Sum of N Natural Numbers:

In [2]:

```
a=as.numeric(readline("enter a number: "))
b=0
while(a>0)
    {
    b=b+a
    a=a-1
}
paste("The sum is: ",b)
```

```
enter a number: 15
```

'The sum is: 120'

Find the Factorial of a given number:

In [4]:

```
fact= function(numb){
  if(numb==1){
    return(1)
  }else{
    return (numb*fact(numb-1))
  }
}
n = as.integer(readline("enter a number: "))
fact(n)
```

```
enter a number: 12
```

479001600

Get the Term 1 result details and Sort it in order:

In [ ]:

```
name=c()
result=c()
name=readline()
```

Create the elective course form and get the Unique ID, Name, Subject and if the subject is empty assign the value as "DataScience" :

In [5]:

```
elective = function(name,uniqueid, dept, subject="DS")
    {
    print(name)
    print(uniqueid)
    print(dept)
    print(subject)
}
n= readline("Enter the name: ")
m= readline("Enter the UniqueID: ")
o= readline("Enter the department: ")
p= readline("Enter your Subject: ")
if(p=="")
    {
    elective(n,m,o)
} else{
    elective(n,m,o,p)
}
```

```
Enter the name: nanthiesh
Enter the UniqueID: E012015
Enter the department: AIML
Enter your Subject: Java
[1] "nanthiesh"
[1] "E012015"
[1] "AIML"
[1] "Java"
```

In [7]:

```
elective = function(name,uniqueid, dept, subject="DataScience")
    {
    print(name)
    print(uniqueid)
    print(dept)
    print(subject)
}
n= readline("Enter the name: ")
m= readline("Enter the UniqueID: ")
o= readline("Enter the department: ")
p= readline("Enter your Subject: ")
if(p=="")
    {
    elective(n,m,o)
} else{
    elective(n,m,o,p)
}
```

```
Enter the name: Nanthiesh
Enter the UniqueID: E0121015
Enter the department: AIML
Enter your Subject: 
[1] "Nanthiesh"
[1] "E0121015"
[1] "AIML"
[1] "DataScience"
```

Write a R code to check whether the given card number is is valid or not by considering the following properties of the credit card number:

```
                    (a) It will start with 4 or 7 or 9
                    (b) It will have 16 digits
                    (c) It will not end with 00
```

In [ ]:

```

```

Reading a .CSV File:

