work area again. This prevents the user from trying to create further secondary lists from the secondary list displayed.

6. How to create user interfaces for lists?

The R/3 system automatically, generates a graphical user interface (GUI) for your lists that offers the basic functions for list processing, such as saving or printing the list. If you want to include additional functionality, such as pushbuttons, you must define your own interface status. To create a new status, the Development Workbench offers the Menu Painter. With the Menu Painter, you can create menus and application toolbars. And you can assign Function Keys to certain functions. At the beginning of the statement block of AT END-OF-SELECTION, activate the status of the basic list using the statement: SET PF-STATUS 'STATUS'.

7. What is interactive reporting?

A classical non-interactive report consists of one program that creates a single list. Instead of one extensive and detailed list, with interactive reporting you create basic list from which the user can call detailed information by positioning the cursor and entering commands. Interactive reporting thus reduces information retrieval to the data actually required.

8. Can we call reports and transactions from interactive reporting lists?

Yes. It also allows you to call transactions or other reports from lists. These programs then use values displayed in the list as input values. The user can, for example, call a transaction from within a list of change the database table whose data is displayed in the list.

9. What are system fields for secondary lists?

SY-LSIND Index of the list created during the current event (basic list = 0)

SY-LISTI Index of the list level from which the event was triggered.

SY-LILLI Absolute number of the line from which the event was triggered.

SY-LISEL Contents of the line from which the event was triggered.

SY-CUROW Position of the line in the window from which the event was triggered (counting starts with 1)

SY-CUCOL Position of the column in the window from which the event was triggered (counting starts with 2).



- SY-CPAGE Page number of the first displayed page of the list from which the event was triggered.
- SY-STARO Number of the first line of the first page displayed of the list from which the event was triggered (counting starts with 1). Possibly, a page header occupies this line.
- SY-STACO Number of the first column displayed in the list from which the event was triggered (counting starts with 1).
- SY-UCOMM Function code that triggered the event.
- SY-PFKEY Status of the displayed list.

10. How to maintain lists?

To return from a high list level to the next-lower level (SY-LSIND), the user chooses Back on a secondary list. The system then releases the currently displayed list and activates the list created one step earlier. The system deletes the contents of the released list. To explicitly specify the list level, into which you want to place output, set the SY-LSIND field. The system accepts only index values, which correspond to existing list levels. It then deletes all existing list levels whose index is greater or equal to the index specify. For example, if you set SY-LSIND to 0, the system deletes all secondary lists and overwrites the basic list with the current secondary list.

11. What are the page headers for secondary lists?

On secondary lists, the system does not display a standard page header and it does not trigger the event. TOP-OF-PAGE. To create page headers for secondary list, you must enhance TOP-OF-PAGE: Syntax TOP-OF-PAGE DURING LINE-SELECTION. The system triggers this event for each secondary list. If you want to create different page headers for different list levels, you must program the processing block of this event accordingly, for example by using system fields such as SY-LSIND or SY-PFKEY in control statements (IF, CASE).

12. How to use messages in lists?

ABAP/4 allows you to react to incorrect or doubtful user input by displaying messages that influence the program flow depending on how serious the error was. Handling messages is mainly a topic of dialog programming. You store and maintain messages in Table T100. Messages are sorted by language, by a two-character ID, and by a three-digit number. You can assign different message types to each message you output. The influence of a message on the program flow depends on the



message type. In our program, use the MESSAGE statement to output messages statically or dynamically and to determine the message type.

Syntax:REPORT <rep> MESSAGE-ID <id>.

13. What are the types of messages?

A message can have five different types. These message types have the following effects during list processing:

.A (=Abend):

.E (=Error) or W (=Warning):

.I (=Information):

.S (=Success):

14. What are the user interfaces of interactive lists?

If you want the user to communicate with the system during list display, the list must be interactive. You can define specific interactive possibilities in the status of the list's user interface (GUI). To define the statuses of interfaces in the R/3 system, use the Menu Painter tool. In the Menu Painter, assign function codes to certain interactive functions. After an user action occurs on the completed interface, the ABAP/4 processor checks the function code and, if valid, triggers the corresponding event.

15. What are the drill-down features provided by ABAP/4 in interactive lists?

ABAP/4 provides some interactive events on lists such as AT LINE-SELECTION (double click) or AT USER-COMMAND (pressing a button). You can use these events to move through layers of information about individual items in a list.

