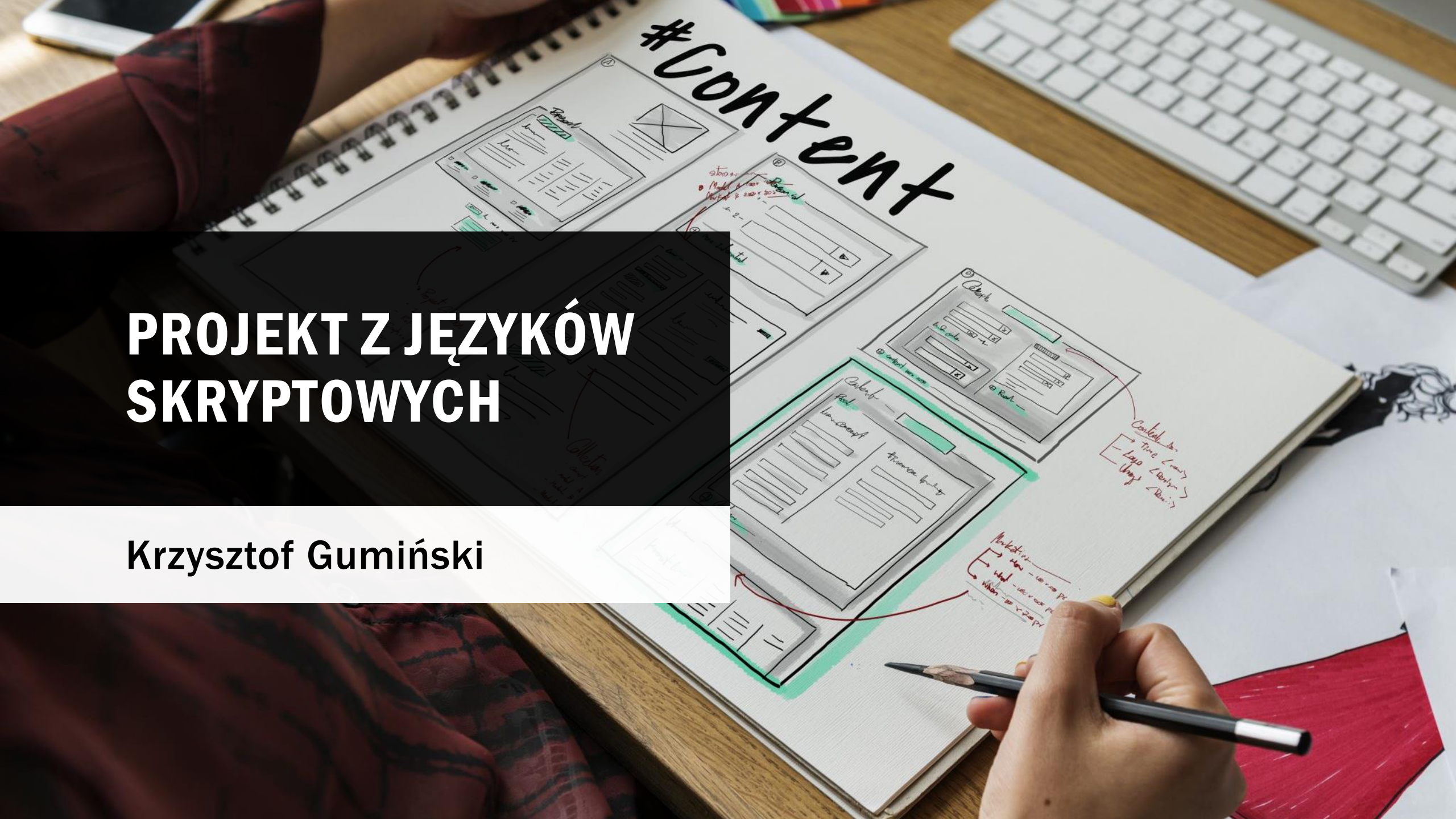

PROJEKT Z JĘZYKÓW SKRYPTOWYCH

Krzysztof Gumiński

TREŚĆ ZADANIA

W wielu przypadkach (np. w rachunku prawdopodobieństwa) spotykamy się z potrzebą obliczania liczb np. typu $C^{n}_{k} = n!/(k!(n-k)!)$. Możemy tu napotkać następujący problem: chociaż końcowy wynik jest stosunkowo mały, to liczby występujące w liczniku i mianowniku mogą być ogromne (np. C^{100}_{4} = 3921225, ale liczba 100! ma aż 158 cyfr). O ile w przykładzie tym możemy program „nauczyć" jak obliczać tego typu liczby bez wykorzystywania dodatkowych bibliotek dla dużych liczb (w tym przykładzie mamy: C^{100}_{4} = 100!/(4!96!) = 96!97·98·99·100/(2·3·4·96!) = 97 · 49 · 33 · 25 = 3921225), to nie zawsze da się to tak łatwo zrobić, np. wyrażenie 13·55·2^13·3^5·7^2·100!/(83·89·97·49!·67!) po uproszczeniu sprowadza się do liczby 409457, ale w tym przypadku „nauczenie" programu obliczania wartości takich wyrażeń (w poprzedni sposób) może być trudne lub niewykonalne. Załóżmy, że w wyrażeniach będą występowały tylko liczby naturalne nie większe od 100 (tak jak w powyższym przykładzie – nie większe od 100 jako „składowe" wyrażenia).

<u>Zadanie do zrealizowania</u>:

Zaproponuj, opisz i zaimplementuj metodę obliczania wartości ww. wyrażeń (bez korzystania z bibliotek dla dużych liczb).

INSTRUKCJA OBSŁUGI

W celu uruchomienia programu należy wykorzystać plik wykonywalny run.exe, który wyświetli początkowy ekran (w postaci menu). Można tam zobaczyć instrukcję obsługi, która pokazuje potencjalne działania. Aby przejść dalej użytkownik musi wybrać liczbę od 1 do 3 (wybrana liczba odpowiada opcji, jaka zostanie wykonana).

```
Zadanie z Algorytmiona - Krzysztof Guminski
.
1. Uruchom program
2. Wyswietl tresc zadania
