PSEUDOCODE PROBLEM SET 1:

1. User inputs a number, program reports if the number is above 5

2. User inputs 3 numbers, program outputs the numbers in the opposite order

3. User inputs 2 numbers, if the first one is larger, it outputs their sum, if the second one is larger, it reports their difference

4a. User inputs a number, the program counts up from zero to that number.

4b. Now, for every other number, the program outputs "boo" , e.g. 0, “boo”, 2, “boo”.

5. Create a needy program that will keep on asking user for hugs until user inputs any phrase including the word "hug“.

6. Create a program that can convert percentage change to actual change, e.g. user enters old size, percentage change and the units and outputs the actual increase in units, e.g. 20, 10%, kg will output 2 kg.

7. User inputs an integer, the program outputs a half of the integer, rounded down, and if that half is odd or even, e.g. if user inputs 7, the half of it rounded down is 3 which is odd.

8. User inputs the volume of sphere, the program outputs the radius.

9. A car can carry up to 5 people and generates 200 g of CO2 per mile, a bus can carry 40 people and generates 1000 g of Co2. Write an algorithm to find out which car is more environmentally friendly, and output the name of that type of transport.

10. Write the algorithm to count up from 1 to 30, outputting the numbers as it goes along and outputting exclamation marks next to all numbers divisible by both 3 and 4.

11. User inputs two numbers, a and b, the program will output all the powers of a, e.g. a*1, a*a, a*a*a, up to b. b can't be less than 2.

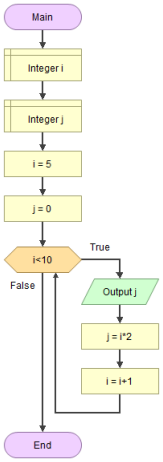
12. User inputs two positive numbers, the program counts up from the smaller of these numbers to the larger, outputting the numbers as it counts up.

13. User inputs 3 names of people, the program returns all of their lenghts (e.g. "Bob" is 3 characters in length).

14. User inputs 3 names of people, the program returns the longest name and its length,
(e.g. if "Bob", "Jeremy" and "Caspasius" have been input, "Caspasius", 9 will be output).

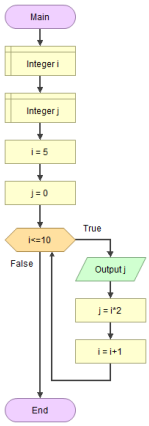
15. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```



16. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i=<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```

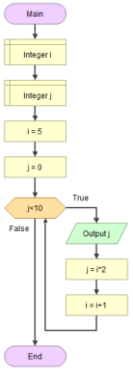


17. What will be the output of the algorithm below?

```
i=5
j=0
WHILE j<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```

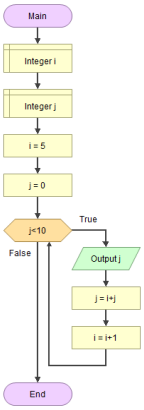
18. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i<10 DO
        PRINT j
        j=i+j
        i=i+1
END WHILE
```



19. User inputs a word, the program outputs all the letters of that word

20. User inputs a word, the program outputs the middle letter of that word, e.g. if "Jeremia" is input, it will output "e",
if "Jeremy" is input, and the word length is an even number without a middle value, the program will divide in 2 and round DOWN and output "r".



21. User inputs a word, the program outputs all the letters of that word but every other letter
(e.g. "monkey" would result in "m", "n", "e")

22. User inputs a word, the program outputs the first and the last letters of that word

23. User inputs a word, the program outputs the second and the second last letters of that word, minimal length is 5 characters.

24. User inputs the price of an item and the banknote given. The program will output the change due to the customer.
The price of the item must be less than the banknote given, otherwise the program outputs "Insufficient amound paid"

25. The program counts up 0 to 100 and also outputs the counter and one of the words “Tic”, “Tac” alternatively, e.g. 0 “Tic”, 1 “Tac”, 2 “Tic”, 3 “Tac”, etc.

26. User inputs the price of an item and the banknote given (must be 1, 2, 5, 10, 20, 50 pound notes only).
The program will output the change needed.
The price of the item must be less than the banknote given, otherwise the program outputs "Insufficient amound paid"

27. Given a list of words, the program will output 3 random things in a sequence. e.g. "cactus","fish","goat"

```
words=Array["donkey","cactus","fish","goat"]
FOR i=0 TO 3
        r=RANDOM_INTEGER(0,SIZE(words))
        OUTPUT words[r]
NEXT i
```

28. Given a list of words, the program will ask user to input a word and will search its list for a match.
If match is found, the program will output the index at which the inputted word is found, otherwise
it would output "not found", e.g. in a list "donkey","cactus","fish","goat", if the user inputs "cactus"
the program should output "found at index 1", if "zebra" is input, the output is "zebra not found"

29. Given a list of words, the program will output 2 of them randomly, putting the word "and" between them and
" are the best friends!" after when priting, e.g. "cactus and goat are best friends"

30. A user is looking for a algorithm that would efficiently output the following phrase "and 1 and 2 and 3 and 4 and 5 and 6 and 7 and 8“

31. The user is looking for a algorithm that would efficiently output the following phrase "1 and 2 and 3 and 4 and 5 and 6 and 7 and 8"

32. The user is looking for a algorithm that would efficiently output the following phrase "1 and 8 or 2 and 7 or 3 and 6 or 4 and 5 or 5 and 4 or 6 and 3"

33. Write an algorithm that will create a string array from the names of all files in a folder, reporting back the names of the files as a numbered list.

34. A known computer virus leaves this signature in infected files: "DeathJeffLOLZ". Write an algorithm that will open all files in a folder
and scan them for that signature, reporting back the names of the files infected.

35. Given a nested list (a 2D array) of names and numbers, display them as in columns separated by tabs.

35B. Given an array a[6,9,2,4] output the second item

37. Given an array a[6,9,2,4] output all items, then same but backwards

38. Given an array a[6,9,2,4] output all items

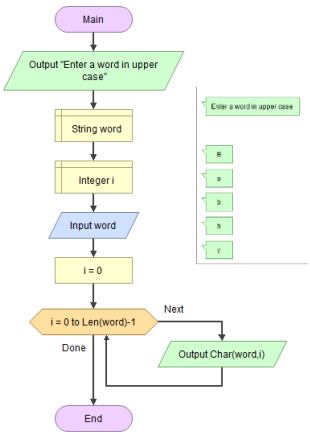
39. Given an array a[6,9,2,4] output every other item

40. Given a[6,9,2,4], write the code that will swap the first and the third item [2,9,6,4],

42. Create a flowchart and pseudocode for the following program: User enters a word and the program spells it one character at a time.

