Design Note of Program 1

**Requirement**:

A program that reads 20 numbers (integers) from the keyboard, prints the numbers to the console printer, requests a number from the user, and searches the 20 numbers read in for the number closest to the number entered by the user. Print the number entered by the user and the number closest to that number. Your numbers should not be 1…10, but distributed over the range of 0 … 65,535. Therefore, as you read a character in, you need to check it is a digit, convert it to a number, and assemble the integer.

**Steps to accomplish the requirement**:

1. read 20 numbers from the keyboard;
2. convert the numbers to digit and save them to memory
3. print the numbers to console printer
4. read 1 number to search from the keyboard;
5. convert the number to digit and save them to memory
6. print the numbers to console printer
7. search for the nearest number in the 20 numbers
8. print the number searched and the number to search to console printer

**Input data:**

1. A string contains the totally 20 + 1 numbers, divided by one (and only one) space
2. The string is read from the keyboard device by the “IN” instruction.

**Output data:**

1. A string contains the 20 numbers
2. A string contains a number to search
3. A sting contains the search result

**Memory map:**

1. p\_idata, type short, size = 21, start address is 20, stores the 20 numbers and the number to search;
2. p\_idiff, type short, size = 20, start address is 45, stores the difference of the 20 numbers to the number to search.
3. p\_sdata, type string, size = 120, start address is 96, p\_sdata = 64; stores the input string.

**Algorithm**:

1. read user input from keyboard

Lable1: IN 0, 0 # read from keyboard to r0

JZ 0, 2, 0 # if 0, end reading, jump to x2

STR 0, 3, 0 # save to Mem[x3]

LDX 3, 1 # x3 = x3 + 1

JMA 1, 0 # jump to Label1

1. Convert user input to integer

Lable3:

LDR 0, 0 ,6 r0 <- 0

Lable6:

LDR 2, 0 ,3,0 r2 <- Mem[x3]

JZ 2, 2 ,23 R2 == 0? goto lable5 (end)

LDX 3, 1 x3++

SMR 2, 0 ,9 if R2 == space, goto Lable4

JZ 2, 2 ,18

LDR 2, 0 ,6 r0 <- r0 \* r2(10) + int(mem[11])

AIR 2, 10

MLT 0, 2

LDR 2, 0 ,11

SMR 2, 0 ,10 r2 - mem[10] = r2 - 48

AMR 1, 0 ,2

JMA 2, 1 go to Lable6

Lable4:

STR 0, 0 ,8, I Store the convert result to M[mem[8]]

LDR 0, 0 ,8 mem[8]++;

AIR 0, 1

STR 0, 0 ,8

JMA 2, 0 goto Lable 3 to convert next number

1. Convert user input to integer

Lable5:

LDR 3, 0 ,6

AIR 3, 20

STR 3, 0 ,13 mem[13] <- &data[0]

AIR 3, 19

STR 3, 0 ,14 mem[14] <- &data[19]

LDX 3, 13 x3 <- &data[0]

LDR 3, 3 ,20 r3 <- &data[20] (target)

STR 3, 0 ,12 mem[12] <- r3

LDR 3, 0 ,6 x2 <- L7 (32\*10 + 18 = 338)

AIR 3, 10

SRC 3, 5 ,1, 1

AIR 3, 18

STR 3, 0 ,7

LDX 2, 7

Lable7:

LDR 2, 3 ,0 R2 <- data[i]

SMR 2, 0 ,12 data[i] - target

JGE 2, 2 ,5 data[i] >= target goto Label8

LDR 2, 0 ,12, I data[i] < target: r2 <- target

SMR 2, 3 ,0

Lable8:

STR 2, 3 ,25 save result to data[i + 25]

LDR 1, 0 ,13 x3++

AIR 1, 1

STR 1, 0 ,13

LDX 3, 13

LDR 1, 0 ,14 &data[19] >= &data[x3] goto Lable7

SMR 1, 0 ,13

JGE 1, 2 ,0

LDR 3, 0 ,6 mem[13]<- &data[0]

AIR 3, 20

STR 3, 0 ,13

STR 3, 0 ,15 mem[15]<- &data[0] minimum difference index

AIR 3, 19 mem[14]<- &data[19] max addr

STR 3, 0 ,14

LDX 3, 13 x3<- &data[0]

LDR 1, 3 ,25 r1<-data[i+25]

