Software Engineering Methods

Laboratory work 1

Abakirov Nursultan

SE (english) 1-16

1.Problem statement

To find max and min element of the array and their indices

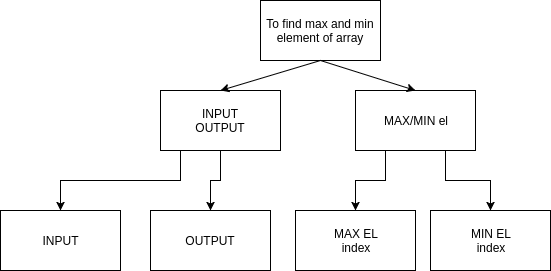
2.Requirements

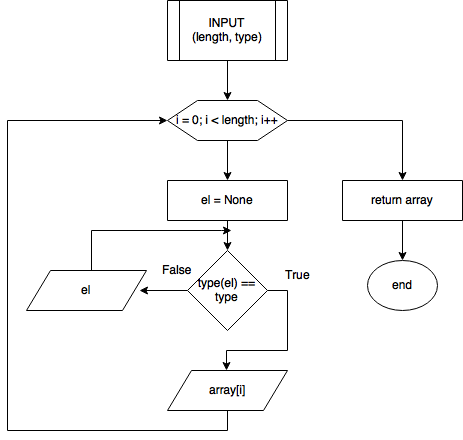
Length of the array, type of the array, array elements.

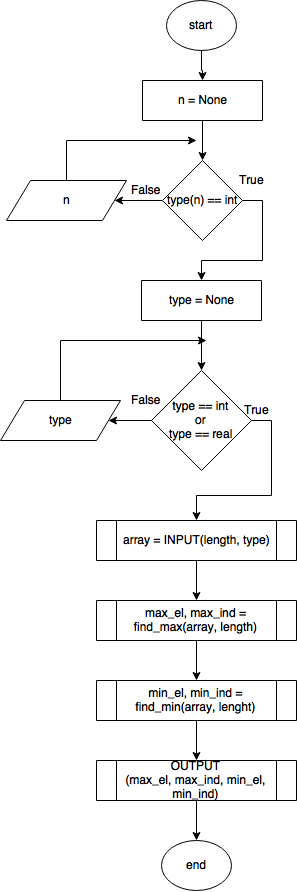
3.Design

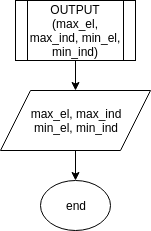
To find the max and min elements of the array we have to know length, type and elements. The available types are integer and real. First we are asking for the length of the array. To verifying inputed element for correctness, trying to convert element to integer. If validation failed printing a user message ‘invalid input’, otherwise asking for type of the array. Checking input for ‘i’ - for integer and ‘r’ - for real. If inputed element didn’t match to the conditions, printing a user message ‘invalid input’. Then asking for the elements and checking their types with previously inputed type. After initialization, configuring min element end max element to the first element of the array. Then passing through elements and comparing them with max and min elements. Then printing the max and the min elements of the array and their indices.