In [9]:

```
df=read.csv("employee.csv")
df
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_
198	Donald	OConnell	DOCONNEL	650.507.9833	21-Jun-07	SH_CLERK	2600	124	5
199	Douglas	Grant	DGRANT	650.507.9844	13-Jan-08	SH_CLERK	2600	124	5
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-03	AD_ASST	4400	101	1
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-Feb-04	MK_MAN	13000	100	2
202	Pat	Fay	PFAY	603.123.6666	17-Aug-05	MK_REP	6000	201	2
203	Susan	Mavris	SMAVRIS	515.123.7777	07-Jun-02	HR_REP	6500	101	4
204	Hermann	Baer	HBAER	515.123.8888	07-Jun-02	PR_REP	10000	101	7
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-Jun-02	AC_MGR	12008	101	11
206	William	Gietz	WGIETZ	515.123.8181	07-Jun-02	AC_ACCOUNT	8300	205	11
100	Steven	King	SKING	515.123.4567	17-Jun-03	AD_PRES	24000	-	9
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-05	AD_VP	17000	100	9
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-01	AD_VP	17000	100	9
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-Jan-06	IT_PROG	9000	102	6
104	Bruce	Ernst	BERNST	590.423.4568	21-May-07	IT_PROG	6000	103	6
105	David	Austin	DAUSTIN	590.423.4569	25-Jun-05	IT_PROG	4800	103	6
106	Valli	Pataballa	VPATABAL	590.423.4560	05-Feb-06	IT_PROG	4800	103	6
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-Feb-07	IT_PROG	4200	103	6
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-Aug-02	FI_MGR	12008	101	10
109	Daniel	Faviet	DFAVIET	515.124.4169	16-Aug-02	FI_ACCOUNT	9000	108	10
110	John	Chen	JCHEN	515.124.4269	28-Sep-05	FI_ACCOUNT	8200	108	10
111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-Sep-05	FI_ACCOUNT	7700	108	10
112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-Mar-06	FI_ACCOUNT	7800	108	10
113	Luis	Popp	LPOPP	515.124.4567	07-Dec-07	FI_ACCOUNT	6900	108	10
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-Dec-02	PU_MAN	11000	100	3
115	Alexander	Khoo	AKHOO	515.127.4562	18-May-03	PU_CLERK	3100	114	3
116	Shelli	Baida	SBAIDA	515.127.4563	24-Dec-05	PU_CLERK	2900	114	3
117	Sigal	Tobias	STOBIAS	515.127.4564	24-Jul-05	PU_CLERK	2800	114	3
118	Guy	Himuro	GHIMURO	515.127.4565	15-Nov-06	PU_CLERK	2600	114	3
119	Karen	Colmenares	KCOLMENA	515.127.4566	10-Aug-07	PU_CLERK	2500	114	3
120	Matthew	Weiss	MWEISS	650.123.1234	18-Jul-04	ST_MAN	8000	100	5
121	Adam	Fripp	AFRIPP	650.123.2234	10-Apr-05	ST_MAN	8200	100	5
122	Payam	Kaufling	PKAUFLIN	650.123.3234	01-May-03	ST_MAN	7900	100	5
123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-Oct-05	ST_MAN	6500	100	5
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-Nov-07	ST_MAN	5800	100	5
125	Julia	Nayer	JNAYER	650.124.1214	16-Jul-05	ST_CLERK	3200	120	5
126	Irene	Mikkilineni	IMIKKILI	650.124.1224	28-Sep-06	ST_CLERK	2700	120	5
127	James	Landry	JLANDRY	650.124.1334	14-Jan-07	ST_CLERK	2400	120	5
128	Steven	Markle	SMARKLE	650.124.1434	08-Mar-08	ST_CLERK	2200	120	5
129	Laura	Bissot	LBISSOT	650.124.5234	20-Aug-05	ST_CLERK	3300	121	5
130	Mozhe	Atkinson	MATKINSO	650.124.6234	30-Oct-05	ST_CLERK	2800	121	5
131	James	Marlow	JAMRLOW	650.124.7234	16-Feb-05	ST_CLERK	2500	121	5
132	TJ	Olson	TJOLSON	650.124.8234	10-Apr-07	ST_CLERK	2100	121	5
133	Jason	Mallin	JMALLIN	650.127.1934	14-Jun-04	ST_CLERK	3300	122	5
134	Michael	Rogers	MROGERS	650.127.1834	26-Aug-06	ST_CLERK	2900	122	5
135	Ki	Gee	KGEE	650.127.1734	12-Dec-07	ST_CLERK	2400	122	5
136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-Feb-08	ST_CLERK	2200	122	5
137	Renske	Ladwig	RLADWIG	650.121.1234	14-Jul-03	ST_CLERK	3600	123	5
138	Stephen	Stiles	SSTILES	650.121.2034	26-Oct-05	ST_CLERK	3200	123	5
139	John	Seo	JSEO	650.121.2019	12-Feb-06	ST_CLERK	2700	123	5
140	Joshua	Patel	JPATEL	650.121.1834	06-Apr-06	ST_CLERK	2500	123	5

In [10]:

```
ncol(df)
```

10

In [11]:

```
nrow(df)
```

50

In [12]:

```
head(df, 10)
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_I
198	Donald	OConnell	DOCONNEL	650.507.9833	21-Jun-07	SH_CLERK	2600	124	5
199	Douglas	Grant	DGRANT	650.507.9844	13-Jan-08	SH_CLERK	2600	124	5
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-03	AD_ASST	4400	101	1
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-Feb-04	MK_MAN	13000	100	2
202	Pat	Fay	PFAY	603.123.6666	17-Aug-05	MK_REP	6000	201	2
203	Susan	Mavris	SMAVRIS	515.123.7777	07-Jun-02	HR_REP	6500	101	4
204	Hermann	Baer	HBAER	515.123.8888	07-Jun-02	PR_REP	10000	101	7
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-Jun-02	AC_MGR	12008	101	11
206	William	Gietz	WGIETZ	515.123.8181	07-Jun-02	AC_ACCOUNT	8300	205	11
100	Steven	King	SKING	515.123.4567	17-Jun-03	AD_PRES	24000	-	9