43. User inputs a word in upper case, the algorithm converts it to lower case. This is based on the fact that in the ASCII table, upper case letters are 32 letters below the lower case ones.
Consider this printout from Python (in Pseudocode we can use ToChar(x) to convert an integer to an ASCII character and ToCode("a") to convert an ASCII character to an integer representing this character's ASCII position:

```
>>> ord("A")
65
>>> ord("a")
97
>>> ord("a")-ord("A"),ord("b")-ord("B")
(32, 32)
>>> chr(65),chr(97)
('A', 'a')
>>>
```



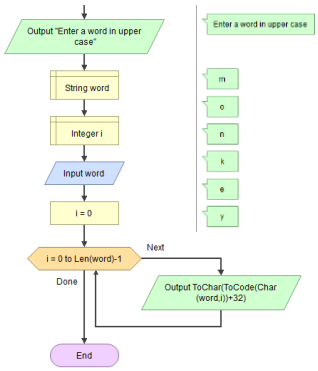
44. The program is to ask user for two words and then output the first two characters of the fir word and the last two characters of the second word.

45. Given a 1D integer array, the program will iterate (step through it) through the array and fo any element larger than 10, it will subtract 10 from that element
and display the element (both adjusted and unadjusted).
e.g. if the array holds [3,6,1,23], the output will be 3, 6, 1, 13 because 23>10, so 23-10=13

46. Given a 1D integer array, the program will iterate (step through it) through the array and fo any element smaller than 10, it will add 10 to that element
and display the element (both adjusted and unadjusted).
e.g. if the array holds [3,6,1,23], the output will be 3, 6, 1, 13 because 23>10, so 23-10=13

47. Given a 1D integer array, the program will iterate (step through it) through the array and output every element and its preceding and the
following element, if the array holds [3,6,1,23], the output will be "3 is first, 6 is after 3 and below 1, 1 is after 6 and before 23, 23 is last"

48. Given a 1D integer array, the program will iterate (step through it) through the array and find the difference between an element and
the one immediately preceding it, then write those differences into another array, e.g. if the array holds [3,6,1,23], the resulting array will be [3,-5,22] from 6-3, 1-6, 23-1
The program will then output the resulting array of differences.



49. Given a 1D integer array, the program will the length of the array (how many elements it), the sum of all integers, and the first and the last elements

50. Given a 1D integer array, the program will iterate (step through it) through the array and output the elements and whether they are odd or even.

51. Given a 1D integer array, user enters a number. The program returns "found" if that number is amongst the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "7" it will be "not found".

52. Given a 1D integer array, user enters a number. The program returns "found" if that number is amongst the FIRST half of the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "1" it will be "not found" because [3,6] is the first
part of the array and it doesn't contain 1. We can find the first half of the array by searching up to LENGTH(n) divided into 2 and rounded down (integer DIV).

```
n=Array[3,6,1,23]
INPUT x
i=0
found=False
WHILE found=False AND i<(LENGTH(n) DIV 2)
        IF n[i]=i THEN
                found=True
        ELSE
                i=i+1
        END IF
END WHILE
IF found=True THEN
        OUTPUT "found"
ELSE
        OUTPUT "not found“
```

53. Given a 1D integer array, user inputs a number. The program returns that number's index if that number is amongst the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "7" it will be "not found".

54. User has a file called “pups.txt” that has the following format:
Bobby
Johnny
Amir
Cecilius
Write a program that will read this file into a 1D array of string elements. The array will hold up to 20 elements.

55. User has a file called “pups.csv” that has the following format:
Bobby, 33
Johnny, 22
Amir, 56
Cecilius, 78
Write a program that will read this file into a 2D array of string elements. The array will hold up to 20 elements. Output the name with the highest number of points. E.g. Cecilius

56. User inputs a word and the program will display every other letter in alternating case: upper/lower, e.g. “Hello people” will become “HeLlo PeOpLe”

57. A teacher needs to count all pupils who did their homework and all those who didn't and output the totals for both in the end.

58. A teacher needs to count all pupils who did their homework and all those who didn't and output the totals for both in the end, while entering those who didn't into a special detention list.

59. Write a program that will count an approximate number of characters in a book, given the input from a user of average characters per word, avg words per line, and avg lines per page, as well as the number of pages, e.g. if there are 200 pages, with average of 30 lines per page and average of 10 words per line and 6 characters per word, we should get 360,000 characters in that book.

60. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program iterates through both arrays, displaying the elements side by side, e.g. given
["Bob","Cecil","Nora","Amir"]
as names, and [45,34,89,100] as integers, the program will output "Bob: 45 points", "Cecil: 34 points", etc.

61. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program iterates through both arrays, displaying the elements side by side, e.g. given
["Bob","Cecil","Nora","Amir"]
as names, and [45,34,89,100] as integers, but only for those who have over 40 points, so the program will output
"Bob - 45 points", "Nora - 89 points", etc, leaving out Cecil who scored fewer than 40 points.

62. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The user inputs a name, the program finds the matching name in the first array, remembers its index,
then it outputs the element from the integer array that is at the same index as the match.
If no match is found, the error code of "-999" is output.

63. A user has a list of prices [12,59,23,101] and would like to see each prices with and without VAT, with and without 10% early bird discount, and with and without the 15% surcharge for overseas customers.

64. A program needs to write “Bob was here” into a text file called “memories.txt”

A program needs to ask a user to write down 3 of their favourite Pikachu characters and write them into a text file called “p.txt”

65. A program needs to ask a user to write down any number of their favourite Pikachu characters and write them into a text file called “p.txt”

66. Given an array of strings, the program needs to write all of its elements to a file called "data.txt"

67. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program needs to write each element of the names array, followed by a comma, followed by an element
of the integer array with the same index, to each new line in a file called "data.csv"

68. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program needs to write each element of the names array, followed by a comma, followed by an element
of the integer array with the same index, to each new line in a file called "data.csv" BUT ONLY IF
the integer element is over 40, e.g. Cecil will not be written as he only has 34 points.