STR 1, 0 ,16 mem[16] <- r1 min diff value

LDR 3, 0 ,6 x2 <- L9 (32 \*11 + 14 = 366)

AIR 3, 11

SRC 3, 5 ,1, 1

AIR 3, 14

STR 3, 0 ,7

LDX 2, 7

Lable9:

LDR 1, 3 ,25 data[x3+25] - mem[16]

SMR 1, 0 ,16

JGE 1, 2 ,6 data[i] >= min value, goto L10

STX 3, 15 update minimun diff index to mem[15]

LDR 0, 3 ,25 Update minimun diff value to mem[16]

STR 0, 0 ,16

Label10:

LDR 1, 0 ,13 x3++

AIR 1, 1

STR 1, 0 ,13

LDX 3, 13

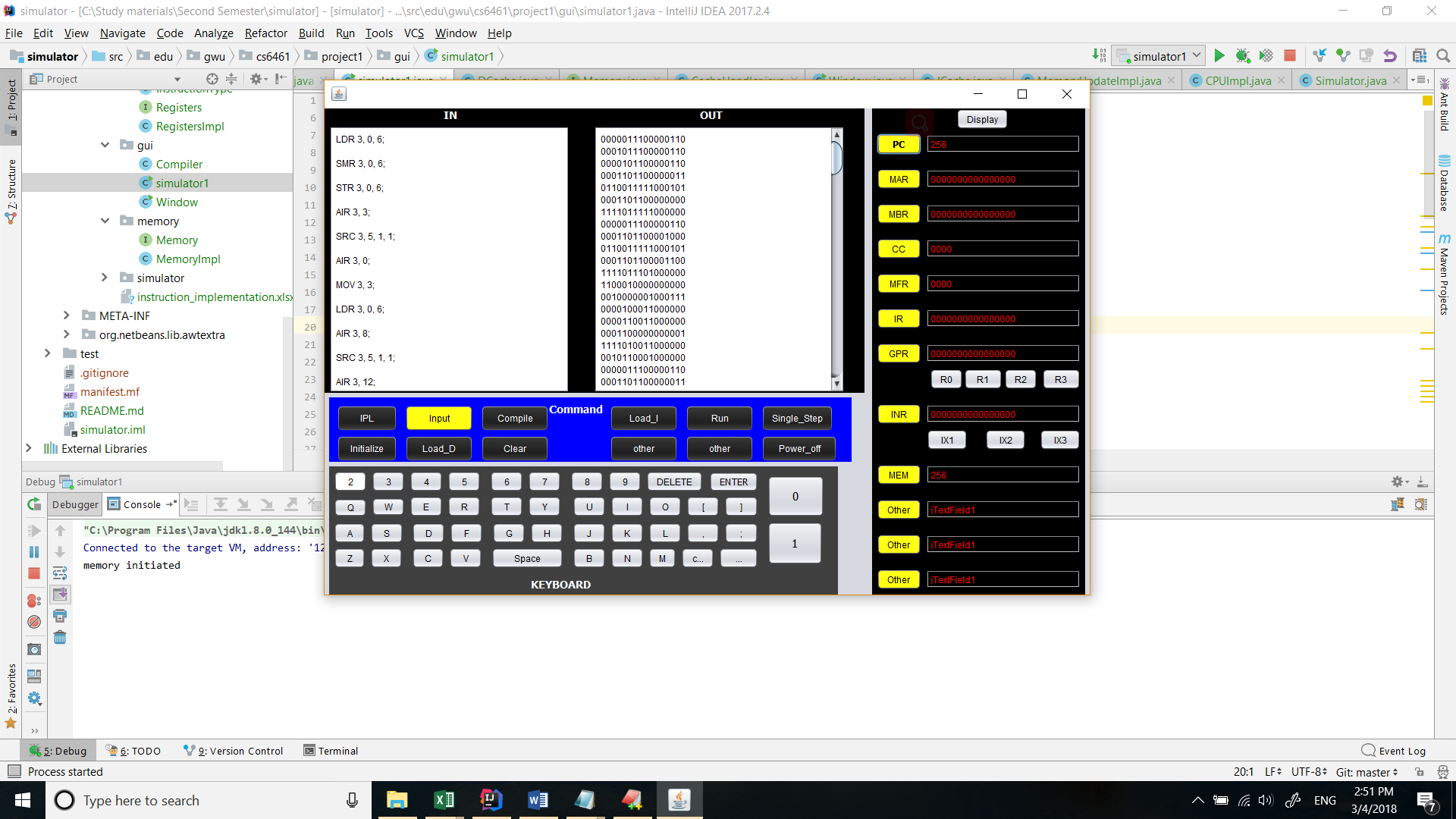
LDR 2, 0 ,14 Mem[14] (end) >= x3? goto Lable9

