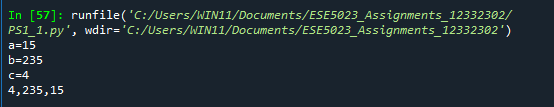
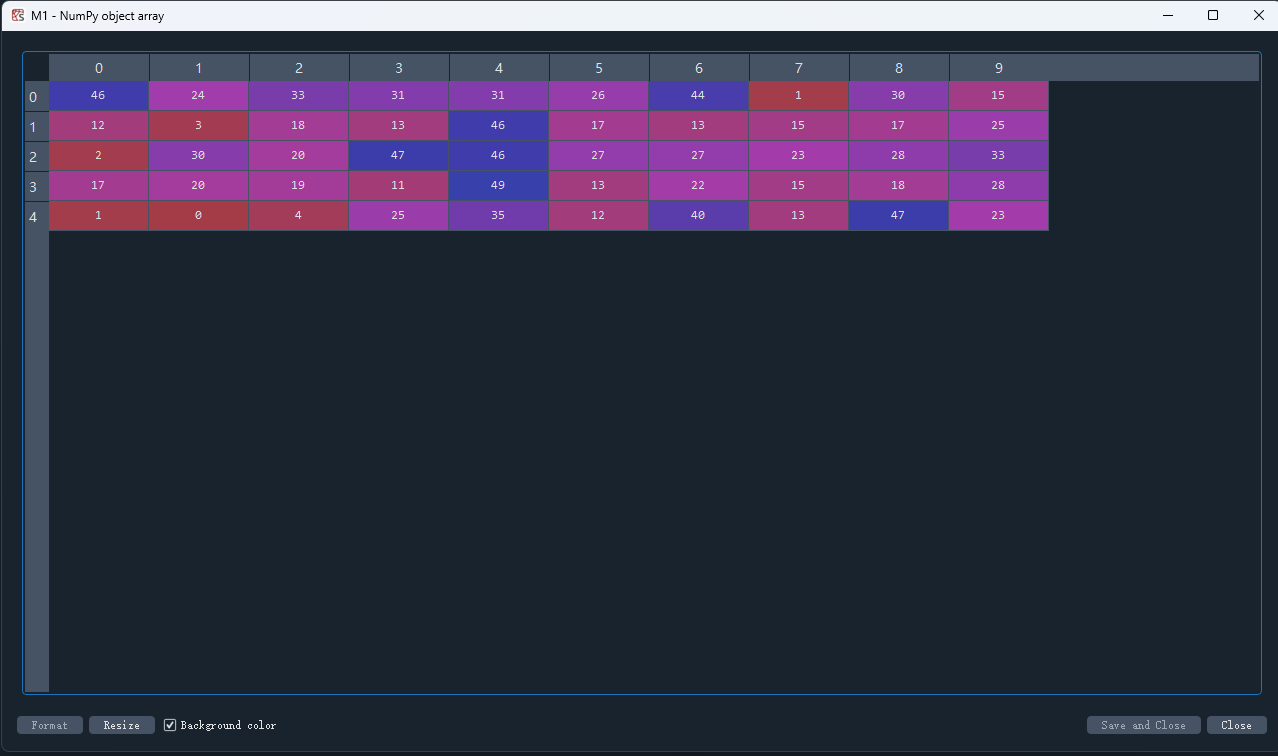
PS1\_1 :random a,b,c value output