68. Given a table of integers, the program will ask user for a column number and return the total of that column, e.g.

4	7	12	10
2	1	8	9
14	9	3	7

if the user inputs "0", they will see 20 (which is 4+2+14)

69. Given a table of integers, the program will ask user for a row number and return the total of that row, e.g.

4	7	12	10
2	1	8	9
14	9	3	7

if the user inputs "0", they will see 33 (which is 4+7+12+10)

70. The user wants to write two procedures which are selectable via a menu from the main screen. e.g. when the program runs, the user
might see "Press 1 to hear praise, 2 to hear a put-down" where praise and put-down are defined in their respective procedures

SOLUTIONS:

1. User inputs a number, program reports if the number is above 5

```
PRINT "Enter a number"
INPUT x
IF x<5 THEN
        PRINT "Number is smaller "
ELSE IF x>=5 THEN
        PRINT "Number is bigger or equal to 5"
END IF
```

or....

```
INPUT x
IF x>5 THEN
        PRINT STRING(x) + " is larger than 5"
ELSE
        PRINT STRING(x) + " is not larger than 5"
END IF
```

2. User inputs 3 numbers, program outputs the numbers in the opposite order

```
INPUT x, y, z
PRINT z,y,x
```

3. User inputs 2 numbers, if the first one is larger, it outputs their sum
if the second one is larger, it reports their difference

```
INPUT a
INPUT b
IF a>b THEN
        PRINT a+b
ELSE IF b> a THEN
        PRINT b-a
ELSE
        PRINT "Both numbers are the same"
END IF
```

4a. User inputs a number, the program counts up from zero to that number.

```
INPUT x
FOR i=0 TO x
        PRINT i
NEXT FOR
```

4b. Now, for every other number, the program outputs "boo" , e.g. 0, “boo”, 2, “boo”

```
INPUT x
c=0
FOR i=0 TO x
        PRINT i
        IF c>0 THEN
                PRINT "boo"
                c=0
        ELSE
                PRINT i
                c=c+1
        END IF
NEXT FOR
```

hints:

```
#types of loops
FOR i=1 TO 5
        PRINT "ha"
END FOR

i=0
REPEAT UNTIL i>5
        PRINT "ha"
        i=i+1
END REPEAT

i=0
REPEAT
        print "ha"
        i=i+1
WHILE i>5
        PRINT "ha"
        i=i+1
END WHILE
```

5. Create a needy program that will keep on asking user for hugs until user inputs any phrase including the word "hug“

```
needy=True
WHILE needy=True
        OUTPUT “I want hugs, can I have a hug? “)
        INPUT u
        words=Array(0 to 100) FROM u SPLIT (delimiter=SPACE)
        found=False
        j=0
        WHILE found=False
                IF words[j]=“hug” THEN
                        found=True
                ELSE
                        j=j+1
                END IF
        END WHILE
END WHILE
OUTPUT “I am so happy now, now feed me!”
```

6. Create a program that can convert percentage change to actual change, e.g. user enters old size, percentage change and the units and outputs the actual increase in units, e.g. 20, 10%, kg will output 2 kg

```
INPUT size
INPUT percChange
INPUT units
OUTPUT STRING(size*percChange) + “ “ + units
```

7. User inputs an integer, the program outputs a half of the integer, rounded down, and if that half is odd or even, e.g. if user inputs 7, the half of it rounded down is 3 which is odd.

```
INPUT x
y=x DIV 2 //integer division which always rounds down
IF y MOD 2=0 THEN
        OUTPUT “even”
ELSE
        OUTPUT “odd”
END IF
```

8. User inputs the volume of sphere, the program outputs the radius. $V = \frac{4}{3}\pi r^3$

```
INPUT volume
R=(volume/( (4/3)*3.14) ) ^ (1/2)
OUTPUT R
```

9. A car can carry up to 5 people and generates 200 g of CO2 per mile, a bus can carry 40 people and generates 1000 g of Co2. Write an algorithm to find out which car is more environmentally friendly, and output the name of that type of transport.

```
Car_per_pass=200/5
Bus_per_pass=1000/40
IF Car_per_pass>Bus_per_pass THEN
        OUTPUT “Car”
ELSE IF Car_per_pass<Bus_per_pass THEN
        OUTPUT “Bus”
ELSE
        OUTPUT “There is not differences”
END IF
```

10. Write the algorithm to count up from 1 to 30, outputting the numbers as it goes along and outputting exclamation marks next to all numbers divisible by both 3 and 4

```
FOR k=1 TO 30
        IF k MOD 3=0 AND k MOD 4=0 THEN
                OUTPUT STRING(k) + “!”
        ELSE
                OUTPUT k //no need to convert to string if printing by itself
        END IF
NEXT k
```

11. User inputs two numbers, a and b, the program will output all the powers of a, e.g. a*1, a*a, a*a*a, up to b. b can't be less than 2.

```
INPUT a
b_is_ok=False  //begin range validation
WHILE b_is_ok=False DO
        INPUT b
        IF b>=2 THEN
                b_is_ok=True
        END IF
END WHILE //good b > 2, proceeding to output the degrees of a
FOR i=1 TO b
        a=a*i
        OUTPUT a
NEXT i
```

12. User inputs two positive numbers, the program counts up from the smaller of these numbers to the larger, outputting the numbers as it counts up.

```
larger=0 // double slash is a pseudocode comment, reset variables
smaller=0
INPUT x
INPUT y
IF x>y THEN
        larger=x
        smaller=y
ELSE IF y>x THEN
        larger=y
        smaller=x
ELSE
        PRINT "Numbers are the same, counting is not possible"
END IF
IF x<>y THEN
        FOR i=smaller TO larger
                PRINT i
        NEXT FOR
END IF
```

13. User inputs 3 names of people, the program returns all of their lenghts (e.g. "Bob" is 3 characters in length)

```
INPUT a
INPUT b
INPUT c
PRINT LENGTH(a)
PRINT LENGTH(b)
PRINT LENGTH(c)
```

