

Task No. 01 Date \_\_\_\_\_ Name of the Task Hello world program  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

Algorithm, Program & Output

CL - USER > (defun hello)  
(format t ("Hello, world!~%"))

O/P:- Hello

CL - USER > (hello)

O/P:- Hello, world !

NIL

Task No. 02 Date 11/11/2023

Name of the Task

Global Variable, Local Variable & Constants

Software used \_\_\_\_\_

Login time \_\_\_\_\_

Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_

Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_

Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

(let ((str "Hello, world!"))

(String-upcase str))

;; ⇒ "Hello, WORLD!"

→ (let ((x 1))

(+ 5))

(+ x 4))

;; ⇒ 6

→ (let & ((x 1))

(+ (+ x 1)))

y)

;; ⇒ 2

Task No Q3 Date 10/10/2023 Name of the Task Macros  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

Algorithm, Program & Output

(defmacro set-to-10 (num)  
(setq num 10) (print num))  
(setq x 25)  
(print x)  
(set-to-10 x)

C1P :- 25  
10

Task No Q6 Date 10/10/2023 Name of the Task Assigning Value to Variables  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

Global Variable

(def var x 234)

(write x)

Op: 234

→ (setq x 10)

Assign value 10 to X

→ (setq x 10)

(setq y 20)

(format t "x>n2d y>n2d ~n~f. x y")

(setq x 60)

(setq y 80)

(format t "x>n2d y>n2d ~x y")

Op: x=10 y=20

x=100 y=200

Task No \_\_\_\_\_ Date \_\_\_\_\_ Name of the Task \_\_\_\_\_  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

#### Local Variables

→ (let (x'a) (y'b) (z'c))

(format t "X=N A Y=N B Z=N C" x y z))

O/P: X=A Y=B Z=C  
(prog ((x'(a b c) (y'(1 2 3)) (z'(P Q 10)))

(format t "X=N A Y=N B Z=N C" x y z))

O/P:

X=(A B C) Y=(1 2 3) Z=(P Q 10)

Task No Q5 Date \_\_\_\_\_ Name of the Task Arithmetic  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

$\rightarrow$  (setq a 10)  
(setq b 20)  
(format t "sum: A+B = ~d ~" (+ a b))  
(format t "diff: A-B = ~d ~" (- a b))  
(format t "prod: A\*B = ~d ~" (\* a b))  
(format t "quotient: B/A = ~d ~" (/ b a))  
(format t "Increment A by 3 = ~d ~" (incf a 3))  
(format t "Decrement A by 4 = ~d ~" (decf a 4))

O/P:

$$\begin{aligned}A+B &= 30 \\A-B &= 10 \\A*B &= 200 \\B/A &= 2\end{aligned}$$

Increment A by 3 = 13  
Decrement A by 4 = 9

Task No 06 Date 21/10/2022 Name of the Task Comparison & logical operators  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

→ (setq a 10)  
(setq b 20)

(format t "n~.A>B or ~n~a" (= a b))

(format t "n~.A=E B or ~n~a" (= a b))

(format t "n~.A< B or ~n~a" (< a b))

(format t "n~.A<=B or ~n~a" (<= a b))

(format t "n~.A>=B or ~n~a" (>= a b))

(format t "n~.A<>B or ~n~a" (<> a b))

(format t "n~.Max of A and B or ~n~a" (max a b))

(format t "n~.Min of A and B or ~n~a" (min a b))

C/P:

A=B or NIL

A1=B or T

A>B or NIL

A<B or T

A>=B or NIL

A<=B or T

Max of A and B or 20

Min of A and B or 10

Task No. 07 Date \_\_\_\_\_ Name of the Task Decision Making  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

→ ; define Value to 100

(Setq val 1 100)

; check the number is equal to 100 ; if yes then

(If (= val 1 100))

(Format t "Equal to (00)"))

(Terpri)

; check the number is greater than 50 ; if yes

(If (> val 1 50))

(Format t "greater than 50"))

(Terpri)

; check the number is less than 150

(If (< val 1 150))

(Format t "less than 150"))