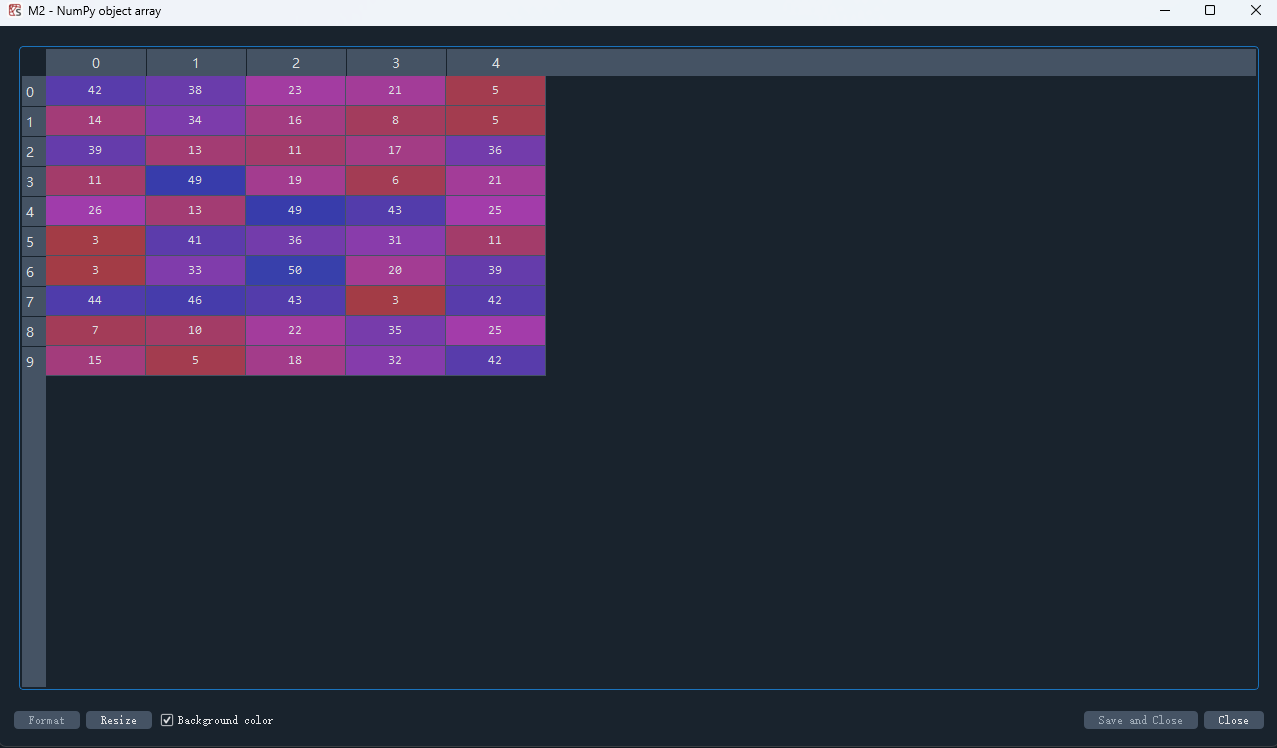
Just define a function named Print\_values,and use if\_else commands to finish the work

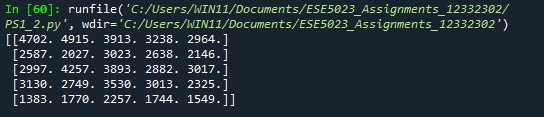


PS1\_2:

Just make two matrices and use function randint from random package to establish the matrix as asked. And for multiplication, I construct a 5 plus 5 matrix and use double for loop to calculate each element one after one.







PS1\_3 :100th and 200th line of Pascal Triangle

First initialize the triangle by type in the value of the first twp lines by hand. And other values can be calculated according to results of previous line. So for line x, first construct an array of n, and the xth element goes to the sum of (x-1)th and xth element of the previous line.

**100th**

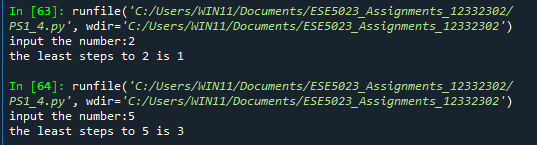
[1, 99, 4851, 156849, 3764376, 71523144, 1120529256, 14887031544, 171200862756, 1731030945644, 15579278510796, 126050526132804, 924370524973896, 6186171974825304, 38000770702498296, 215337700647490344, 1130522928399324306, 5519611944537877494, 25144898858450330806, 107196674080761936594, 428786696323047746376, 1613054714739084379224, 5719012170438571889976, 19146258135816088501224, 60629817430084280253876, 181889452290252840761628, 517685364210719623706172, 1399667836569723427057428, 3599145865465003098147672, 8811701946483283447189128, 20560637875127661376774632, 45764000431735762419272568, 97248500917438495140954207, 197443926105102399225573693, 383273503615787010261407757, 711793649572175876199757263, 1265410932572757113244012912, 2154618614921181030658724688, 3515430371713505892127392912, 5498493658321124600506947888, 8247740487481686900760421832, 11868699725888281149874753368, 16390109145274293016493707032, 21726423750712434928840495368, 27651812046361280818524266832, 33796659167774898778196326128, 39674339023040098565708730672, 44739148260023940935799206928, 48467410615025936013782474172, 50445672272782096667406248628, 50445672272782096667406248628, 48467410615025936013782474172, 44739148260023940935799206928, 39674339023040098565708730672, 33796659167774898778196326128, 27651812046361280818524266832, 21726423750712434928840495368, 16390109145274293016493707032, 11868699725888281149874753368, 8247740487481686900760421832, 5498493658321124600506947888, 3515430371713505892127392912, 2154618614921181030658724688, 1265410932572757113244012912, 711793649572175876199757263, 383273503615787010261407757, 197443926105102399225573693, 97248500917438495140954207, 45764000431735762419272568, 20560637875127661376774632, 8811701946483283447189128, 3599145865465003098147672, 1399667836569723427057428, 517685364210719623706172, 181889452290252840761628, 60629817430084280253876, 19146258135816088501224, 5719012170438571889976, 1613054714739084379224, 428786696323047746376, 107196674080761936594, 25144898858450330806, 5519611944537877494, 1130522928399324306, 215337700647490344, 38000770702498296, 6186171974825304, 924370524973896, 126050526132804, 15579278510796, 1731030945644, 171200862756, 14887031544, 1120529256, 71523144, 3764376, 156849, 4851, 99, 1]