14. User inputs 3 names of people, the program returns the longest name and its length,
(e.g. if "Bob", "Jeremy" and "Caspasius" have been input, "Caspasius", 9 will be output)

```
INPUT a
INPUT b
INPUT c
IF LENGTH(a)>LENGTH(b) AND LENGTH(a)>LENGTH(c) THEN
        OUTPUT a, LENGTH(a)
ELSE IF LENGTH(b)>LENGTH(c) AND LENGTH(b)>LENGTH(a) THEN
        OUTPUT b, LENGTH(b)
ELSE IF LENGTH(c)>LENGTH(a) AND LENGTH(c)>LENGTH(b) THEN
        OUTPUT c, LENGTH(c)
ELSE
        OUTPUT "None are longer than others"
END IF
```

15. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```

ANSWER:
0
10
12
14
16

16. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i=<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```

ANSWER:
0
10
12
14
16
18

17. What will be the output of the algorithm below?

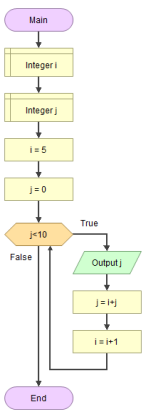
```
i=5
j=0
WHILE j<10 DO
        PRINT j
        j=i*2
        i=i+1
END WHILE
```

ANSWER: 0

18. What will be the output of the algorithm below?

```
i=5
j=0
WHILE i<10 DO
        PRINT j
        j=i+j
        i=i+1
END WHILE
```

ANSWER:
0
5



19. User inputs a word, the program outputs all the letters of that word

```
INPUT word
len=LENGTH(word)
FOR i=0 TO len
        PRINT word.SUBSTRING(i,1)
NEXT i  //could also use END FOR
```

20. User inputs a word, the program outputs the middle letter of that word, e.g. if "Jeremia" is input, it will output "e", if "Jeremy" is input, and the word length is an even number without a middle value, the program will divide in 2 and round DOWN and output "r".

```
INPUT word
len=LENGTH(word)
PRINT word.SUBSTRING(len DIV 2,1)    // DIV is integer division, eg. 7DIV2=3, "e" after "r"
```

21. User inputs a word, the program outputs all the letters of that word but every other letter (e.g. "monkey" would result in "m", "n", "e")

```
INPUT word
len=LENGTH(word)
FOR i=0 TO len STEP 2 //STEP 2 means the loop will count up in increments of 2
        PRINT word.SUBSTRING(i,1)
NEXT i  //could also use END FOR
```

22. User inputs a word, the program outputs the first and the last letters of that word

```
INPUT word
len=LENGTH(word)
first_char=word.SUBSTRING(0,1)  //means start at index 0, get 1 characters
last_char=word.SUBSTRING(len-1,1) //started from zero, so the last letter in a 6-char word is at position=5
PRINT first_char, last_char
```

23. User inputs a word, the program outputs the second and the second last letters of that word, minimal length is 5 characters.

```
INPUT word
len=LENGTH(word)
IF len<5 THEN
        OUTPUT “The word is too short”
ELSE
        sec_char=word.SUBSTRING(1,1)  //means start at index 1, get 1 characters
        seclast_char=word.SUBSTRING(len-2,1) //started from zero, so the second last letter in a 6-char word is at
        position=4
        PRINT sec_char, seclast_char
END IF
```

24. User inputs the price of an item and the banknote given. The program will output the change due to the customer. The price of the item must be less than the banknote given, otherwise the program outputs "Insufficient amound paid"

```
change=0
INPUT price
INPUT note
IF price>note THEN
        OUTPUT "Insufficient amount paid" // PRINT is also acceptable
ELSE IF price=note THEN
        OUTPUT "No change is due"
ELSE
        change=note-price
        OUTPUT change
END IF
```

25. The program counts up 0 to 100 and also outputs the counter and one of the words “Tic”, “Tac” alternatively, e.g. 0 “Tic”, 1 “Tac”, 2 “Tic”, 3 “Tac”, etc.

```
is_tic=True  /this variable will toggle between True and False at every iteration of the loop
FOR i=0 TO 100
        IF is_tic=True THEN
                OUTPUT STRING(i) + “ Tic”
                is_tic=False
        ELSE
                OUTPUT STRING(i)+ “ Tac”
                is_tic=True
        END IF
NEXT i
```

26. User inputs the price of an item and the banknote given (must be 1, 2, 5, 10, 20, 50 pound notes only).
The program will output the change needed.
The price of the item must be less than the banknote given, otherwise the program outputs "Insufficient amound paid"

```
change=0
INPUT price
INPUT note
IF note=1 OR note=2 or note=5 OR note=10 OR note=20 OR note=50 THEN
        IF price>note THEN
                OUTPUT "Insufficient amount paid" // PRINT is also acceptable
        ELSE IF price=note THEN
                OUTPUT "No change is due"
        ELSE
                change=note-price
                OUTPUT change
        END IF //this END IF belongs to the IF price>note
END IF  // this END IF belongs to the IF note=1 OR ...
```

27. Given a list of words, the program will output 3 random things in a sequence. e.g. "cactus","fish","goat"

```
words=Array["donkey","cactus","fish","goat"]
FOR i=0 TO 3
        r=RANDOM_INTEGER(0,SIZE(words))
        OUTPUT words[r]
NEXT i
```

28. Given a list of words, the program will ask user to input a word and will search its list for a match.
If match is found, the program will output the index at which the inputted word is found, otherwise
it would output "not found", e.g. in a list "donkey","cactus","fish","goat", if the user inputs "cactus"
the program should output "found at index 1", if "zebra" is input, the output is "zebra not found"

```
words=Array["donkey","cactus","fish","goat"]
i=0
found=False
INPUT w
WHILE i< 3 \\3 is the highest index in this list, could have used SIZE(words) instead
        IF w=words[i] THEN
                OUTPUT words[i] + " is found at index " + STRING(i)
                found=True
        ELSE
                i=i+1 //if not found, keep on searching while incrementing our counter i
        END IF
NEXT i //finished going through the array
IF found=False THEN  //realised there was no match
        OUTPUT w + " not found"
END IF
```

