Algorithms and Programming
Laboratory number 01

Exercise 01

Write a C program able to:
- Read an array of integer values of size DIM, where DIM is a pre-defined constant.
- Find and print-out the longest ascending sub-sequence on integer values within the array.

Example

Let DIM be equal to 15, ant let the array be the following one:

2 3 4 -1 -10 5 6 8 11 -9 8 -10 9 3 0

The program has to print-out the sub-sequence:

-10 5 6 8 11

Exercise 02

Write a C program able to:
- Read two integer values N1 and N2.
- Check whether N1<=DIM and N2<=DIM (DIM is a pre-defined constant value).
- Read from standard input an array v1 of N1 integer values
- Compute an array v2 of N1 integer values, where each v2[i] is equal to the arithmetic average of
  v1[i-N2], v1[i-N2+1], ..., v1[i-1], v1[i], v1[i+1], ..., v1[i+N2]
  if they exist.
- Print-out v2.

Example

Let us suppose DIM = 100, N1 = 10, N2 = 2, ant let v1 be the following one:

0 2 3 4 -1 -10 5 1 8 3

Each element of v2[i] has to be equal to the average of at most 5 (2 on the left of v1[i], 2 on the right of v1[i], and v1[i] itself) elements "around" v1[i], e.g.:

v2[0] = (0 + 2 + 3) / 3 = 1.67
v2[1] = (0 + 2 + 3 + 4) / 4 = 2.25
v2[2] = (0 + 2 + 3 + 4 + (-1)) / 5 = 1.60
v2[3] = (2 + 3 + 4 + (-1) + (-10)) / 5 = -0.40
etc.

As a consequence, the program has to print-out the sub-sequence:
1.67 2.25 1.60 -0.40 0.20 -0.20 0.60 1.40 4.25 4.00

Exercise 03

Write a C program able to:
- Read a matrix m1 of R rows and C columns of integer values
- Compute a matrix m2 of R rows and C columns such that
  - m2[i][j] is equal to zero
    if m1[i][j] is equal to zero.
  - m2[i][j] is the factorial number of -m1[i][j]
    if m1[i][j] is a negative number.
  - m2[i][j] is the smallest power of 10 larger than m1[r][j]
    if m1[i][j] is a positive number.

Example

Let R and C be equal to 3 and 5 respectively.
Let m1 be the following:

9	23	155	-1	33
0	-2	-4	8	-6

20 -1 0 0 499

m2 has to be:

10 100 1000 1 100
0 2 24 10 720
100 1 0 0 1000

Exercise 04

Write a C program able to compute the product of two matrices
containing real values.
The program has to proceed as follows:
1. It defines two matrix's m1 and m2 of "physical" size equal to
   MAX_DIM rows and MAX_DIM columns, where MAX_DIM is a pre-defined
   constant.
2. It reads the "logical" (actual) size of the two matrices:
   r1 rows and c1 columns for m1
   and
   r2 rows and c2 columns for m2.
3. It verifies whether r1, c1, r2, and c2 are smaller than MAX_DIM,
   and whether c1==r2 (i.e., it is possible to compute the product).
4. It reads m1 and m2 from standard input.
5. It computes the product
   m3 = m1 x m2
   with m3 having r1 rows and c2 columns.
6. It prints-out m3 on standard output.

Example

If m1 is the following on (with 2 rows and 3 columns)
 1.00 2.00 3.00
 4.00 5.00 6.00
and m2 is the next one (with 3 rows and 2 columns)
 1.00 2.00
 3.00 4.00
 5.00 6.00
the output matrix must be the following:
22.00 28.00
49.00 64.00