

Q) A file Employees.txt contains Employee IDs
 of 20 employees followed by their Names. <ID><Name>
 An array, Salry, contains the Salary of each
 Employee.
 Employee ID = Index of his Salary in Array
 The Salary is to be increased by a specific
 percentage:
 If the current salary is above 70000,
 the increase should be of 12%.
 If it is between 20000 and 70000, the
 increase should be of 15%.
 Otherwise, the salary is to be increased
 by an amount of 5000.
 The Employee ID, followed by the new salary
 is to be written to a file "Salaries.txt".

→ EmployeeID of 2 digits from
the left.

```

DECLARE ArrayIndex : INTEGER
DECLARE DataToWrite, FileData, EmployeeID: STRING
DECLARE NewSalary: REAL
  
```

```

OPENFILE "Salaries.txt" FOR WRITE
OPENFILE "Employees.txt" FOR READ
WHILE NOT EOF ("Employees.txt") DO
  READFILE "Employees.txt", FileData
  EmployeeID ← LEFT (FileData, 2)
  ArrayIndex ← STRING-TO-NUM (EmployeeID)
  
```

NewSalary ← Salary [ArrayIndex]

CASE OF NewSalary

> 70 000 : NewSalary ← NewSalary * 1.12

> 20 000 AND <= 70 000 : NewSalary ← NewSalary * 1.15

OTHERWISE

NewSalary ← NewSalary + 5000

END CASE

DataToWrite ← EmployeeID & NUM-TO-STRING(NewSalary)

WRITEFILE "Salaries.txt", DataToWrite

END WHILE

CLOSEFILE "Salaries.txt"

CLOSEFILE "Employees.txt"

Loops

- 1) Write a program that inputs 100 numbers and displays their average.
- 2) Write a program that inputs 30 names and appends it to an Array Names.
- 3) Make a program that outputs first 20 multiples of a given number.
- 4) Make a program that takes 30 strings as input and outputs the Total Digits/letters in those strings combined.
- 5) Write a program that keeps on displaying cube of the number input until "0" is given as input.

Q1-

DECLARE Average: REAL
DECLARE i : INTEGER
DECLARE Number: INTEGER

Average \leftarrow 0

FOR i \leftarrow 1 TO 100

 OUTPUT "Enter Number"

 INPUT Number \rightarrow Input is of declared data type.

 Average \leftarrow Average + Number

END FOR

Average \leftarrow Average / 100

OUTPUT "Average of 100 numbers is " & NUM-TO-STRING (Average)

Q2-

DECLARE Name: STRING * USE FOR Loop.

DECLARE NameCount, i : INTEGER

NameCount \leftarrow 1

i \leftarrow 1

WHILE NameCount $<$ 31 DO

 OUTPUT "Enter Name"



 INPUT Name

 Names[i] \leftarrow Name

 NameCount \leftarrow NameCount + 1

 i \leftarrow i + 1

END WHILE

Q3-

```
DECLARE Number: STRING  
DECLARE i, Multiple: INTEGER  
OUTPUT "Enter Number"  
INPUT Number  
FOR i ← 1 TO 20  
    Multiple ← STRING-TO-NUM(Number) * i  
    OUTPUT Multiple  
END FOR
```



Q4-

```
DECLARE i: INTEGER  
DECLARE NewString, String: STRING  
NewString ← "" // Empty  
FOR i ← 1 TO 30  
    OUTPUT "Enter string"  
    INPUT String  
    NewString ← NewString & String
```

END FOR

OUTPUT NewString



*

Q 5-

DECLARE Number : STRING

DECLARE NumToCube, Cube : INTEGER

Number ← "1"

WHILE Number <> "0" DO

 OUTPUT "Enter Number"



 INPUT Number

 IF Number <> "0"

 THEN

 NumToCube ← STRING-TO-NUM(Number)

 Cube ← NumToCube * NumToCube * NumToCube

 OUTPUT Cube

 END IF

END WHILE

SELECTION

1) Write a program that outputs the highest Number out of 3 input Numbers.

2) A program that tells if a given char is vowel or consonant.

3) A program that increases salary of 20 people from salary array.

Salary $\geq 100,000 \Rightarrow$ increase by 10%.

Salary between 30,000 and 100,000 \Rightarrow inc. by 15%.

Update the salary and reassign it at the same place.

4) Write a program that accepts numbers 1-7 and outputs a day accordingly.
1 \Rightarrow Monday.
7 \Rightarrow Sunday. etc..

5) A program that inputs a char and tells if it is an integer or alphabet.

Q1-

DECLARE Num1, Num2, Num3: STRING

OUTPUT "Enter Number"

INPUT Num1

OUTPUT "Enter Number"

INPUT Num2

OUTPUT "Enter Number"

INPUT Num1, Num2, Num3

INPUT Num3

IF (Num1 > Num2) AND (Num1 > Num3)

THEN

OUTPUT Num1 & " " & "is the greatest number"

ELSE IF Num2 > Num3

THEN

OUTPUT Num2 & " " & "is the greatest number"

ELSE

OUTPUT Num3 & " " & "is the greatest number"



END IF

END IF

Q2-

DECLARE Alphabet: CHAR

OUTPUT "Enter Alphabet"

INPUT Alphabet

Alphabet \leftarrow LCASE (Alphabet)

CASE OF Alphabet

= 'a' : OUTPUT "vowel"

* Use array.