**200th**

[1, 199, 19701, 1293699, 63391251, 2472258789, 79936367511, 2203959847089, 52895036330136, 1122550215450664, 21328454093562616, 366461620334848584, 5741232051912627816, 82585414900589338584, 1097206226536401212616, 13532210127282281622264, 155620416463746238656036, 1675208012521503627885564, 16938214348828536681954036, 161358778796735007338614764, 1452229009170615066047532876, 12378523459120956991548018324, 100153507987433197477070330076, 770746561468507650149628192324, 5652141450769056101097273410376, 39564990155383392707680913872632, 264781087962950397351403038993768, 1696560304355200694140471323923032, 10421727583896232835434323846955768, 61452255753319166029629978545842632, 348229449268808607501236545093108248, 1898412158917053376377708907120493352, 9966663834314530225982971762382590098, 50437359403955349931489584373269471102, 246252990031076120253743264881256829498, 1160906953003644566910503963011639339062, 5288576119238825249258962498164134766838, 23298321822592662584573267221641999107962, 99324424612105561544759718155421154091838, 410031599039717830992469605718533482276562, 1640126396158871323969878422874133929106248, 6360490170469769280761235835048470603119352, 23927558260338655865720839569944246554591848, 87363410392399278393445856104215039745835352, 309743000482142896122217126187671504553416248, 1066892557216269975532081212424201849017322632, 3571770735028382091998706667681023581492775768, 11627253669347711916506428088408438467412653032, 36819636619601087735603688946626721813473401268, 113464594480811515266860347570217040690499665132, 340393783442434545800581042710651122071498995396, 994483798684759751456599516938961121346144123804, 2830453888564316215684167855903197037677487121596, 7850504181489707239727786317316414425256426544804, 21225437231435134388893644487559194557174782880396, 55957970882874445207083244558110603832551700321044, 143891925127391430532499771720855838426561515111256, 360992022688017097651709953615480436754356081770344, 883808055546524618388669196782727965846871786403256, 2112151454780677477844107741463807511600151218353544, 4928353394488247448302918063415550860400352842824936, 11230182325145350742854190341225599501568017133650264, 24996212272097716169578681727244076309941715555544136, 54356842559958525638607609470356165943841508430310264, 115508290439911866982041170124506852630663205414409311, 239901833990586185270393199489360386232915888168388569, 487073420526341648882313465629913511442586803250970731, 966877088507514019423099864608634283908418579587747869, 1876879054161644861233076207769701845233989007435039981, 3563350088335876475674391061127984662690616811217249819, 6617650164052342026252440542094828659282574077974892521, 12023617903700734104036124365214547845738761352940297679, 21375320717690193962730887760381418392424464627449418096, 37187201796529515524203051309156714189560369968302412304, 63318749004901607514183573850726297133575765081163566896, 105531248341502679190305956417877161889292941801939278160, 172182563083504371310499192050220632556214799782111453840, 275044873497026463262225982106196594862524939911684530160, 430198391879964468179379100217384417605487726528532213840, 658911460980705071515251533244348285193215378606992378160, 988367191471057607272877299866522427789823067910488567240, 1452045626975998213153980230668100850703567223226520240760, 2089529072965460843319142283156535370524645516350358395240, 2945480741409143598413730688304995642787753318228818460760, 4067568642898341159714199521944993982897373629935035017240, 5503181105097755686672152294396168329802329028735635611560, 7294914488152838933495643739083292902296110572975144880440, 9475003875416905741206985546165656298384603387887257143560, 12059095841439698216081617967847198925216767948220145455440, 15039995937076477550393928027315045850551249912948720736560, 18382217256426805894925912033385056039562638782492880900240, 22018260230225514753262905622406275915520083816392571627760, 25847522878960386884265150078476932596480098393156497128240, 29738547828481305339960979122548728901326564817932744007760, 33534958189564025170594295606278353867453360326605009200240, 37064953788465501504341063564833970064027398255721325958160, 40153699937504293296369485528570134236029681443698103121340, 42637433954257136180681000097347668312485125656710356922660, 44377737380961509086014918468667981304831457316167922511340, 45274257328051640582702088538742081937252294837706668420660, 