SMR 2, 0 ,13

JGE 2, 3 ,0

**Program1 Running example:**

**Input the program to the input area, compile it. Store the program in Address 0x100.**

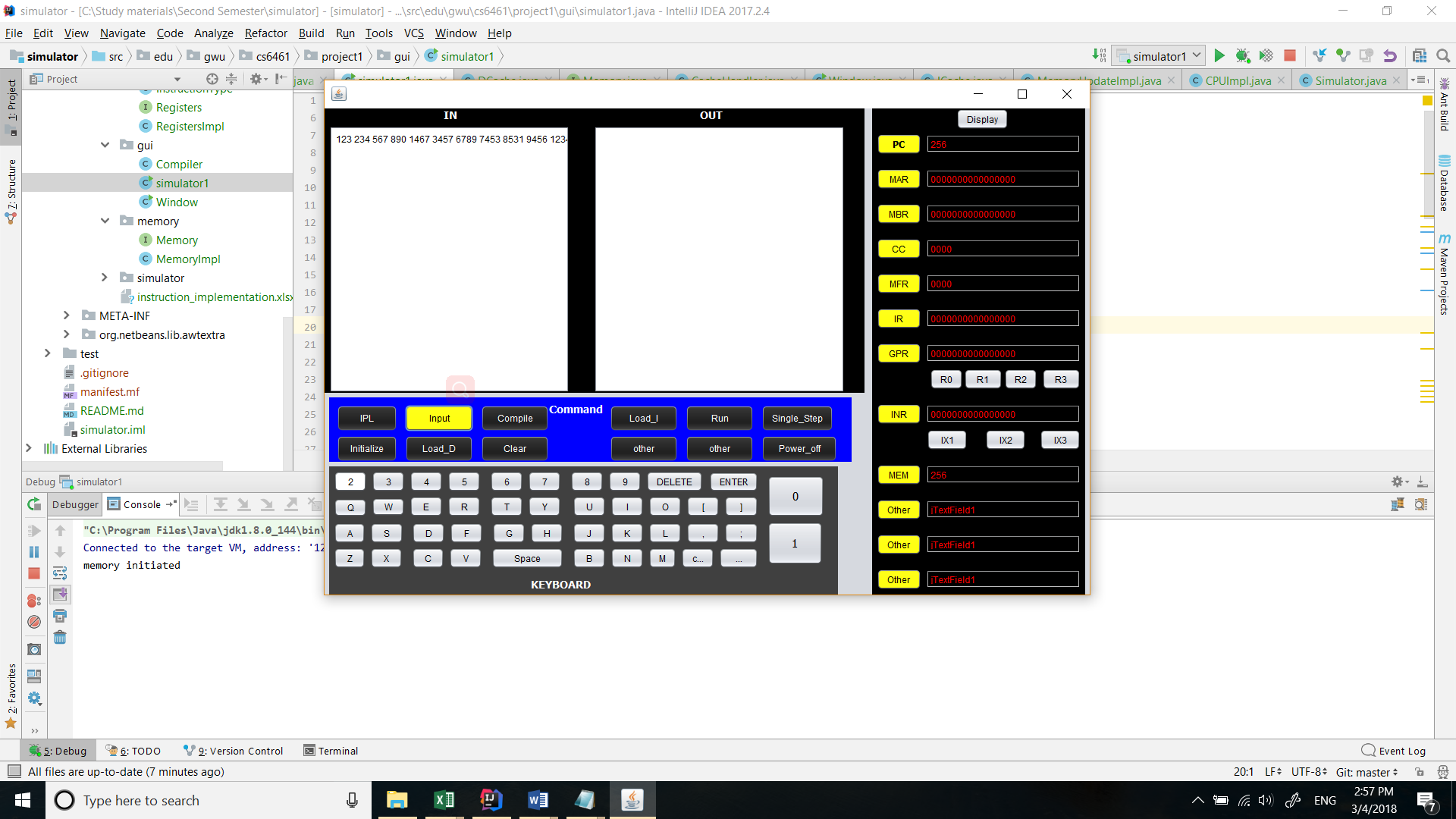


Set the PC value as decimal value 256 (0x100).