= 'e' : OUTPUT "vowel"

= 'i' : OUTPUT "vowel"

= 'o' : OUTPUT "vowel"

= 'u' : OUTPUT "vowel"

OTHERWISE

OUTPUT "Consonant"



END CASE

Q3-

DECLARE i, LocalSal : INTEGER

FOR i \leftarrow 1 TO 20

LocalSal \leftarrow Salary [i]

CASE OF LocalSal

$\geq 100\ 000 : \text{LocalSal} \leftarrow \text{LocalSal} * 1.10$

$> 30\ 000 \text{ AND } < 100\ 000 : \text{LocalSal} \leftarrow \text{LocalSal} * 1.15$

END CASE

$\text{Salary}[i] \leftarrow \text{LocalSal}$

END FOR

Q4-

DECLARE Number : CHAR

OUTPUT "Enter Number"

INPUT Number

CASE OF Number

$= '1' : \text{OUTPUT} \text{ "Monday"}$

$= '2' : \text{OUTPUT} \text{ "Tuesday"}$

$= '3' : \text{OUTPUT} \text{ "Wednesday"}$

$= '4' : \text{OUTPUT} \text{ "Thursday"}$

$= '5' : \text{OUTPUT} \text{ "Friday"}$

= '6' OUTPUT "Saturday"
= '7' OUTPUT "Sunday"

END CASE

Q5-

DECLARE Character: CHAR

OUTPUT "Enter Character"

INPUT Character

IF ISNUM(Character) = TRUE

THEN

 OUTPUT "Integer"

ELSE

 Character ← LCASE(Character)

 IF Character ≥ 'a' AND Character ≤ 'z'

 THEN

 OUTPUT "Alphabet"

END IF

END IF

STRING MANIPULATION

i) Input Full name of student and Print only the first Name.

ii) Input Name and Marks of student and print a suitable message with remarks.

≥ 90 : Excellent.

≥ 80 : Outstanding

≥ 70 : Good

e.g.

"A scored 85 marks"
"outstanding"

Q1-

DECLARE NewString, Name: STRING

DECLARE Loop: BOOLEAN

DECLARE i : INTEGER

NewString \leftarrow "" // Empty

Loop \leftarrow TRUE

OUTPUT "Enter Full Name"

INPUT Name

WHILE Loop = TRUE DO

NextChar ← MID (Name , i, 1)

IF NextChar <> ' ' // space check 'D'

THEN

NewString ← NewString & NextChar

i ← i+1

ELSE

Loop ← FALSE

END IF

END WHILE ✓

OUTPUT NewString

Q2-

DECLARE Marks, Name: STRING

OUTPUT "Enter Name"

INPUT Name

OUTPUT "Enter Marks"

INPUT Marks

IF Marks >= "90"

THEN

 OUTPUT Name & " scored " & Marks & " Marks"

 OUTPUT "Excellent"

ELSE IF Marks >= "80"

THEN

 OUTPUT Name & " scored " & Marks & " Marks"

 OUTPUT "outstanding"

ELSE IF Marks >= "70"

THEN

 OUTPUT Name & " scored " & Marks & " Marks"

 OUTPUT "Good"

END IF

END IF

END IF

iii) An Array Student contains names of 20 students.
Make a program that concatenates an ID (1-20) to
the start of each name.

e.g

Munam
Aleesa
Mubassil
Zain
:
:

Program -
⇒

1 Munam
2 Aleesa
3 Mubassil
4 Zain
:

iv) Write a program that inputs a string and
checks if first letter is a number or an alphabet.

Q3-

DECLARE i : INTEGER

DECLARE Gap: CHAR

Gap ← " " // 1 space

FOR i ← 1 TO 20

Name ← NUM-TO-STRING(i) & Gap & Student [i]

Student [i] ← Name

END FOR



Q4-

DECLARE Newstring: STRING

DECLARE NextChar: CHAR

OUTPUT "Enter String"

INPUT Newstring

NextChar ← LEFT (Newstring, 1)

IF ISNUM (NextChar) = TRUE

THEN

 OUTPUT "Number"

ELSE IF LCASE (NextChar) >= 'a' AND LCASE (NextChar) <= 'z'

 THEN

 OUTPUT "Alphabet"



END IF

END IF

1D ARRAY

1. Create an Array Customer, input 30 Names and add them to the array.
2. An Array Acc contains Account Numbers of 100 people. Output them all.
3. An Array Num contains 1000 numbers. Print only the odd Numbers from the array.

MEGA:

Create an Array Tourists.
Keep inputting and adding names to array.
When * is input, end the loop.

Q1-

DECLARE i : INTEGER

DECLARE Customer: ARRAY [1:30] OF STRING

DECLARE Name: STRING

FOR i ← 1 TO 30

 OUTPUT "Enter Name"

 INPUT Name

 Customer[i] ← Name

END FOR

Q2-

DECLARE i : INTEGER

FOR i ← 1 TO 100

 OUTPUT Acc[i]

END FOR

Q3-

DECLARE i : INTEGER

DECLARE Number : INTEGER

FOR i ← 1 TO 1000

 Number ← Num[i]

 IF MOD(Number, 2) = 1

 THEN

 OUTPUT Number

 END IF

END FOR

MEGA ☺

DECLARE Loop: BOOLEAN

DECLARE i: INTEGER

DECLARE Tourists: ARRAY OF STRING, Name: STRING

i ← 1

Loop ← TRUE

WHILE Loop = TRUE DO

 OUTPUT "Enter your Name"

 INPUT Name

 IF Name = "*"

 THEN

 Loop ← FALSE

 ELSE

 Tourists[i] ← Name

 i ← i + 1

 END IF

END WHILE

WHILE Name <> "*" DO

 INPUT Name

 Tourists[i] ← Name

 i ← i + 1

END WHILE