O/P: Equal to 100  
greater than 50  
less than 150

Task No \_\_\_\_\_ Date \_\_\_\_\_ Name of the Task \_\_\_\_\_

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

- To check whether a number is greater than 200 or not
  - ; Set Value 1 to 500  
(setq val 1 500)
  - ; Check whether the Val is greater Than 200  
(cond ((> val 1 200))
    - (format t "Greater than 200")
    - (t (format t "less than 200")))
- Check whether the number is equal to 50 or not
  - ; Set number 0 to 50  
(setq number 50)
  - ; Condition Check the given number or Equal to 50  
(when (= number 50))
    - ; Statement  
(format t "Equal to 50")
  - )

Task No 08 Date \_\_\_\_\_ Name of the Task Looping Operation  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

→ Range from 1 to 5  
(Loop for i from 1 to 5)  
; display each number  
do (point 1)  
)

Ques: 1

((Loop until value of input is 7) )  
Ex: 2 → It keeps reading till you type 7  
3 → It reads 3  
4 → It reads 4  
5 → It reads 5  
Ex: if loop is reading user input within 1 to 10  
it will stop at 10 and then it will start again from 1

((Input a number ) )

Read a number  
((if it is less than 10))

((Display "i is less than 10"))

Task No 09

Date 11/11/2023

Name of the Task

Function

Software used

Login time

Logout time

Whether task Executed Yes / No

Sign of the Lab Asst.

Remarks of Lab co-ordinator

Sign of Lab co-ordinator

### Algorithm, Program & Output

(Coldgun averagenum ( $n_1, n_2, n_3, n_4$ )), if  $n_1 = n_2 = n_3 = n_4$   
 $\{ \{ \{ \{ (n_1 + n_2 + n_3 + n_4) / 4 \} \} \} \}$   
)  
(white averagenum ( $10, 20, 30, 40$ )) at not  
minimis ( $10, 20, 30, 40$ ) in order "Min to Max" from  $10, 20, 30, 40$   
GP: 25

Task No 10 Date \_\_\_\_\_ Name of the Task Factorial  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

Algorithm, Program & Output

→ defun factorial (n)  
(if (= n 0)  
 (\* n (factorial (- n 1))))  
(loop for (i from 0 to 6)  
doC format "nD!->nDn". " i (factorial(i))")



Task No. 11 Date \_\_\_\_\_ Name of the Task max3 (a b c) 3 marks

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

```
(defun max3 (a b c)
  (cond ((>a b)(cond((>a c)
    (format t "max ~d"
           (t (format t "max ~d"
           ((>b c)(format t "max ~d"
           (t (format t "max ~d"
           c)))))))
```

Task No 12 Date \_\_\_\_\_ Name of the Task GCD OF 2 numbers

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

```
(DEFUN GCD(X Y)
  (COND ((OR (= X 0) (= Y 0)) 0)
        ((< X 0) (GCD(-X) Y)))
        ((< Y 0) (GCD X (-Y)))
        (+ (GCD-POSITIVE X Y)
            (COND ((= X Y) X)
                  ((> X Y) (GCD-POSITIVE (-X Y) Y))
                  (+ (GCD-POSITIVE X (-Y X)))))))
(GCD-POSITIVE X Y)
  (COND ((= Y 0) X)
        ((> Y X) (GCD-POSITIVE Y X))
        (+ (GCD-POSITIVE X (-Y X))
            (COND ((= X Y) X)
                  ((> X Y) (GCD-POSITIVE (-X Y) Y))
                  (+ (GCD-POSITIVE X (-Y X)))))))
(DEFUN MAIN(X Y) #main program
  (GCD X Y))
```

Task No. 12 Date \_\_\_\_\_ Name of the Task Fibonacci Series  
Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_  
Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_  
Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

```
;; Find the nth Fibonacci number for any n>=0
;; PreCondition: n>0, n is integer. Behavior
;; undefined otherwise
(defun fibonacci (n)
  (cond
    ((= n 0)) ;;; Base case
    ((<= n 2)) ;;; if n<=2
    ((+ (fibonacci (- n 1)) (fibonacci (- n 2)))) ;;; then return 1
    )) ;;; else
    (+ (fibonacci (- n 1)) (fibonacci (- n 2))) ;;; return the sum of
    ;;; the results of calling
    (fibonacci (- n 1)) ;;; fibonacc(n-1) &
    (fibonacci (- n 2)) ;;; fibonacc(n-2)
    )) ;;; This is the recursive case
```