29. Given a list of words, the program will output 2 of them randomly, putting the word "and" between them and
" are the best friends!" after when priting, e.g. "cactus and goat are best friends"

```
message="" //empty string will be built from the parts needed
words=Array["donkey","cactus","fish","goat"]
r=RANDOM_INTEGER(0,SIZE(words))
message=words[r] //start with a first random word
r=RANDOM_INTEGER(0,SIZE(words))
message=message + " and " + words[r] +" are best friends"
OUTPUT message
```

30. A user is looking for a algorithm that would efficiently output the following phrase "and 1 and 2 and 3 and 4 and 5 and 6 and 7 and 8"

```
message=""
FOR i=1 TO 8
        message=message + " and " + STRING(i)
NEXT i //terminates the FOR loop and increments the counter i by 1
PRINT message
```

31. The user is looking for a algorithm that would efficiently output the following phrase "1 and 2 and 3 and 4 and 5 and 6 and 7 and 8"

```
message="1"
FOR i=2 TO 8
        message=message + " and " + STRING(i)
NEXT i //terminates the FOR loop and increments the counter i by 1
PRINT message
```

32. The user is looking for a algorithm that would efficiently output the following phrase "1 and 8 or 2 and 7 or 3 and 6 or 4 and 5 or 5 and 4 or 6 and 3"

```
message="1 and 8"
j=7
FOR i=2 TO 5
        message=message + " or " + STRING(i) + " and " + STRING(j)
        j=j-1
NEXT i //terminates the FOR loop and increments the counter i by 1
PRINT message
```

33. Write an algorithm that will create a string array from the names of all files in a folder, reporting back the names of the files as a numbered list.

```
FUNCTION LIST_FILES
        i=0
        WHILE NOT END_OF_FOLDER //find out how many files in a folder, END_OF_FOLDER is an oper sys word
                i=i+1
        END WHILE
        how_many_files=i
        files=STRING Array(0 TO how_many_files) //fixed size array for as many file names as files in the folder
        j=0
        WHILE NOT END_OF_FOLDER //fill the array with file names
                files[j]=READ FOLDER.FILE_NAME() //special command to get a file name
                j=j+1
        END WHILE

        FOR k=0 TO how_many_files
                OUTPUT k, files[k]
                j=j+1
        END FOR //another way to terminate a FOR loop
END FUNCTION
```

34. A known computer virus leaves this signature in infected files: "DeathJeffLOLZ". Write an algorithm that will open all files in a folder
and scan them for that signature, reporting back the names of the files infected.

```
FUNCTION SCAN_FILES  //FUNCTION and END FUNCTION replace the words BEGIN  and HALT
        WHILE NOT END_OF_FOLDER //fill the array with file names
                file=READ FOLDER.FILE_NAME() //special command to get a file name
                contents=file.READ_CONTENTS() //special command to get the file's contents as a string
                virus_found=False
                j=0
                WHILE virus_found=False DO
                        IF contents.SUBSTRING(j,LENGTH("DeathJeffLOLZ"))="DeathJeffLOLZ" THEN
        //contents.SUBSTRING(j,LENGTH("DeathJeffLOLZ")) - this will take parts of the file the same length (13 chars) as
                        //"DeathJeffLOLZ" and check if they are equal to the signature, e.g. if the file contained
                        //"£2kDeathJeffLOLZ", with values of j=0, 1, 2, ... we will look at "£2kDeathJeffL" - no match,
"2kDeathJeffLO"- no match
                        //"kDeathJeffLOL" - no match, "DeathJeffLOLZ" - match!
                                virus_found=True
                                OUTPUT file + " is infected"
                        END IF
                        j=j+1
                END WHILE //this terminates the loop that reads through a file's contents
        END WHILE //this terminates the loop that iterates through all files in a folder
END FUNCTION
```

35. Given a nested list (a 2D array) of names and numbers, display them as in columns separated by tabs.

```
LINE_BREAK="\n" \\constants
TAB="\t"
data=Array[["Bobby","3"],["Jemma","12"],["Cecil","20"] ]
message="" //empty string will hold the built-up parts of the list
FOR r=0 TO 3
        one_row="" //empty string to hold on row at a time, spaced with tabs
        FOR c=0 TO 2
                        one_row=one_row+TAB
        NEXT c
        message=message+LINE_BREAK+one_row
NEXT r
OUTPUT message
```

35B. Given an array a[6,9,2,4] output the second item

```
BEGIN test1
        OUTPUT a[1]
END
```

36. Given an array a[6,9,2,4] output the sum of the second item and the last items BEGIN test1

```
SET sum TO A[1]+A[3]  // sum:=A[1]+A[3]  // sum A[1]+A[3]
PRINT sum
END
```

37. Given an array a[6,9,2,4] output all items, then same but backwards

```
BEGIN test2
        FOR counter=0 TO LENGTH(a)
                OUTPUT a[counter]
        NEXT counter //means jump back to FOR line and counter:=counter+1
END
BEGIN test2 //and then back, using Stepping of -1
        FOR counter=LENGTH(a) TO 0 STEP -1
                OUTPUT a[counter]
        NEXT counter //means jump back to FOR line and counter:=counter-1
END
BEGIN test3 //yes another approach – using conditional loop
        counter:=0
        WHILE counter<LENGTH(a)
                OUTPUT a[counter]
                counter:=counter+1
        END WHILE //means jump back to FOR line and counter:=counter+1
END
```

38. Given an array a[6,9,2,4] output all items

```
BEGIN test4
        counter:=0
        WHILE counter<LENGTH(a)
                OUTPUT a[counter]
                counter:=counter+2
        END WHILE //means jump back to FOR line and counter:=counter+1
END
```

39. Given an array a[6,9,2,4] output every other item

```
BEGIN test2
        FOR counter=0 TO LENGTH(a) STEP 2
                OUTPUT a[counter]
        NEXT counter //means jump back to FOR line and counter:=counter+1
END
```

40. Given a[6,9,2,4], write the code that will swap the first and the third item [2,9,6,4],

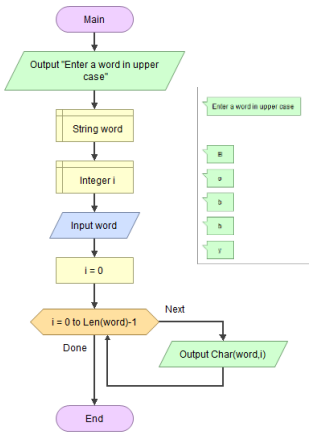
```
BEGIN swap1
        SET temp TO a[0]
        SET temp1 TO a[2]
        SET a[0] TO temp1
        SET a[2] TO temp
END
BEGIN swap2
        SET temp TO a[0]
        SET a[0] TO a[2]
        SET a[2] TO temp
END
```