3. Wyjscie
.
Wybierz jedna z powyzszych opcji (1-3)
```

```
Zadanie z Algorytmiona - Krzysztof Guminski
.
1. Uruchom program
2. Wyswietl tresc zadania
3. Wyjscie
.
Wybierz jedna z powyzszych opcji (1-3)1
-dane1.txt
-dane2.txt
-dane3.txt
10:31:13 30/12/2023
```

```
Zadanie z Algorytmiona - Krzysztof Guminski
.
1. Uruchom program
2. Wyswietl tresc zadania
3. Wyjscie
.
Wybierz jedna z powyzszych opcji (1-3)2
W wielu przypadkach (np. w rachunku prawdopodobienstwa) spotykamy sie z potrzeba
obliczania liczb np. typu Cn,k = n!/(k!(n-k)!).
Mozemy tu napotkac nastepujacy problem: chociaz koncowy wynik jest stosunkowo
maly, to liczby wystepujace w liczniku i mianowniku moga byc ogromne (np. C4,100 = 3921225, ale liczba 100! ma az 158 cy
fr!)
O ile w przykladzie tym mozemy program "nauczyc" jak obliczac tego typu liczby bez
wykorzystywania dodatkowych bibliotek dla duzych liczb (w tym przykladzie mamy: C4,100 = 97*49*33*25), to nie zawsze da
sie to tak latwo
zrobic, np. wyrazenie 13*55*2**13*3**5*7**2*100!/(83*89*97*49!*67!)
       po uproszczeniu sprowadza sie do liczby 409457, ale
w tym przypadku "nauczenie" programu obliczania wartosci takich wyrazen (jak w poprzedni sposob) moze byc trudne lub nie
wykonywalne
Zalozmy, ze w wyrazeniach beda wystepowaly tylko liczby naturalne nie wieksze od 100
(tak jak w powyzszym przykladzie - nie wieksze od 100 jako "skladowe" wyrazenia).
Zaproponuj, opisz i zaimplementuj metode obliczania wartosci takich wyrazen (bez
korzystania z bibliotek dla duzych liczb).
```

```
Zadanie z Algorytmiona - Krzysztof Guminski
.
1. Uruchom program
2. Wyswietl tresc zadania
3. Wyjscie
.
Wybierz jedna z powyzszych opcji (1-3)3

D:\Informatyka\Projekt z jezykow skryptowch>
```

OPIS DZIAŁANIA

W celu uproszczenia działania najpierw zajmujemy się skracaniem ze sobą silni (jest to nasza pierwsza czynność, ponieważ silnia jest funkcją rosnącą geometrycznie). W tym celu gdy mamy iloraz dwóch silni, zapisujemy większy składnik w postaci silnia mniejszego składnika razy pozostałe czynniki (np. 100!/90! = (90!*91*92…*100)/90!), co spowoduje, że duża część silni nam się uprości. Następnie doprowadzamy równanie do postaci, której elementy są liczbami całkowitymi (wykonujemy potęgowania oraz pozostałe silnie rozpisujemy). Na sam koniec skracamy ze sobą składniki, wykorzystując algorytm NWD (największy wspólny dzielnik). Przechodzimy po kolei przez kolejne elementy licznika i skracamy z elementem mianownika, o ile NWD rozważanych liczb jest większe od 1. Gdy już przeszliśmy po wszystkich elementach skracamy wszystkie elementy występujące i w liczniku i w mianowniku a następnie mnożymy przez siebie elementy licznika, następnie robimy to samo dla mianownika a na sam koniec dzielimy otrzymany licznik przez otrzymany mianownik.

```python
import sys
import os


def czyPierwsza(liczba):
    for x in range(2,liczba):
        if (liczba % x) == 0:
            return False
    return True
def maks(lista):
    max = lista[0]
    for x in lista:
        if x > max:
            max = x
    return max
def potegaNaLiczbe(liczba):
    liczby = liczba.split("^")
    a = int(liczby[0])
    b = int(liczby[1])
    return a**b

def silniaPrzezSilnie(silnieMianownika, silnielicznika, mianownik,
licznik):
    gorneSilnie = []
    dolneSilnie = []
    for x in silnielicznika:
        gorneSilnie.append(int(x[:-1]))
    for x in silnieMianownika:
        dolneSilnie.append(int(x[:-1]))
    a = maks(gorneSilnie)
    b = maks(dolneSilnie)
    silnielicznika.remove(str(a) + "!")
    silnieMianownika.remove(str(b) + "!")
    if a >= b:
        for y in range(b+1,a+1):
            licznik.append(y)
    else:
        for y in range(a+1,b+1):
            mianownik.append(y)

def silnia(n):
    if n == 1:
        return 1
    else:
        return n*silnia(n-1)


def nwd(a,b):
    while a != 0 and b != 0:
        if a >= b:
            a = a % b
        else:
            b = b % a
    if a == 0:
        return b
    else:
        return a


os.chdir("input")
gora = []
dol = []
with open(sys.argv[1], "r") as file:
    gora = file.readline().rstrip().split("*")
    dol = file.readline().rstrip().split("*")
licznik = []
mianownik = []
potegiLicznika = []
potegiMianownika = []
silnieLicznika = []
silnieMianownika = []
a = 0
b = 0
for x in gora:
    if '^' in x and '!' in x:
        potegowanie = x.split("^")
        if "!" in potegowanie[0]:
            a = silnia(int(potegowanie[0][:-1]))
        else:
            a = int(potegowanie[0])
        if "!" in potegowanie[1]:
            b = silnia(int(potegowanie[1][:-1]))
        else:
            b = int(potegowanie[1])
        potegiLicznika.append(str(a) + "^" + str(b))
    elif '^' in x:
        potegiLicznika.append(x)
    elif '!' in x:
        silnieLicznika.append(x)
    else:
        licznik.append(int(x))
for x in dol:
    if '^' in x and '!' in x:
        potegowanie = x.split("^")
        if "!" in potegowanie[0]:
            a = silnia(int(potegowanie[0][:-1]))
        else:
            a = int(potegowanie[0])
        if "!" in potegowanie[1]:
            b = silnia(int(potegowanie[1][:-1]))
        else:
            b = int(potegowanie[1])
        potegiMianownika.append(str(a) + "^" + str(b))
    elif '^' in x:
        potegiMianownika.append(x)
    elif '!' in x:
        silnieMianownika.append(x)
    else:
        mianownik.append(int(x))
while len(silnieLicznika) > 0 and len(silnieMianownika) > 0:
    silniaPrzezSilnie(silnieMianownika,silnieLicznika,mianownik,licznik)
for x in silnieLicznika:
    a = int(x[:-1])
    for i in range(2,a+1):
        licznik.append(i)
silnieLicznika.clear()
for x in silnieMianownika:
    a = int(x[:-1])
    for i in range(2,a+1):
        mianownik.append(i)
silnieMianownika.clear()
for x in potegiLicznika:
    a = potegaNaLiczbe(x)
    licznik.append(a)
for x in potegiMianownika:
    a = potegaNaLiczbe(x)
    mianownik.append(a)
liczbyPierwsze = []
for x in range(2,101):
    if czyPierwsza(x):
        liczbyPierwsze.append(x)
dzielnik = 0
for x in range(len(licznik)):
    if licznik[x] not in liczbyPierwsze:
        for y in range(len(mianownik)):
            if mianownik[y] not in liczbyPierwsze:
                while nwd(licznik[x],mianownik[y]) > 1:
                    dzielnik = nwd(licznik[x], mianownik[y])
                    licznik[x] = licznik[x]//dzielnik
                    mianownik[y] = mianownik[y]//dzielnik
kopiaLicznika = licznik
kopiaMianownika = mianownik
licznik = [x for x in kopiaLicznika if x not in kopiaMianownika]
mianownik = [y for y in kopiaMianownika if y not in kopiaLicznika]
wartoscLicznika = 1
wartoscMianownika = 1
for x in licznik:
    wartoscLicznika = wartoscLicznika*x
for x in mianownik:
    wartoscMianownika = wartoscMianownika*x
wynik = wartoscLicznika//wartoscMianownika
os.chdir("..")
os.chdir("output")
with open(sys.argv[1],"a+") as file2:
    file2.write(str(wynik))
```

PEŁNY KOD APLIKACJI-DUZELICZBY.PY

PEŁNY KOD APLIKACJI-RAPORT.PY

```
import datetime
import os
from os.path import isfile, join

now = datetime.datetime.now()
data = now.strftime("%H:%M:%S %d/%m/%Y")
with open("raport.html", "w") as file1:
    file1.write(f"""
    <html>
        <head>
            <title> Raport z obliczania </title>
        </head>
        <body>
        <h1>{data}</h1>
        <table>
            <tr>
                <th>input</th>
                <th>output</th>
            <tr>
    """)
    a = ""
    b = ""
    wynik = ""
```

```
    filein = [file for file in os.listdir("input") if
isfile(join("input", file))]
    for x in range(len(filein)):
        with open(f"input/dane{x+1}.txt", "r") as file2:
            a = file2.readline().rstrip()
            b = file2.readline().rstrip()
        file1.write(""" <tr><td>""" + "(" + a + ")/(" + b + ")")
        file1.write("""</td>
            <td>""")
        with open(f"output/dane{x+1}.txt", "r") as file3:
            wynik = file3.readline().rstrip()
        file1.write(wynik)
        file1.write("""</td>
            </tr>""")
    file1.write("""
        </table>
        </body>
    </html>""")
```

PEŁNY KOD APLIKACJI-PROJEKT.BAT

```
@echo off

:menu
echo Zadanie z Algorytmiona - Krzysztof Guminski
echo .
echo 1. Uruchom program
echo 2. Wyswietl tresc zadania
echo 3. Wyjscie
echo .
set /p wybor=Wybierz jedna z powyzszych opcji (1-3)

if %wybor% == 1 goto opcja1
if %wybor% == 2 goto opcja2
if %wybor% == 3 goto exit
echo Musisz wybrac liczbe od 1 do 3
goto menu

:opcja1
IF EXIST raport.html DEL raport.html
IF NOT EXIST output mkdir output
echo "<HTML>" >> raport.html
DEL /Q output
for /f "delims=" %%a in ('dir /b input') do (
    echo -%%a
     call python duzeLiczby.py %%a "wynik"
)
call python raport.py
goto menu
```

```
:opcja2
echo W wielu przypadkach (np. w rachunku prawdopodobienstwa) spotykamy sie z
potrzeba
echo    obliczania liczb np. typu Cn,k = n!/(k!(n-k)!).
echo    Mozemy tu napotkac nastepujacy problem: chociaz koncowy wynik jest
stosunkowo
echo    maly, to liczby wystepujace w liczniku i mianowniku moga byc ogromne
(np. C4,100 = 3921225, ale liczba 100! ma az 158 cyfr!)
echo    O ile w przykladzie tym mozemy program "nauczyc" jak obliczac tego
typu liczby bez
echo    wykorzystywania dodatkowych bibliotek dla duzych liczb (w tym
przykladzie mamy: C4,100 = 97*49*33*25), to nie zawsze da sie to tak latwo
echo    zrobic, np. wyrazenie 13*55*2**13*3**5*7**2*100!/(83*89*97*49!*67!)
echo    po uproszczeniu sprowadza sie do liczby 409457, ale
echo    w tym przypadku "nauczenie" programu obliczania wartosci takich
wyrazen (jak w poprzedni sposob) moze byc trudne lub niewykonywalne
echo    Zalozmy, ze w wyrazeniach beda wystepowaly tylko liczby naturalne nie
wieksze od 100
echo    (tak jak w powyzszym przykladzie - nie wieksze od 100 jako "skladowe"
wyrazenia).
echo    Zaproponuj, opisz i zaimplementuj metode obliczania wartosci takich
wyrazen (bez
echo    korzystania z bibliotek dla duzych liczb).
goto menu
:exit
```

PEŁNY KOD APLIKACJI-RUN.PY

```
import subprocess

path = r"D:\Informatyka\Projekt z jezykow
skryptowych\projekt.bat"

subprocess.call(path, True)
```

TESTY

W celu przetestowania programu będziemy wprowadzać różne dane wyjściowe i na podstawie widoku z raport.html oraz kalkulatora, będzie można sprawdzić poprawność otrzymanych wyników.

12:58:02 30/12/2023

input	output
(150!)/(100!*50!)	20128660909731932294240234380929315748140
(13*55*2^13*3^5*7^2*100!)/(83*89*97*49!*67!)	409457
(140!)/(20!*120!)	827163809330939321148600

13:45:24 30/12/2023

input	output
(100!)/(96!*4!)	3921225
(13*55*2^13*3^5*7^2*100!*5!)/(83*89*97*49!*67!*4!)	2047285
(10!)/(9!*1!)	10

13:50:14 30/12/2023

input	output
(100!)/(90!)	62815650955529472000
(10!*2!*13*55*2^13*3^5*7^2*100!*5!)/(9!*83*89*97*49!*67!*4!)	40945700
(16!)/(12!*4!)	1820

DZIĘKUJE ZA UWAGĘ