Task No 14 Date \_\_\_\_\_ Name of the Task Concept of Arrays

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

- create an array with 10-cells named my-array, we can write  
(Setf my-array (make-array '(10)))
- To access the "content" of the tenth cell,  
we write -  
caref my-array 9)

Task No. 15 Date \_\_\_\_\_ Name of the Task String Processing

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

- (write-line "Hello World")  
(write-line "Welcome to Tutorials point")  
; escaping the double quote characters  
(write-line "Welcome to \"Tutorials Point\"")  
O/P:-

Hello world

Welcome to Tutorials point

Welcome to "Tutorials point"

- String comparison functions

; case-sensitive comparison

(write (string "this is test" "This is test"))

(terpri)

(write (string > "this is test" "This is test"))

(terpri)

(write (string < "this is test" "This is test"))

(terpri)

; case-insensitive comparison

(write (string-equal "This is test" "this is test"))

(terpri)

(write (string-greaterp "this is test" "This is test"))

(terpri)

(write (string-lessp "this is test" "This is test"))

(terpri)

Task No \_\_\_\_\_ Date \_\_\_\_\_ Name of the Task \_\_\_\_\_

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

```
; checking non-equal  
(write (string1="this is test" "this is Test"))  
(teepri)  
(write (string-not-equal "this is test" "this is test"))  
(teepri)  
(write (string1="lisp" "lispng"))  
(teepri)  
(write (string1="decent" "decency"))
```

O/P:- NIL

NIL

0

NIL

T

NIL

NIL

8

NIL

4

S

-case controlling Function

```
(write-line (string-upcase "a big hello from  
tutorials point"))
```

```
(write-line (string-capitalize "a big hello  
from tutorials point"))
```

O/P:-

A BIG HELLO FROM TUTORIALS POINT

A Big Hello From Tutorials Page No. Point

Task No \_\_\_\_\_ Date \_\_\_\_\_ Name of the Task \_\_\_\_\_

Software used \_\_\_\_\_ Login time \_\_\_\_\_ Logout time \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

Remarks of Lab co-ordinator \_\_\_\_\_ Sign of Lab co-ordinator \_\_\_\_\_

### Algorithm, Program & Output

#### - Trimming strings

(write-line (string-trim " " " " a big hello  
from tutorials point"))

(write-line (string-left-trim " " " "  
a. big hello from tutorials  
point"))

(write-line (string-right-trim " " " " a big  
hello from tutorials point"))

(write-line (string-trim "a" " " . a big hello  
from tutorials point))

O/P:-

a big hello from tutorials point

a big hello from tutorials point

a big hello from tutorials point

big hello from tutorials point

#### - Accessing a character in a string

(write (length "Hello world"))

(terpri)

(write-line (subseq "Hello world" 6))

(write (char "Hello world" 6))

O/P:- 11

world

#\w

Task No \_\_\_\_\_ Date \_\_\_\_\_ Name of the Task \_\_\_\_\_

**Software used** \_\_\_\_\_ **Login time** \_\_\_\_\_ **Logout time** \_\_\_\_\_

Whether task Executed Yes / No \_\_\_\_\_ Sign of the Lab Asst. \_\_\_\_\_

**Remarks of Lab co-ordinator** \_\_\_\_\_ **Sign of Lab co-ordinator** \_\_\_\_\_

## **Algorithm, Program & Output**

## Sorting and Merging of strings

; sorting the strings

```
write (sort (vector "Arnold" "Akbar" "Anthony"))
# 'string<))
```

(teopti)

; merging the strings

```
write (merge 'vector "Rishi" "Zara" "priyanka"  
             (vector "Anju" "Anuj" "Avni"))# 'string))
```

OP :-

**#("Akbas" "Amal" "Anthony")**

#("Anju" "Anuj" "Avni" "Rishi" "Zara")

" priyanka")

## -Reversing a string

(write-line (reverse "Are we not drawn onward, we few, drawn onward to new era"))

q.P.-

are wen·ot drawno nward, wef· ew, drawno  
nward ton ew era

## - Concatenating strings

O.P.: Are we not drawn onward, we few,  
drawn onward to new era