45274257328051640582702088538742081937252294837706668420660, 44377737380961509086014918468667981304831457316167922511340, 42637433954257136180681000097347668312485125656710356922660, 40153699937504293296369485528570134236029681443698103121340, 37064953788465501504341063564833970064027398255721325958160, 33534958189564025170594295606278353867453360326605009200240, 29738547828481305339960979122548728901326564817932744007760, 25847522878960386884265150078476932596480098393156497128240, 22018260230225514753262905622406275915520083816392571627760, 18382217256426805894925912033385056039562638782492880900240, 15039995937076477550393928027315045850551249912948720736560, 12059095841439698216081617967847198925216767948220145455440, 9475003875416905741206985546165656298384603387887257143560, 7294914488152838933495643739083292902296110572975144880440, 5503181105097755686672152294396168329802329028735635611560, 4067568642898341159714199521944993982897373629935035017240, 2945480741409143598413730688304995642787753318228818460760, 2089529072965460843319142283156535370524645516350358395240, 1452045626975998213153980230668100850703567223226520240760, 988367191471057607272877299866522427789823067910488567240, 658911460980705071515251533244348285193215378606992378160, 430198391879964468179379100217384417605487726528532213840, 275044873497026463262225982106196594862524939911684530160, 172182563083504371310499192050220632556214799782111453840, 105531248341502679190305956417877161889292941801939278160, 63318749004901607514183573850726297133575765081163566896, 37187201796529515524203051309156714189560369968302412304, 21375320717690193962730887760381418392424464627449418096, 12023617903700734104036124365214547845738761352940297679, 6617650164052342026252440542094828659282574077974892521, 3563350088335876475674391061127984662690616811217249819, 1876879054161644861233076207769701845233989007435039981, 966877088507514019423099864608634283908418579587747869, 487073420526341648882313465629913511442586803250970731, 239901833990586185270393199489360386232915888168388569, 115508290439911866982041170124506852630663205414409311, 54356842559958525638607609470356165943841508430310264, 24996212272097716169578681727244076309941715555544136, 11230182325145350742854190341225599501568017133650264, 4928353394488247448302918063415550860400352842824936, 2112151454780677477844107741463807511600151218353544, 883808055546524618388669196782727965846871786403256, 360992022688017097651709953615480436754356081770344, 143891925127391430532499771720855838426561515111256, 55957970882874445207083244558110603832551700321044, 21225437231435134388893644487559194557174782880396, 7850504181489707239727786317316414425256426544804, 2830453888564316215684167855903197037677487121596, 994483798684759751456599516938961121346144123804, 340393783442434545800581042710651122071498995396, 113464594480811515266860347570217040690499665132, 36819636619601087735603688946626721813473401268, 11627253669347711916506428088408438467412653032, 3571770735028382091998706667681023581492775768, 1066892557216269975532081212424201849017322632, 309743000482142896122217126187671504553416248, 87363410392399278393445856104215039745835352, 23927558260338655865720839569944246554591848, 6360490170469769280761235835048470603119352, 1640126396158871323969878422874133929106248, 410031599039717830992469605718533482276562, 99324424612105561544759718155421154091838, 23298321822592662584573267221641999107962, 5288576119238825249258962498164134766838, 1160906953003644566910503963011639339062, 246252990031076120253743264881256829498, 50437359403955349931489584373269471102, 9966663834314530225982971762382590098, 1898412158917053376377708907120493352, 348229449268808607501236545093108248, 61452255753319166029629978545842632, 10421727583896232835434323846955768, 1696560304355200694140471323923032, 264781087962950397351403038993768, 39564990155383392707680913872632, 5652141450769056101097273410376, 770746561468507650149628192324, 100153507987433197477070330076, 12378523459120956991548018324, 1452229009170615066047532876, 161358778796735007338614764, 16938214348828536681954036, 1675208012521503627885564, 155620416463746238656036, 13532210127282281622264, 1097206226536401212616, 82585414900589338584, 5741232051912627816, 366461620334848584, 21328454093562616, 1122550215450664, 52895036330136, 2203959847089, 79936367511, 2472258789, 63391251, 1293699, 19701, 199, 1]