41. Given a[6,9,2,4], write the code that will attempt to sort it

```
BEGIN swap2
        FOR j=0 TO LENGTH(a)-1
                FOR i=0 TO LENGTH(a)-1
                        IF a[i]>a[i+1] THEN
                                SET temp TO a[i]
                                SET a[i] TO a[i+1]
                                SET a[i+1] TO temp
                        END IF
                NEXT I
        NEXT j
END
```

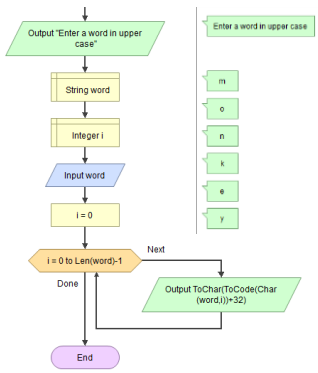
42. Create a flowchart and pseudocode for the following program: User enters a word and the program spells it one character at a time.

```
function Spell
   output "Enter a word in upper case"
   input word
   i = 0
   loop i from 0 to Len(word) – 1//slightly different syntax
      output Char(word, i)
   end loop
end function
```



43. User inputs a word in upper case, the algorithm converts it to lower case. This is based on the fact that in the ASCII table, upper case letters are 32 letters below the lower case ones.
Consider this printout from Python (in Pseudocode we can use ToChar(x) to convert an integer to an ASCII character and ToCode("a") to convert an ASCII character to an integer representing this character's ASCII position:

```
>>> ord("A")
65
>>> ord("a")
97
>>> ord("a")-ord("A"),ord("b")-ord("B")
(32, 32)
>>> chr(65),chr(97)
('A', 'a')
>>>
```



ANSWER:

```
function Main
   output "Enter a word in upper case"
   input word
   i = 0
   loop i from 0 to Len(word) - 1
      output ToChar(ToCode(Char(word, i)) + 32)
   end loop
end function
```

// http://www.flowgorithm.org/documentation/intrinsic-functions.htm

44. The program is to ask user for two words and then output the first two characters of the first word and the last two characters of the second word.

```
INPUT a
INPUT b
OUTPUT SUBSTRING(a,0,2)+SUBSTRING(b,LENGTH(b)-3,2) //LENGTH-2 – 2nd last char (-1 is the last)
```

45. Given a 1D integer array, the program will iterate (step through it) through the array and for any element larger than 10, it will subtract 10 from that element
and display the element (both adjusted and unadjusted).
e.g. if the array holds [3,6,1,23], the output will be 3, 6, 1, 13 because 23>10, so 23-10=13

```
n=Array[3,6,1,23]
FOR i=0 TO LENGTH(n)-1
        IF n>10 THEN
                n=n-10 //meaning whatever n is, reduce it by 10
        END IF
        OUTPUT n
NEXT i
```

46. Given a 1D integer array, the program will iterate (step through it) through the array and for any element smaller than 10, it will add 10 to that element
and display the element (both adjusted and unadjusted).
e.g. if the array holds [3,6,1,23], the output will be 3, 6, 1, 13 because 23>10, so 23-10=13

```
n=Array[3,6,1,23]
FOR i=0 TO LENGTH(n)-1
        IF n>10 THEN
                n=n-10 //meaning whatever n is, reduce it by 10
        END IF
        OUTPUT n
NEXT i
```

47. Given a 1D integer array, the program will iterate (step through it) through the array and output every element and its preceding and the
following element, if the array holds [3,6,1,23], the output will be "3 is first, 6 is after 3 and below 1, 1 is after 6 and before 23, 23 is last"

```
n=Array[3,6,1,23]
FOR i=0 TO LENGTH(n)-1
        IF i=0 THEN
                OUTPUT STRING(n[i]) + " is first"
        ELSE IF i<LENGTH(n)-2 THEN
                OUTPUT STRING(n[i])) + " is after " + STRING(n[i-1]) + " and before " + STRING(n[i+1])
        ELSE
                OUTPUT STRING(n[i]) + " is last"
        END IF
        OUTPUT n
NEXT i
```

48. Given a 1D integer array, the program will iterate (step through it) through the array and find the difference between an element and
the one immediately preceding it, then write those differences into another array, e.g. if the array holds [3,6,1,23], the resulting array will be [3,-5,22] from 6-3, 1-6, 23-1
The program will then output the resulting array of differences.

```
n=Array[3,6,1,23]
o=Array[0,0,0] //creates an initialised array to hold the worked out values/differences
FOR i=1 TO LENGTH(n)-1 #the first item has no difference, so we start with 2nd element
        x=n[i]-n[i-1]
        o[i-1]=x #we need to write to 0th element of the o array but our i is 1, so we need i-1
        OUTPUT x
NEXT i
```

49. Given a 1D integer array, the program will the length of the array (how many elements are in it), the sum of all integrs, and the first and the last elements

```
n=Array[3,6,1,23]
total=0
len=LENGTH(n)
FOR m=0 TO len
        total=total+n[m]
NEXT m
OUTPUT len, total, n[0],n[len-1]
```

50. Given a 1D integer array, the program will iterate (step through it) through the array and output the elements and whether they are odd or even.

```
n=Array[3,6,1,23]
FOR i=0 TO LENGTH(n)-1
        OUTPUT n
        IF n MOD 2 = 0 and n>1 THEN  //wouldn't apply to zero which we start from
                OUTPUT STRING(n) + " is even"
        ELSE
                OUTPUT STRING(n) + " is odd"
        END IF
NEXT i
```

51. Given a 1D integer array, user enters a number. The program returns "found" if that number is amongst the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "7" it will be "not found".

```
n=Array[3,6,1,23]
INPUT x
i=0
found=False
WHILE found=False AND i<LENGTH(n)
        IF n[i]=i THEN
                found=True
        ELSE
                i=i+1
        END IF
END WHILE
IF found=True THEN
        OUTPUT "found"
ELSE
        OUTPUT "not found"
END IF
```