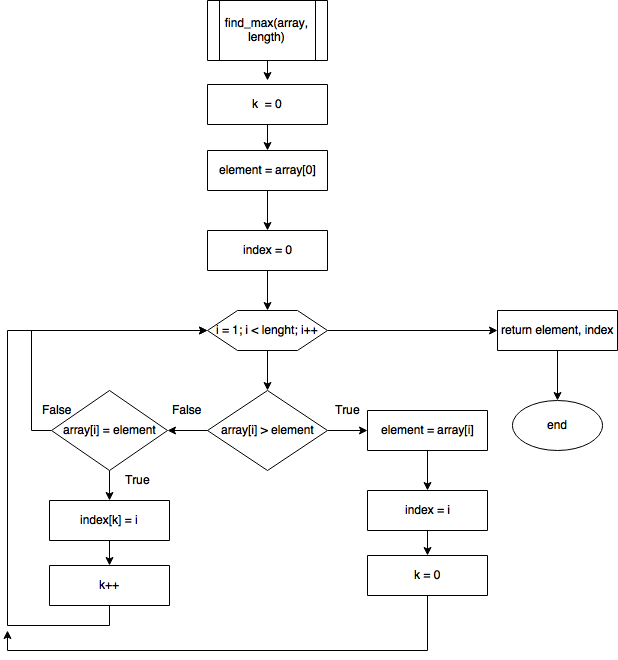
Decomposition

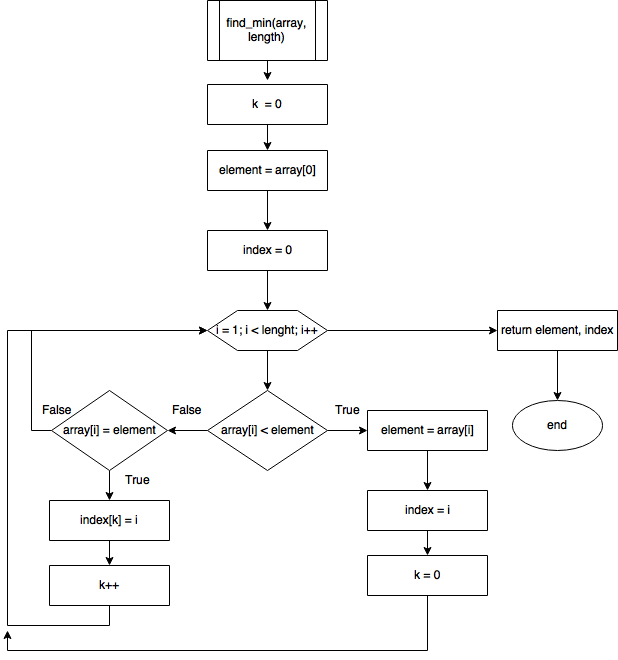












**Tests 1**

input:

5

i

1 2 3 4 5

output:

inputed array: [1, 2, 3, 4, 5]

max element: 5

max element index: 4

min element: 1

min element index: 0

**Test 2**

input:

8

r

2.6 1.5 6.9 8.2 1.2 0.3 7.5 9.8

output:

inputed array: [2.6, 1.5, 6.9, 8.2, 1.2, 0.3, 7.5, 9.8]

max element: 9.8

max element index: 7

min element: 0.3

min element index: 5

**Test 3**

input:

4

i

0 0 0 0

output:

inputed array: [0, 0, 0, 0]

max element: 0

indexes of max elements: [0, 1, 2, 3]

min element: 0

indexes of min elements: [0, 1, 2, 3]

CODE

types = dict(i=int, r=float)

def init\_array(length, arr\_type):

array = []

for i in range(length):

element = None

while type(element) is not arr\_type:

element = input('input {} element: '.format(i + 1))

try:

element = arr\_type(element)

array.append(element)

except:

print('invalid type of inputed element')

return array

def find\_max(arr):

max\_el = arr[0]

max\_in = []

for i, el in enumerate(arr):

if el > max\_el:

max\_el = el

max\_in = [i]

elif el == max\_el:

max\_in.append(i)

return (max\_el, max\_in)

def find\_min(arr):

min\_el = arr[0]

min\_in = []

for i, el in enumerate(arr):

if el < min\_el:

min\_el = el

min\_in = [i]

elif el == min\_el:

min\_in.append(i)

return (min\_el, min\_in)

def print\_arr(arr, max\_el, max\_in, min\_in, min\_el):

print('\n\ninputed array: ', arr)

print('max element:\t{}'.format(max\_el))

if len(max\_in) == 1:

print('max element index:\t{}'.format(max\_in[0]))

else:

print('indexes of max elements:\t{}'.format(max\_in))

print('min element:\t{}'.format(min\_el))

if len(min\_in) == 1:

print('min element index:\t{}'.format(min\_in[0]))

else:

print('indexes of min elements:\t{}'.format(min\_in))

def main():

lenght = None

while type(lenght) != int:

lenght = input('input length of the array\n')

try:

lenght = int(lenght)

except:

print('\ninvalid input, only integers')

arr\_type = None

while True:

arr\_type = input('input type of the array\n\ti - integer\n\tr - real\n')

if arr\_type == 'i' or arr\_type == 'r':

break

else:

print('\ninvalid input, allowed only -i -r')

arr = init\_array(lenght, types[arr\_type])

max\_el, max\_in = find\_max(arr)

min\_el, min\_in = find\_min(arr)

print\_arr(arr, max\_el, max\_in, min\_in, min\_el)

if \_\_name\_\_ == '\_\_main\_\_':

main()