PS1\_4:

We can see for a number x, the fastest way to decline always lies on being divided by two. So we come to the approach that for any number, we divide it by 2 if even, otherwise we minus 1. By continuing this loop, we finally get to 1, when the steps we take should be the least move.



PS1\_5:

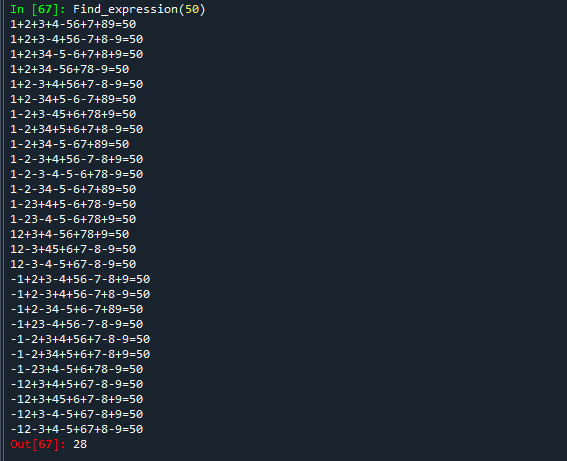
First there exists three types of possibilities between numbers:’+’ ‘-’ ‘ ‘

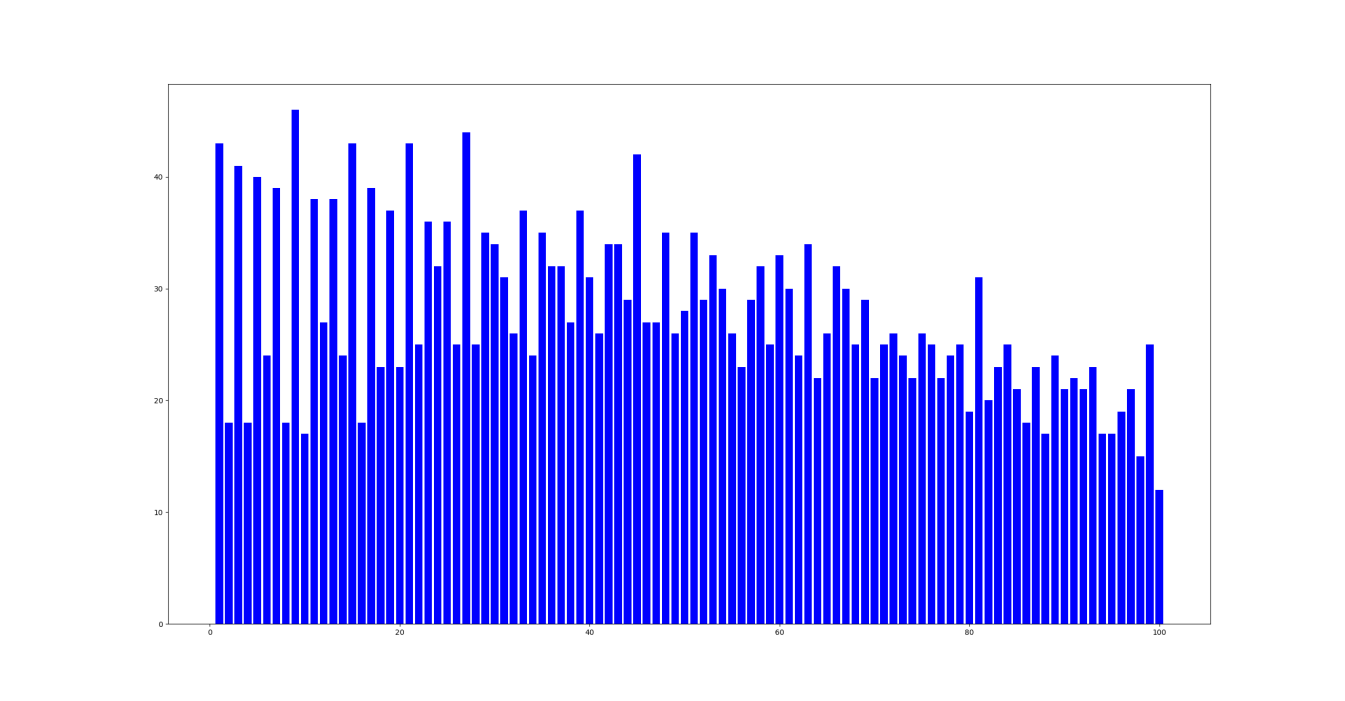
So naturally it reminds me of ternary number. With 0 stands for’+’ 1 stands for ‘-’and 2 stands for ‘ ‘

