Assignment No.8

1. Read in following data:

John 1 Jan 1960

Mary 11 Jul 1955

Kate 12 NOV 1962

Mark 8 Jun 1959

1. Crete at temporary data file called ‘dates’ and Read in the dates in the data set using format date11.

Data dates;

Input name $ 1-4 @6 bday date11.;

Cards;

John 1 Jan 1960

Mary 11 Jul 1955

Kate 12 NOV 1962

Mark 8 Jun 1959

;

Run;

Proc print data= dates;

Run;

1. Check the output and note that for John the date is 1 Jan 1960 and the value for bday is 0. This is because dates are store internally in SAS as the number of days from Jan 1, 1960. Since Mary was born before 1960 the value of bday for her is negative (-1635)
2. SAS date has no meaning to Us, How can we format to show dates that we understand? (lets try formatting using date9. This time)

Proc print data=dates;

Format bday date9.;

Run;

Q.2: Want to know more about format? Lets try another one:

Let's look at the following data. At first glance it looks like the dates are so different that they couldn't be read. They do have two things in common:

1) they all have numeric months,   
2) they all are ordered month, day, and then year.

John 1 1 1960

Mary 07/11/1955

Joan 07-11-1955

Kate 11.12.1962

Mark 06081959

These dates can be read with the same format, **mmddyy11.** An example of the use of that format in a **data step** follows.

**DATA dates;**

**INPUT name $ 1-4 @6 bday mmddyy11.;**

**CARDS;**

**John 1 1 1960**

**Mary 07/11/1955**

**Joan 07-11-1955**

**Kate 11.12.1962**

**Mark 06081959**

**;**

**RUN;**

**PROC PRINT DATA=dates;**

**FORMAT bday date9. ;**

**RUN;**

The results of the above **proc print** show that all of the dates are read correctly.

**OBS NAME BDAY**

**1 John 01JAN1960**

**2 Mary 11JUL1955**

**3 Joan 11JUL1955**

**4 Kate 12NOV1962**

**5 Mark 08JUN1959**

Q3. Ready for another example to try?

Consider the following data in which the order is **month**, **year**, and **day**.

7 1948 11

1 1960 1

10 1970 15

12 1971 10

You may read these data with each portion of the date in a separate variable as in the **data step** that follows.

**DATA dates;**

**INPUT month 1-2 year 4-7 day 9-10;**

**bday=MDY(month,day,year);**

**CARDS;**

**7 1948 11**

**1 1960 1**

**10 1970 15**

**12 1971 10**

**;**

**RUN;**

**PROC PRINT DATA=dates;**

**FORMAT bday date9. ;**

**RUN;**

Notice the function **mdy(month,day,year)** in the **data step**. This function is used to create a date value from the individual components. The result of the**proc print** follows.

OBS MONTH YEAR DAY BDAY

1 7 1948 11 11JUL1948

2 1 1960 1 01JAN1960

3 10 1970 15 15OCT1970

4 12 1971 10 10DEC1971

How about Two digit dates?

Consider the following data, which are the same as above except that only two digits are used to signify the year, and year appears last.

7 11 18

7 11 48

1 1 60

10 15 70

12 10 71

Reading the data is the same as we just did.

**DATA dates;**

**INPUT month day year ;**

**bday=MDY(month,day,year);**

**CARDS;**

**7 11 18**

**7 11 48**

**1 1 60**

**10 15 70**

**12 10 71**

**;**

**RUN;**

**PROC PRINT DATA=dates;**

**FORMAT bday date9. ;**

**RUN;**

The results of the **proc print** are shown below.

OBS MONTH DAY YEAR BDAY

1 7 11 18 11JUL1918

2 7 11 48 11JUL1948

3 1 1 60 01JAN1960

4 10 15 70 15OCT1970

5 12 10 71 10DEC1971

Two digit years work here because SAS assumes a cutoff (**yearcutoff**) before which value two digit years are interpreted as the year **2000** and above and after which they are interpreted as **1999** and below. The default **yearcutoff** differs for different versions of SAS:

**SAS 6.12 and before (YEARCUTOFF=1900)**

**SAS 7 and 8 (YEARCUTOFF=1920)**

The **options** statement in the program that follows changes the **yearcutoff**value to **1920**. This causes in two digit years lower than **20** to be read as after the year **2000**. Running the same program then will yield different results when this option is set.

**OPTIONS YEARCUTOFF=1920;**

**DATA dates;**

**INPUT month day year ;**

**bday=MDY(month,day,year);**

**CARDS;**

**7 11 18**

**7 11 48**

**1 1 60**

**10 15 70**

**12 10 71**

**;**

**RUN;**

**PROC PRINT DATA=dates;**

**FORMAT bday date9. ;**

**RUN;**

The results of the **proc print** are shown below. The first observation is now read as occurring in 2018 instead of 1918.

OBS MONTH DAY YEAR BDAY

1 NA NA 18 11JUL2018

2 7 11 48 11JUL1948

3 1 1 60 01JAN1960

4 10 15 70 15OCT1970

5 12 10 71 10DEC1971