In [13]:

```
tail(df)
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
45	135	Ki	Gee	KGEE	650.127.1734	12-Dec-07	ST_CLERK	2400	122	50
46	136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-Feb-08	ST_CLERK	2200	122	50
47	137	Renske	Ladwig	RLADWIG	650.121.1234	14-Jul-03	ST_CLERK	3600	123	50
48	138	Stephen	Stiles	SSTILES	650.121.2034	26-Oct-05	ST_CLERK	3200	123	50
49	139	John	Seo	JSEO	650.121.2019	12-Feb-06	ST_CLERK	2700	123	50
50	140	Joshua	Patel	JPATEL	650.121.1834	06-Apr-06	ST_CLERK	2500	123	50

In [14]:

```
max(df$SALARY)
```

24000

In [15]:

```
min(df$SALARY)
```

2100

In [16]:

```
a=subset(df, SALARY == max(SALARY)) #details of the perosn having maximum salary
a
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
10	100	Steven	King	SKING	515.123.4567	17-Jun-03	AD_PRES	24000	-	90

In [17]:

```
table(df$JOB_ID)
```

```
AC_ACCOUNT     AC_MGR    AD_ASST    AD_PRES      AD_VP FI_ACCOUNT     FI_MGR 
         1          1          1          1          2          5          1 
    HR_REP    IT_PROG     MK_MAN     MK_REP     PR_REP   PU_CLERK     PU_MAN 
         1          5          1          1          1          5          1 
  SH_CLERK   ST_CLERK     ST_MAN 
         2         16          5 
```

In [18]:

```
factor(df$JOB_ID)
```

```
SH_CLERK  SH_CLERK  AD_ASST  MK_MAN  MK_REP  HR_REP  PR_REP  AC_MGR  AC_ACCOUNT  AD_PRES  AD_VP
AD_VP  IT_PROG  IT_PROG  IT_PROG  IT_PROG  IT_PROG  FI_MGR  FI_ACCOUNT  FI_ACCOUNT  FI_ACCOUNT
FI_ACCOUNT  FI_ACCOUNT  PU_MAN  PU_CLERK  PU_CLERK  PU_CLERK  PU_CLERK  PU_CLERK  ST_MAN  ST_MAN
ST_MAN  ST_MAN  ST_MAN  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK
ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK  ST_CLERK
```

▶ **Levels**:

In [25]:

```
a= subset(df, JOB_ID=="SH_CLERK")
a
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
198	Donald	OConnell	DOCONNEL	650.507.9833	21-Jun-07	SH_CLERK	2600	124	50
199	Douglas	Grant	DGRANT	650.507.9844	13-Jan-08	SH_CLERK	2600	124	50

In [26]:

```
b= subset(df, SALARY>5000 & JOB_ID=="IT_PROG")
b
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	MANAGER_ID	DEPARTMENT_ID
13	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-Jan-06	IT_PROG	9000	102	60
14	104	Bruce	Ernst	BERNST	590.423.4568	21-May-07	IT_PROG	6000	103	60

In [ ]:

In [27]:

```
install.packages("lubridate")
```

```
also installing the dependencies 'cpp11', 'timechange', 'generics'



  There are binary versions available but the source versions are later:
           binary source needs_compilation
cpp11       0.2.7  0.4.3             FALSE
timechange  0.0.2  0.1.1              TRUE
generics    0.1.0  0.1.3             FALSE
lubridate  1.7.10  1.9.0              TRUE

  Binaries will be installed
package 'timechange' successfully unpacked and MD5 sums checked
package 'lubridate' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\nanth\AppData\Local\Temp\RtmpeIwF9F\downloaded_packages

installing the source packages 'cpp11', 'generics'
```

In [42]:

```
library("lubridate")
```

```
Warning message:
"package 'lubridate' was built under R version 3.6.3"
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

In [ ]:

```

```

In [ ]:

```
ymd(df$Invoice)
```

In [ ]:

```
day(df$Invoice)
```

In [ ]:

```
year(df$Invoice)
```

In [ ]:

```
#the number of days to settle the invoice:
a=as.Date(df$due)-as.Date(df$Invoice)
a
```

In [ ]:

```
ymd(2017)
```

In [40]:

```
month(a)
```

```
Error in month(a): could not find function "month"
Traceback:
```

In [ ]:

```
# find overdue:
a=Sys.Date() -as.Date(df$due)
a
```

In [ ]:

```
df[year(df$due)%%4==0]
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

```
             Sys.Date()-as.Date(df$Due)
Leap year in invoice:
             df[year{df$Invoice)%%4==0,]
The number of days to settle the invoice:
             as.date(df$Due) -as.Date(df$Invoice)
Quarter of Invoice:
             quarter(df$Invoice)
```

In [ ]: