МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

TASK

|  |
| --- |
| 9. Найти первую пару чисел n и k (n + k имеет минимальное значение), при которых превысит значение 10m (n, k, m – задаются как входные данные). |

CODE

|  |
| --- |
| import numpy as np  def A(n, k):  result = 1  for i in range(k):  result \*= (n - i)  return result  permutations = []  ns = []  ks = []  sums = []  max\_nk = 12  for n in range(max\_nk):  for k in range(n):  permutations.append(A(n, k))  ns.append(n)  ks.append(k)  sums.append(n+k)  idxs = np.argsort(sums)  permutations = np.array(permutations).astype(int)[idxs]  ns = np.array(ns).astype(int)[idxs]  ks = np.array(ks).astype(int)[idxs]  sums = np.array(sums).astype(int)[idxs]  for m in range(8):  i = 0  while permutations[i] < 10 \*\* m:  if (i >= len(permutations) - 1):  answer = None  break  answer = (permutations[i+1], ns[i+1], ks[i+1], sums[i+1])  i += 1  print(f'M: {m}, permutations: {answer[0]}, n: {answer[1]}, k: {answer[2]}, n+k: {answer[3]}') |

OUTPUT

|  |
| --- |
| M: 0, permutations : 1814400, n : 10, k : 8, n + k : 18  M : 1, permutations : 12, n : 4, k : 2, n + k : 6  M : 2, permutations : 120, n : 6, k : 3, n + k : 9  M : 3, permutations : 1680, n : 8, k : 4, n + k : 12  M : 4, permutations : 20160, n : 8, k : 6, n + k : 14  M : 5, permutations : 181440, n : 9, k : 7, n + k : 16  M : 6, permutations : 1814400, n : 10, k : 8, n + k : 18  M : 7, permutations : 19958400, n : 11, k : 9, n + k : 20 |

TASK

|  |
| --- |
| 9. Сколько существует перестановок из n элементов, в которых только хотя бы два элемента находятся на своих местах. Перестановки вывести. |

CODE

|  |
| --- |
| import numpy as np  from itertools import permutations  n = 5  lst = np.arange(n).tolist()  perm\_lst = lst[2:]  for perm in permutations(perm\_lst) :  print(lst[:2] + [\_ for \_ in perm]) |

OUTPUT

|  |
| --- |
| [0, 1, 2, 3, 4]  [0, 1, 2, 4, 3]  [0, 1, 3, 2, 4]  [0, 1, 3, 4, 2]  [0, 1, 4, 2, 3]  [0, 1, 4, 3, 2] |