

```

> #Instructor:

#BRIJ MOHAN
#Hansraj College,
#University of Delhi, Delhi.

>
> #Programming in Maple
>
> #if statement
> restart;
> a := 6; b := 4;

a := 6
b := 4
(1)

> if a > b then a end if;

6
(2)

> if a > b then
    a
  else
    b
  end if;

6
(3)

> #Checking a number
x := -10 :
if x ≥ 0 then
  print("Number is Positive");
else
  print("Number is Negative");
end if;

"Number is Negative"
(4)

> #swaping the numbers
a := 15 :
b := 10 :
if a > b then
  a := a + b :
  b := a - b :
  a := a - b :
else:
  print("No Swapping");
end if:
a; b;

10
15
(5)

> #Checking for division
marks := 319 :
total := 500 :
percentage := evalf(marks * 100 / total);

```

```

if percentage ≥ 60 then
    printf("First Division");
else
    printf("Below 60%");
end if;

```

percentage := 63.80000000

First Division

```

> #Use of nested if statement
    marks := 219 :
    total := 500 :
    percentage := evalf(marks * 100 / total);

```

```

if percentage ≥ 60 then
    print("First Division");
elif percentage ≥ 45 then
    print("Second Division");
else
    print("Third Division");
end if;

```

percentage := 43.80000000

"Third Division"

(6)

```

> #for statement

```

```

> for i from 1 to 10 by 2 do
    print(i);
end do;

```

1

3

5

7

9

(7)

```

> #finding sum
    mysum := 0 :
    for i from 1 to 10 do
        mysum := mysum + i;
    end do:
    mysum;

```

55

(8)

```

> #finding product
    myprod := 1 :
    for i from 1 to 10 do
        myprod := myprod · i;
    end do:
    myprod;

```

3628800

(9)

```

> #checking a number for prime

```

```

# Initialize
n := 37 :
flag := 1 :

# Check divisors from 2 to n-1
for j from 2 to n-1 do
    if mod(n,j) = 0 then
        flag := 0 :
    end if:
end do:

# Report result
if flag = 1 then
    print("Prime Number")
else
    print("Not a Prime Number")
end if:

```

"Prime Number"

(10)

> #finding the prime numbers between 2 and 20

```

for i from 2 to 20 do
    n := i :
    flag := 1 :
    for j from 2 to n - 1 do
        if n mod j = 0 then
            flag := 0 :
            break:
        end if:
    end do:
    if flag = 1 then
        printf("Prime Number : %d\n", n);
    end if:
end do:
Prime Number : 2
Prime Number : 3
Prime Number : 5
Prime Number : 7
Prime Number : 11
Prime Number : 13
Prime Number : 17
Prime Number : 19

```

> #finding largest number in given list

```

mylist := [12, 34, 96, 31, 78, 65, 45, 18, 67] :
len := nops(mylist) :
mymax := mylist[1] :

```

```

for i from 2 to len do
    if mylist[i] > mymax then
        mymax := mylist[i];
    end if;
end do:

```

mymax,

96

(11)

```
> #searching the element in given list
mylist := [12, 34, 96, 31, 78, 65, 45, 78, 18, 67]:
len := nops(mylist):
n := 78:

for i from 1 to len do
  if mylist[i] = n then
    printf("Element found at position : %d\n", i);
  end if;
end do;
Element found at position : 5
Element found at position : 8
```

```
> #matrix multiplication
# Load LinearAlgebra package
with(LinearAlgebra):

# Define matrices A and B
A := Matrix([ [1, 2], [3, 1]]);
B := Matrix([ [6, 2], [7, 4]]);

# Initialize zero matrix P
n := 2:
P := Matrix(n, n, 0): # 2x2 zero matrix

# Manual matrix multiplication using nested loops
for i from 1 to n do
  for j from 1 to n do
    for k from 1 to n do
      P[i,j] := P[i,j] + A[i,k]*B[k,j];
    end do;
  end do;
end do;

# Display resulting matrix
P := P;

# Compare with built-in matrix product
C := Multiply(A, B);
```

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$$

$$B := \begin{bmatrix} 6 & 2 \\ 7 & 4 \end{bmatrix}$$

$$P := \begin{bmatrix} 20 & 10 \\ 25 & 10 \end{bmatrix}$$

$$C := \begin{bmatrix} 20 & 10 \\ 25 & 10 \end{bmatrix}$$

(12)

```
> #sorting a list
  restart :
  with(LinearAlgebra) :
```

```
mylist := [12, 134, 96, 131, 78, 65, 45, 78, 18, 67] :
len := nops(mylist) :
```

```
#using bubble sort
```

```
for i from 1 to len-1 do
  mymax := mylist[1];
  for j from 2 to len - (i - 1) do
    if mymax > mylist[j] then
      mylist[j - 1] := mylist[j];
      mylist[j] := mymax;
      mymax := mylist[j];
    else
      mymax := mylist[j];
    end if;
  end do;
end do:
mylist,
```

[12, 18, 45, 65, 67, 78, 78, 96, 131, 134]

(13)

```
> #while statement
```

```
> i := 1 :
while i ≤ 10 do
  print(i);
  i := i + 2 :
end do:
```

1
3
5

7

9

(14)

```
> #finding sum
mysum := 0 :
i := 1 :
while i ≤ 10 do
    mysum := mysum + i;
    i := i + 1;
end do:
mysum;
```

55

(15)

```
> #checking for prime number
n := 37 :
flag := 1 :
j := 2 :

while j < n do
    if n mod j = 0 then
        flag := 0;
    end if;
    j := j + 1;
end do:

if flag = 1 then
    printf("Prime Number\n");
else
    printf("Not a Prime Number\n");
end if;
Prime Number
```

```
>
```