4. Digits in a large number

The five adjacent digits in the 1000-digit number that have the greatest product are 9 × 9 × 8 × 7 x 9 = 40824.

73167176531330624919225119674426574742355349194934

96983520312774506326239578318016984801869478851843

85861560789112949495459521737958331952853208805511

12540698747158523863050715693290963295227443043597

66896648950445244523161731856403098711121722383113

62229893423380308135336276614282806444486645238749

30358907296290491560440772390713810515859307960866

70172427121883998797908792274921901699720888093776

65727333001053367881220235421809751254540594752243

52584907711670556013604839586446706324415722155397

53697817977846174064955149290862569321978468622482

83972241375657056057490261407972968652414535100474

82166370484403199990008895243450658541227588666881

16427171479924442928230863465674813919123162824586

17866458359124566529476545682848912883142607690042

24219022671055626321111109370544217506941658960408

07198403850962455444362981230987879927244284909188

84580156166097919133875499200524063689912560717606

05886116467109405077541002256983155200055935729725

71636269561882670428252483600823257530420752963450

Write a program in your favourite programming language to find the thirteen adjacent digits in the 1000-digit number that have the greatest product. What is the value of this product?

Ans:

n=7316717653133062491922511967442657474235534919493496983520312774506326239578318016984801869478851843858615607891129494954595017379583319528532088055111254069874715852386305071569329096329522744304355766896648950445244523161731856403098711121722383113622298934233803081353362766142828064444866452387493035890729629049156044077239071381051585930796086670172427121883998797908792274921901699720888093776657273330010533678812202354218097512545405947522435258490771167055601360483958644670632441572215539753697817977846174064955149290862569321978468622482839722413756570560574902614079729686524145351004748216637048440319989000889524345065854122758866688116427171479924442928230863465674813919123162824586178664583591245665294765456828489128831426076900422421902267105562632111110937054421750694165896040807198403850962455444362981230987879927244284909188845801561660979191338754992005240636899125607176060588611646710940507754100225698315520005593572972571636269561882670428252483600823257530420752963450

while n>0:

count = 0

maximp = []

pro = 1

while count < 13:

num = n%10

pro \*= num

count += 1

maximp += [pro]

n = n//10

print(maximp)

Output:

[96889010407]

5. Powered Sequence

The series, 11 + 22 + 33 + ... + 1010 = 10405071317.

Write a program in your favourite programming language to find the last ten digits of the series, 11 + 22 + 33 + ... + 10001000.

Ans:

def self\_power(n):

total = 0

for i in range(1, n + 1):

total += i \*\* i

return str(total)[-10:]

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

print(self\_power(1000))

Output:

9110846700

6. Multiple permutations

It can be seen that the number, 125874, and its double, 251748, contain exactly the same digits, but in a different order.

Write a program in your favourite programming language to find the smallest positive integer, x, such that 2x, 3x, 4x, 5x, and 6x, contain the same digits.

Ans:

public class factorial {

public static void main(String[] args) {

boolean run=true;

long count=1;

while(run){

boolean passed=true;

for (int i = 2; i < 7; i++) {

String mul=(count\*i)+"";

if(mul.length()==(""+count).length()){

passed = check(count+"",mul);

if(!passed){

break;

}

}

Else{

break;

}

if(i==6 && passed){

System.out.println(count);

System.exit(1);

}

}

count++;

}

}

public static boolean check(String ori,String mul){

char dig[]=ori.toCharArray();

for (int i = 0; i < dig.length; i++) {

if(!mul.contains(dig[i]+"")){

return false;

}

}

return true;

}

}

Output:

142857