R-PROGRAM

Roshan.J

192125065

ITA0443

BASIC OPERATIONS IN R

1.Program:

```
v1+v2
v2-v1
v1*v2
v2/v1
print(class(v1))
print(typeof(v1))
v3<-c("qwer","erty","uiop")
print(class(v3))
v4<-c(6.5,5.5,7.5)
print(class(v4))
print(typeof(v4))
b<-TRUE
print(class(b))
a<-c(3i,9+4i,34i)
print(class(a))
```

output:

```
[1]  3  6  9 12
[1] 1 2 3 4
[1]  2  8 18 32
[1] 2 2 2 2
[1] "numeric"
[1] "double"
[1] "character"
[1] "numeric"
[1] "double"
[1] "logical"
```

[1] "complex"

DATA STRUCTURES IN R

2.Program:

```
#Vectors
v1<-c(1, 2, 3)
v2<-c(4, 5, 6)
vector<-v1+v2
vector
#Lists
a<-c("qwer","rtyu","yuio","asdf")
b<-c(1,2,3,4)
c<-4
dlist<-list("names"=a,"roll_no"=b,"no_of_persons"=c)
print(dlist)
print(dlist[[2]])
print(dlist[[1]][2])
#Matrices
u<-matrix(c(1:9),nrow=3,byrow=TRUE)
u
#Arrays
s<-array(1:9,dim=c(2:3))
s
#Factors
data=c(1,2,2,3,1,2,3,3,1,2,3,3,1)
fdata=factor(data)
fdata
#Dataframes
df<-data.frame(
  name=c("a", "b", "c"),
  age=c(20, 25, 30),
  salary=c(50000, 60000, 70000)
)
df
df$name
```

output:

```
[1] 5 7 9
$names
[1] "qwer" "rtyu" "yuio" "asdf"

$roll_no
[1] 1 2 3 4

$no_of_persons
[1] 4

[1] 1 2 3 4
[1] "rtyu"
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
 [1] 1 2 2 3 1 2 3 3 1 2 3 3 1
Levels: 1 2 3
  name age salary
1    a  20  50000
2    b  25  60000
3    c  30  70000
[1] "a" "b" "c"
```

WORKING WITH LOOPING & FUNCTION IN R

3.Program:

```
#looping
sum<-0
for (i in 1:10)
{
```

```
  sum<-sum+i
}
print(sum)
#function
add_n<-function(x,y)
{
  x+y
}
result<-add_n(3,4)
print(result)
#rec_func
n<-as.numeric(readline(prompt="Enter a number:"))
factorial<-function(n)
{
  if (n==0)
  {
    return(1)
  }
  else
  {
    return(n*factorial(n-1))
  }
}
```

output:

```
[1] 55
[1] 7
Enter a number:5
[1]120
```

IMPLEMENTATION OF VECTOR RECYCLING,APPLY FAMILY & RECURSION

4.Program:

```
#vector recycling
a=1:6
```

```
b=1:2
print(a+b)
#recursion
Sum<-function(sum,x)
{
   res=0
   if(x==1)
   {
      res=sum+1
   }
   else if(x == 0&sum<5)
   {
      res=0
   }
   else
   {
      res=sum
   }
   res
}
x=c(1,1,0,1,1,1,1,1,0,1,1,1,1,1,1,0,0,1,1)
Reduce(x=x,f=Sum,accumulate=T)
```

output:

```
[1] 2 4 4 6 6 8
 [1] 1 2 0 1 2 3 4 0 1 2 3 4 5 5 5 6 7
```