

# Documentație PROIECT - EGIPT ANTIC

(10LF222) Geană Andrei-Niculae-Marian, (10LF222) Ene Călin Tudor

May 2023

## Cuprins

<b>1</b>	<b>Prefață</b>	<b>3</b>
1.1	Sistemul de calcul antic egiptean - regăsit în arhitectura sistemelor de calcul moderne . . . . .	3
1.2	Noțiuni privind sistemul de calcul antic egiptean . . . . .	3
1.3	Propunere față de subiect . . . . .	4
1.4	Module folosite . . . . .	5
<b>2</b>	<b>Interfață</b>	<b>6</b>
2.1	Meniu Principal . . . . .	6
2.2	Reprezentarea grafică . . . . .	6
2.3	Produsul . . . . .	7
2.4	Împărțirea . . . . .	8
<b>3</b>	<b>Autoevaluare</b>	<b>9</b>
3.1	Rezultatul final . . . . .	9
3.2	Utilitate . . . . .	9
<b>4</b>	<b>References</b>	<b>10</b>

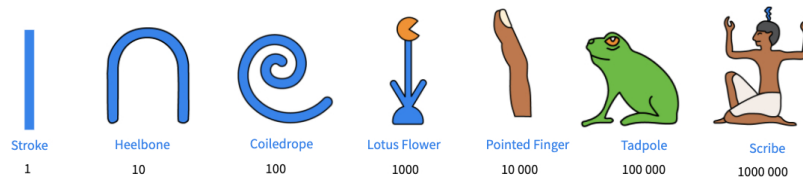
# 1 Prefață

## 1.1 Sistemul de calcul antic egiptean - regăsit în arhitectura sistemelor de calcul moderne

Legătura dintre arhitectura sistemelor de calcul moderne și sistemul de calcul antic egiptean constă în utilizarea principiilor matematice și dezvoltarea infrastructurii tehnologice pentru a efectua operații complexe și a procesa informații. Sistemul de calcul egiptean, bazat pe numerație folosind simboluri și utilizarea acestora pentru reprezentarea numerelor, a fost un pas semnificativ în evoluția matematicii și a calculului, această abordare influențând dezvoltarea arhitecturii sistemelor de calcul moderne.[1] Astfel, legătura între arhitectura sistemelor de calcul și sistemul de calcul egiptean constă în evoluția și aplicarea principiilor matematice și a tehnologiilor pentru a rezolva probleme complexe și a crea sisteme de calcul moderne eficiente.[2]

## 1.2 Noțiuni privind sistemul de calcul antic egiptean

Înainte de a putea prezenta proiectul, trebuie cunoscută modalitatea de numerație folosită în egiptul antic. În principal, au existat două modalități prin care numerele puteau fi reprezentate: inițial, acestea erau reprezentate folosind un set de 7 simboluri [3]:



În egiptul antic existau doar fracții cu numărător 1. Pentru a reprezenta virgula, era folosit un "ochi" albastru:



De exemplu, pentru imaginea de mai jos, numerele, de la stânga la dreapta, sunt: 2, 21, 12, 12, 23:



Se poate observa că în exemplul dat, simboluri sunt în ordine descrescătoare. Aceasta nu este o regulă fixată pentru reprezentarea numerelor, acestea putând fi și crescătoare sau puse într-o altă ordine, simbolurile fiind unice.

Evident, se poate realiza că acest sistem are dezavantajele sale: nu există caracter pentru cifra 0, există un număr finit de numere reprezentabile (0 nu poate fi reprezentat, 9999999.9999999 este maximul), inexistența numerelor negative, utilizarea unui număr considerabil de simboluri la reprezentarea unor numere relativ mici (cifra 9 necesită utilizarea a 9 simboluri).

Ulterior, un nou sistem, "heiratic", a fost introdus pentru scrierea pe papyrus, acesta folosind mai multe simboluri [4]:

1	𐎠	10	𐎡	100	𐎢	1000	𐎣
2	𐎠𐎠	20	𐎡𐎡	200	𐎢𐎢	2000	𐎣𐎣
3	𐎠𐎠𐎠	30	𐎡𐎡𐎡	300	𐎢𐎢𐎢	3000	𐎣𐎣𐎣
4	𐎠𐎠𐎠𐎠	40	𐎡𐎡𐎡𐎡	400	𐎢𐎢𐎢𐎢	4000	𐎣𐎣𐎣𐎣
5	𐎠𐎡	50	𐎡𐎢	500	𐎢𐎣	5000	𐎣𐎣𐎢
6	𐎠𐎢	60	𐎡𐎣	600	𐎢𐎣𐎢	6000	𐎣𐎣𐎣𐎢
7	𐎠𐎣	70	𐎡𐎣𐎢	700	𐎢𐎣𐎢𐎢	7000	𐎣𐎣𐎣𐎢𐎢
8	𐎠𐎣𐎢	80	𐎡𐎣𐎢𐎢	800	𐎢𐎣𐎢𐎢𐎢	8000	𐎣𐎣𐎣𐎢𐎢𐎢
9	𐎠𐎣𐎢𐎢	90	𐎡𐎣𐎢𐎢𐎢	900	𐎢𐎣𐎢𐎢𐎢𐎢	9000	𐎣𐎣𐎣𐎢𐎢𐎢𐎢