Basically, there are nines places to insert,so the overall possibilities goes to 2\*3\*\*8(we note that for the first place,’ ‘ is the same as’+’),and we can go through all situations by using a for loop.

For i as a decimal, we convert it to ternary integer using ‘ten2three’ function, and calculate the result. If the result is equal to the number we input, then output the result the specific i goes to.

As for the list Total\_solutions, we merely found a list of 100, and run the loop to find whether the result falls on 1-100,and take down the overall count.



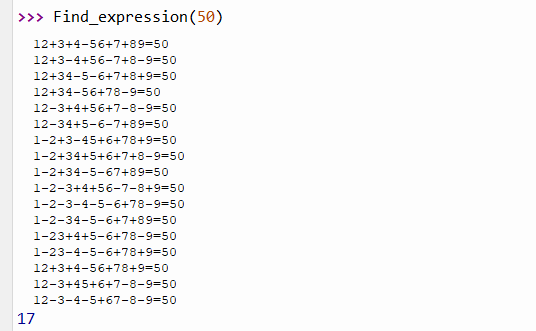


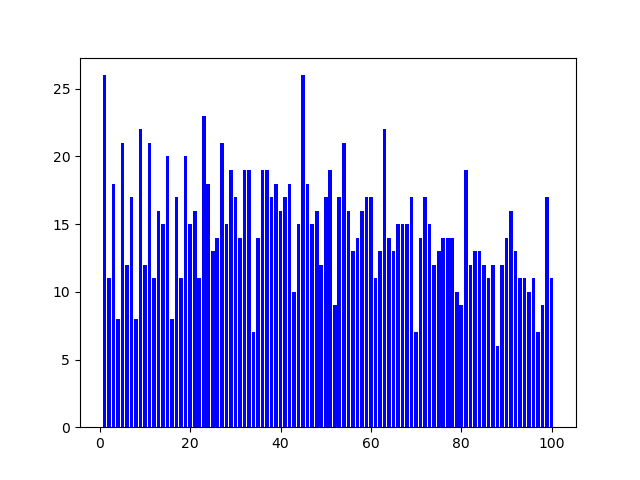
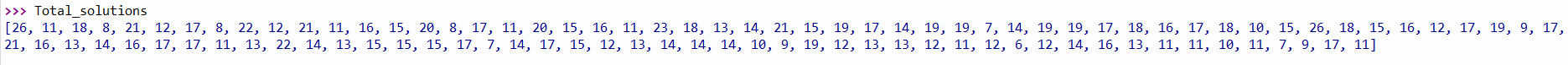
The max number falls on Find\_expression(9), with a value of 46

The min number falls on Find\_expression(100), with a value of 12

PS1\_5M:

Same as the previous one but I found that ‘between’ maybe means that there is supposed to be no other choice in front of 1. I wonder if I get it wrong, so I add a modified version named PS1\_5M in which there’s nothing added before 1. And the result seems to be as below.(Time limited, I modify it on Thonny so the UI is different)





The max number falls on Find\_expression(1&45), with a value of 26

The min number falls on Find\_expression(88), with a value of 6