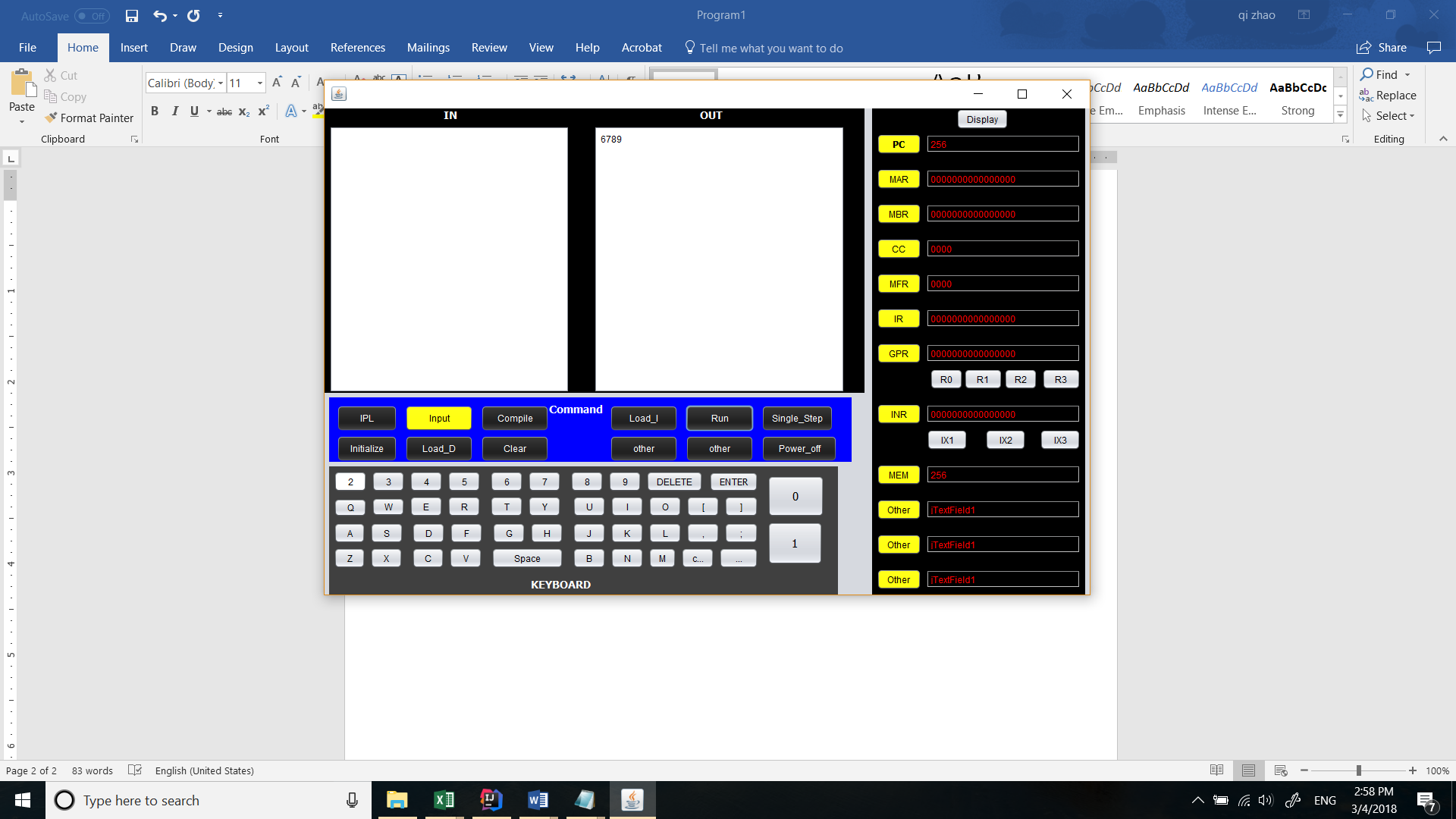
**Then clear the contents of Input area and output area and Input 21 numbers:**

123 234 567 890 1467 3457 6789 7453 8531 9456 12345 23457 456 789 345 456 865 78943 1345 9876 6785

The numbers should be separated by a space character. There must be a space character after the last number.



Then hit run:



Bingo! We got the correct answer!