```
/*PUT Function:

The PUT function is used to convert a SAS variable's value into a character representation.
It is commonly used to format numeric values into specific character formats.
Syntax: PUT(variable, format.)


INPUT Function:

The INPUT function is used to convert character data into numeric values.
It is useful when reading data from external sources, such as CSV files, where the data is stored as character strir
Syntax: INPUT(variable, informat.)INPUT Function:

The INPUT function is used to convert character data into numeric values.
It is useful when reading data from external sources, such as CSV files, where the data is stored as character strir
Syntax: INPUT(variable, informat.)

*/

data sample;
input id name$ grage$;
datalines;
2144 chetan A+
2109 Pradip A+
2119 Kanchan A+
2131 Nikita  A
;
proc print data=sample;

data new_data;
set sashelp.baseball;
proc print data=new_data(obs=10);

data trial;
set new_data;
new_variable=put(nhits,best.);  /*Convert Numeric to character*/
new_variable2=input(nhits,best.);?/* Convert Character to Numeric*/

proc contents data=trial;



/*DATE Function:

Syntax: DATE()
Explanation: Returns the current date in SAS date format.

TODAY Function:
Syntax: TODAY()
Explanation: Returns the current date in SAS date format, similar to the DATE function.
INTNX Function:

Syntax: INTNX(interval, start_date, increment)
Explanation: Computes the date resulting from adding or subtracting a specified interval to/from a given date.
INTCK Function:

Syntax: INTCK(interval, start_date, end_date)
Explanation: Calculates the number of interval units between two dates.
MDY Function:

Syntax: MDY(month, day, year)
Explanation: Creates a SAS date value from individual month, day, and year components.
YEAR Function:

Syntax: YEAR(date)
Explanation: Extracts the year component from a SAS date value.
MONTH Function:

Syntax: MONTH(date)
Explanation: Extracts the month component from a SAS date value.
DAY Function:

Syntax: DAY(date)
Explanation: Extracts the day component from a SAS date value.
These functions provide functionality to work with dates and perform operations such as obtaining the current date,
manipulating dates, calculating intervals, extracting components from dates, and more. By utilizing these functions
 you can perform various date-related tasks in SAS.

*/

data dates;
```

```
current_date=today();
previous_date=today()-1;
format current_date previous_date date date9.;
day=day(current_date);
weekday=weekday(current_date);
year=year(current_date);
quarter=qtr(current_date);
month=month(current_date);
date=mdy(month,day,year);
date2=mdy(3,05,2000);
day2=day(date2);
run;


/*INFORMAT Function:

Syntax: INFORMAT(variable, informat.)
Explanation: Converts character data to SAS internal numeric or date values based on a specified informat. It is use
 when reading data into SAS from external sources, such as CSV files.
FORMAT Function:

Syntax: FORMAT(variable, format.)
Explanation: Converts SAS numeric or date values to character data based on a specified format. It is primarily used
or writing data with specific formatting.
Informat:

An informat is a SAS instruction that informs SAS how to read or interpret the data values during input.
It is used to specify the structure or format of incoming data when reading data into SAS.
Informat examples include mmddyy. for month-day-year format, dollar12. for currency format, and comma8. for numeric
Format:

A format is a SAS instruction that determines how data is displayed or written.
It specifies the appearance or representation of data values when output or displayed in SAS.
Format examples include date9. for date values displayed as ddMONyyyy, dollar8. for currency format, and comma10.
for numeric values with commas. Custom Informat and Format:

SAS allows you to define custom informats and formats using the PROC FORMAT procedure. Custom informats and formats
useful when you need to handle specific data formats or apply custom transformations. Application:

The INFORMAT function is typically used when reading data into SAS to ensure that the incoming data is correctly int
 and stored in appropriate SAS formats.
The FORMAT function is commonly used when displaying or exporting data to ensure that the data is presented in the c



/*SAS counts the dates numbers from date  1st JAN 1960*/


data dates1;
format date ddmmyy10.;
informat date
date=03/05/2000;
date2=03/05/2000;
run;


/*Purpose: INTCK is used to determine the number of intervals (such as days, weeks, months, or years) between two gi

Syntax: The syntax of INTCK may vary depending on the programming language, but the general structure is INTCK(inter
 start_date, end_date).
The "interval" parameter specifies the type of interval you want to calculate, such as 'day', 'week', 'month', or 'y
The "start_date" and "end_date" are the dates or times between which you want to calculate the interval.
Return Value: INTCK returns an integer value representing the count of intervals between the start and end dates or

Examples:

INTCK('day', '01JAN2023'd, '31DEC2023'd) would return 365, representing the number of days between January 1, 2023,


data date_test;
format admission_date discharge_date application_date current_date date9.;
admission_date="10Jan2023"d;
discharge_date="8feb2023"d;
hospitalization_day=intck('day',admission_date,discharge_date);
months_on_book=intck('month',admission_date,discharge_date);
months_on_boo_conti=intck('month',admission_date,discharge_date,'c');
application_date="9Jan2023"d;
current_date=today();


/*Calculating visits visits in hospital_data*/
data visit;
```

```
format first_visit second_visit third_visit fourth_visit date9.;
first_visit='10oct2010'd;
second_visit=intck('day',first_visit,20);
third_visit=intck('day',second_visit,40);
fourth_visit=intck('day',third_visit,60);

proc print data=visit;
```