52. Given a 1D integer array, user enters a number. The program returns "found" if that number is amongst the FIRST half of the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "1" it will be "not found" because [3,6] is the first
part of the array and it doesn't contain 1. We can find the first half of the array by searching up to LENGTH(n) divided into 2 and rounded down (integer DIV).

```
n=Array[3,6,1,23]
INPUT x
i=0
found=False
WHILE found=False AND i<(LENGTH(n) DIV 2)
        IF n[i]=i THEN
                found=True
        ELSE
                i=i+1
        END IF
END WHILE
IF found=True THEN
        OUTPUT "found"
ELSE
        OUTPUT "not found"
```

53. Given a 1D integer array, user inputs a number. The program returns that number's index if that number is amongst the elements of the array, or "not found" if it's not.
e.g. if the array is n=Array[3,6,1,23] and the user inputs 1, the output will be "found", if they enter "7" it will be "not found".

```
n=Array[3,6,1,23]
INPUT x
i=0
found=False
WHILE found=False AND i<LENGTH(n)
        IF n[i]=i THEN
                found=True
        ELSE
                i=i+1
        END IF
END WHILE
IF found=True THEN
        OUTPUT i //because we stopped incrementing the counter, it will hold the value of the x's index
ELSE
        OUTPUT "not found"
END IF
```

54. User has a file called “pups.txt” that has the following format:

```
Bobby
Johnny
Amir
Cecilius
```

Write a program that will read this file into a 1D array of string elements. The array will hold up to 20 elements.

```
s=Array[0 TO 20] AS STRING
OPEN FILE “pups.txt" FOR READING AS FILE1
i=0
WHILE NOT END OF FILE
        s[i]=FILE1.READLINE()
NEXT i
```

55. User has a file called “pups.csv” that has the following format:

```
Bobby, 33
Johnny, 22
Amir, 56
Cecilius, 78
```

Write a program that will read this file into a 2D array of string elements. The array will hold up to 20 elements. Output the name with the highest number of points. E.g. Cecilius

```
s=Array[0 TO 20, 0 TO 1] AS STRING  //even digits will be stored as strings
OPEN FILE “pups.csv" FOR READING AS FILE1
j=0
WHILE NOT END OF FILE
        s[j,0],s[j,1]=FILE1.READLINE().SPLIT(“,”) #slightly Pythonic but explains the process well enough
NEXT j
Winner=s[0,0]
maxPoints=s[0,1]
FOR i=0 TO LENGTH(s)
        IF s[i,1]>maxPoints THEN
                maxPoints=s[i,1]
                Winner=s[i,0]
        END IF
NEXT I
OUTPUT Winner
```

56. User inputs a word and the program will display every other letter in alternating case: upper/lower, e.g. “Hello people” will become “HeLlo PeOpLe”

```
INPUT w
Up=True
FOR m=0 TO LENGTH(w)
	IF Up=True THEN
		OUTPUT UPPER_CASE(w[m])
		Up=False //toggle the Up variables between True/False
	ELSE
		OUTPUT LOWER_CASE(w[m])
		Up=True //toggle the Up variables between True/False
	END IF
NEXT m
```

57. A teacher needs to count all pupils who did their homework and all those who didn't and output the totals for both in the end.

```
end_data=-999 //rogue value, used to indicate the end of data entry
keen=0 #an integer count of those with homework
naughty=0 #an integer count of those who didn't do their homework
done=False
WHILE done=False DO
	OUTPUT "Did you do you homework? y/n "
	INPUT homework
	IF homework="y" THEN
		keen=keen+1
	ELSE IF homework="n" THEN
		naughty=naughty+1
	ELSE IF homework="-999" THEN
		done=True
	ELSE
		OUTPUT "Stop mumbling and no excuses!"
	END IF
END WHILE //once everybody has been counted...
OUTPUT keen + " did their homework"
OUTPUT naughty + " didn't do their homework"
```

58. A teacher needs to count all pupils who did their homework and all those who didn't and output the totals for both in the end, while entering those who didn't into a special detention list.

```
end_data=-999 //rogue value, used to indicate the end of data entry
keen=0 #an integer count of those with homework
naughty=0 #an integer count of those who didn't do their homework
done=False
detentions=Array(1 TO 100) AS STRING //assume no more than 100 pupils
WHILE done=False DO
	OUTPUT "Did you do you homework? y/n "
	INPUT homework
	IF homework="y" THEN
		keen=keen+1
	ELSE IF homework="n" THEN
		INPUT name
		detentions[naughty]=name
		naughty=naughty+1
	ELSE IF homework="-999" THEN
		done=True
	ELSE
		OUTPUT "Stop mumbling and no excuses!"
	END IF
END WHILE //once everybody has been counted...
OUTPUT keen + " did their homework"
OUTPUT naughty + " didn't do their homework"
FOR i=0 TO SIZE(detentions)
	OUTPUT STRING(i) + ". " + detentions[i] //will display names as bulleted list
NEXT I
```

59. Write a program that will count an approximate number of characters in a book, given the input from a user of average characters per word, avg words per line, and avg lines per page, as well as the number of pages, e.g. if there are 200 pages, with average of 30 lines per page and average of 10 words per line and 6 characters per word, we should get 360,000 characters in that book.

```
INPUT pages
INPUT lines
INPUT words
INPUT chars_per_line
Result=pages*lines*words*chars_per_line
OUTPUT Result
```

60. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program iterates through both arrays, displaying the elements side by side, e.g. given
["Bob","Cecil","Nora","Amir"]
as names, and [45,34,89,100] as integers, the program will output "Bob: 45 points", "Cecil: 34 points", etc.

```
s=Array["Bob","Cecil","Nora","Amir"]
n=Array[45,34,89,100]
FOR i=0 TO LENGTH(s)-1 //could have used LENGTH(n), too as they have the same length
        OUTPUT s[i] + ": " + STRING(n[i]) + " points"
NEXT i
```

61. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program iterates through both arrays, displaying the elements side by side, e.g. given
["Bob","Cecil","Nora","Amir"]
as names, and [45,34,89,100] as integers, but only for those who have over 40 points, so the program will output "Bob - 45 points", "Nora - 89 points", etc, leaving out Cecil who scored fewer than 40 points.

```
s=Array["Bob","Cecil","Nora","Amir"]
n=Array[45,34,89,100]
FOR i=0 TO LENGTH(s)-1 //could have used LENGTH(n), too as they have the same length
        IF n[i]>40 THEN
                OUTPUT s[i] + ": " + STRING(n[i]) + " points"
        END IF
NEXT i
```

62. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The user inputs a name, the program finds the matching name in the first array, remembers its index,
then it outputs the element from the integer array that is at the same index as the match.
If no match is found, the error code of "-999" is output.

```
s=Array["Bob","Cecil","Nora","Amir"]
n=Array[45,34,89,100]
found=False
at_index=-999 //rogue aka nonsense value that is used as a default - easy to see if it never changed then
INPUT criteria //criteria is a word commonly used as "whatever we are searching for"
FOR i=0 TO LENGTH(s)-1 //could have used LENGTH(n), too as they have the same length
        IF n[i]=criteria THEN
                found=True
                at_index=i
                EXIT FOR //breaks the loop, tehcnically not as clever as using WHILE loops
        END IF
NEXT i
IF found=True THEN
        OUTPUT n[at_index]
ELSE
        OUTPUT at_index
END IF
```

63. A user has a list of prices [12,59,23,101] and would like to see each prices with and without VAT, with and without 10% early bird discount, and with and without the 15% surcharge for overseas customers.

```
N=Array [12,59,23,101]
Factors=Array[1,1.2,1,0.9,1,1.15]
FOR j=0 TO LENGTH(N)
        FOR k=0 TO LENGTH(Factors)
                OUTPUT N[j]*Factors[k]
        NEXT k
NEXT j
```

64. A program needs to write “Bob was here” into a text file called “memories.txt”

```
OPEN FILE “memories.txt" FOR WRITING AS FILE1
FILE1.WRITE(“Bob was here”)
CLOSE AND SAVE FILE1
```

A program needs to ask a user to write down 3 of their favourite Pikachu characters and write them into a text file called “p.txt”

```
OPEN FILE “p.txt" FOR WRITING AS FILE1
FOR i=0 TO 3
        INPUT pname
        FILE1.WRITE(pname+LINE_BREAK)
NEXT i
CLOSE AND SAVE FILE1
```

65. A program needs to ask a user to write down any number of their favourite Pikachu characters and write them into a text file called “p.txt”

```
OPEN FILE “p.txt" FOR WRITING AS FILE1
Finished=False
END_VALUE=-999
WHILE Finished=False //have to use conditional loops as we don’t know how many characters are there
        INPUT pname
        IF pname=END_VALUE THEN
                Finished=True
        ELSE
                FILE1.WRITE(pname + LINE_BREAK)
        END IF
END WHILE
CLOSE AND SAVE FILE1
```

66. Given an array of strings, the program needs to write all of its elements to a file called "data.txt"

```
s=Array["Bob","Cecil","Nora","Amir"]
OPEN FILE "data.txt" FOR WRITING AS FILE1
FOR i=0 TO LENGTH(s)-1
        FILE1.WRITE(s[i]+LINE_BREAK)
NEXT i
CLOSE AND SAVE FILE1
```

67. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program needs to write each element of the names array, followed by a comma, followed by an element of the integer array with the same index, to each new line in a file called "data.csv"

```
s=Array["Bob","Cecil","Nora","Amir"]
n=Array[45,34,89,100]
OPEN FILE "data.csv" FOR WRITING AS FILE2
FOR i=0 TO LENGTH(s)-1 //could have used LENGTH(n), too as they have the same length
        FILE1.WRITE(s[i]+","+STRING(n[i])+LINE_BREAK)
NEXT i
CLOSE AND SAVE FILE2
```

68. We have two arrays, of 4 elements each. The first array contains names,
the other array contains integers representing points earned by each of the names in the other array.
The program needs to write each element of the names array, followed by a comma, followed by an element of the integer array with the same index, to each new line in a file called "data.csv" BUT ONLY IF the integer element is over 40, e.g. Cecil will not be written as he only has 34 points.

```
s=Array["Bob","Cecil","Nora","Amir"]
n=Array[45,34,89,100]
OPEN FILE "data.csv" FOR WRITING AS FILE2
FOR i=0 TO LENGTH(s)-1 //could have used LENGTH(n), too as they have the same length
        IF n[i]>40 THEN
                FILE1.WRITE(s[i]+","+STRING(n[i])+LINE_BREAK)
        END IF
NEXT i
CLOSE AND SAVE FILE2
```

68. Given a table of integers, the program will ask user for a column number and return the total of that column, e.g.

```
4       7       12      10
2       1       8       9
14      9       3       7
```

if the user inputs "0", they will see 20 (which is 4+2+14)

```
n=Array[[4,7,12,10],[2,1,8,9],[14,9,3,7] ]
INPUT column
total=0
FOR i=0 TO 3
        total=total+n[i,column]  //row first, column second
NEXT i
OUTPUT total
```

69. Given a table of integers, the program will ask user for a row number and return the total of that row, e.g.

```
4       7       12      10
2       1       8       9
14      9       3       7
```

if the user inputs "0", they will see 33 (which is 4+7+12+10)

```
n=Array[[4,7,12,10],[2,1,8,9],[14,9,3,7] ]
INPUT row
total=0
FOR i=0 TO 3
        total=total+n[total,i]
NEXT i
OUTPUT total
```

70. The user wants to write two procedures which are selectable via a menu from the main screen. e.g. when the program runs, the user
might see "Press 1 to hear praise, 2 to hear a put-down" where praise and put-down are defined in their respective procedures

```
Function praise
        OUTPUT "You are the amazingnest!"
END function
Function put_down
        OUTPUT "You should be doing better in everything!"
END function
Function main
        carry_on=True
        WHILE carry_on=True
                OUTPUT "Press 1 to hear praise, 2 to hear a put-down"
                INPUT choice
                IF choice="1" THEN
                        CALL praise
                ELSE IF choice="2" THEN
                        CALL put_down
                ELSE
                        OUTPUT "That was not a valid choice"
                END IF
        END WHILE
END function
```