16. What is meant by stacked list?

A stacked list is nothing but secondary list and is displayed on a full-size screen unless you have specified its coordinates using the window command.

17. Is the basic list deleted when the new list is created?

No. It is not deleted and you can return back to it using one of the standard navigation functions like clicking on the back button or the cancel button.



18. What is meant by hotspots?

A Hotspot is a list area where the mouse pointer appears as an upright hand symbol. When a user points to that area (and the hand cursor is active), a single click does the same thing as a double-click. Hotspots are supported from R/3 release 3.0c.

19. What is the length of function code at user-command?

Each menu function, push button, or function key has an associated function code of length FOUR (for example, FREE), which is available in the system field SYUCOMM after the user action.

20. Can we create a gui status in a program from the object browser?

Yes. You can create a GUI STATUS in a program using SET PF-STATUS.

21. In which system field does the name of current gui status is there?

The name of the current GUI STATUS is available in the system field SY-PFKEY.

22. Can we display a list in a pop-up screen other than full-size stacked list?

Yes, we can display a list in a pop-up screen using the command WINDOW with the additions starting at X1 Y1 and ending at X2 Y2 to set the upper-left and the lower-right corners where x1 y1 and x2 y2 are the coordinates.

23. What is meant by hide area?

The hide command temporarily stores the contents of the field at the current line in a system-controlled memory called the HIDE AREA. At an interactive event, the contents of the field are restored from the HIDE AREA.

24. When the get cursor command used in interactive lists?

If the hidden information is not sufficient to uniquely identify the selected line, the command GET CURSOR is used. The GET CURSOR command returns the name of the field at the cursor position in a field specified after the addition field, and the value of the selected field in a field specified after value.

25. How can you display frames (horizontal and vertical lines) in lists?

You can display tabular lists with horizontal and vertical lines (FRAMES) using the ULINE command and the system field SY-VLINE. The corners arising at the intersection of horizontal and vertical lines are automatically drawn by the system.



26. What are the events used for page headers and footers?

The events TOP-OF-PAGE and END-OF-PAGE are used for pager headers and footers.

27. How can you access the function code from menu painter?

From within the program, you can use the SY-UCOMM system field to access the function code. You can define individual interfaces for your report and assign them in the report to any list level. If you do not specify self-defined interfaces in the report but use at least one of the three interactive event keywords. AT LINE-SELECTION, AT PF<nn>, OR AT USER-COMMAND in the program, the system automatically uses appropriate predefined standard interfaces. These standard interfaces provide the same functions as the standard list described under the standard list.

28. How the at-user command serves mainly in lists?

The AT USER-COMMAND event serves mainly to handle own function codes. In this case, you should create an individual interface with the Menu Painter and define such function codes.

29. How to pass data from list to report?

ABAP/4 provides three ways of passing data:

---Passing data automatically using system fields

---Using statements in the program to fetch data

---Passing list attributes

30. How can you manipulate the presentation and attributes of interactive lists?

---Scrolling through Interactive Lists.

---Setting the Cursor from within the Program.

---Modifying List Lines.

31. How to call other programs?

	Report	Transaction
Call and return	SUBMIT AND RETURN	CALL TRANSACTION



Call without return SUBMIT

LEAVE TO TRANSACTION

You can use these statements in any ABAP/4 program.

32. What will exactly the hide statement do?

For displaying the details on secondary lists requires that you have previously stored the contents of the selected line from within the program. To do this, ABAP/4 provides the HIDE statement. This statement stores the current field contents for the current list line. When calling a secondary list from a list line for which the HIDE fields are stored, the system fills the stored values back into the variables in the program. In the program code, insert the HIDE statement directly after the WRITE statement for the current line. Interactive lists provide the user with the so-called 'INTERACTIVE REPORTING' facility. For background processing the only possible method of picking the relevant data is through 'NON INTERACTIVE REPORT'. After starting a background job, there is no way of influencing the program. But whereas for dialog sessions there are no such restrictions.

33. How many lists can a program can produce?

Each program can produce up to 21 lists: one basic list and 20 secondary lists. If the user creates a list on the next level (that is, SY-LSIND increases), the system stores the previous list and displays the new one. Only one list is active, and that is always the most recently created list.

FALSE.