𐎣𐎣𐎢𐎢𐎢𐎢
2765

De exemplu, numărul 2765 este reprezentat astfel:

Pentru realizarea proiectului, au fost folosite simbolurile de la prima modalitate de reprezentare.

### 1.3 Propunere față de subiect

Dorim a realiza un meniu cu interfață grafică ce conține 3 sub-meniuri:

1. Reprezentarea grafică eficientizată a numerelor folosind sistemul de numerație egiptean.
2. Calcularea produsului a două numere folosind algoritmul folosit în acea perioadă.
3. Calcularea împărțirii a două numere folosind algoritmul folosit în acea perioadă.

## 1.4 Module folosite

La realizarea proiectului, au fost folosite următoarele module (necesită instalare):

1. tkinter: Modul folosit la realizarea meniului grafic. Acesta a fost folosit în special la realizarea butoanelor/entry field-urilor. <https://docs.python.org/3/library/tkinter.html>
2. customtkinter: Modul folosit la realizarea meniului grafic. Acesta a fost folosit la modificarea aparenței ferestrei meniului. <https://pypi.org/project/customtkinter/0.3/>
3. PIL: Modul folosit la includerea pozelor. Acesta poate realiza deschiderea și facilitează utilizarea imaginilor cu alte extensii (.png, .jpg, etc). <https://pypi.org/project/Pillow/>
4. subprocess: Modul folosit la inițializarea subproceselor meniului. Acesta a fost utilizat la utilizarea mai multor script-uri .py. <https://docs.python.org/3/library/subprocess.html>

## 2 Interfață

### 2.1 Meniu Principal

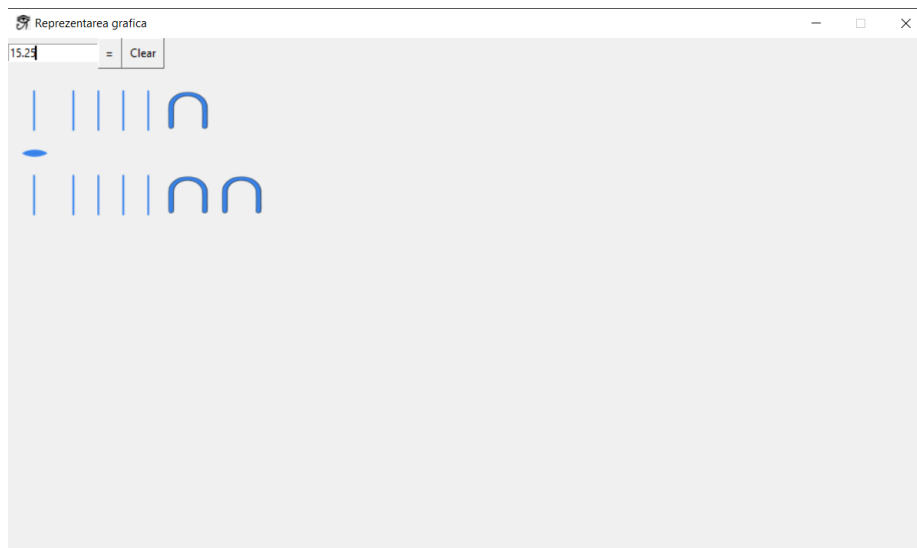
Pentru a lucra modular, am decis asupra unui stil de lucru împărțit pe mai multe fișiere .py, fiecare reprezentând o alegere din meniul:



Prin urmare, apăsând unul din cele trei butoane, programul va crea un nou proces, acesta conținând interfața asociată respectivei alegeri. Până când procesul nou creat nu este întrerupt, cel principal nu poate fi operat.

### 2.2 Reprezentarea grafică

Algoritmul se bazează pe descompunerea unui număr citit de la tastatură în sumă de puteri ale lui 10. De exemplu, pentru numărul 423 reprezentarea sa va fi  $3 \cdot 10^0 + 2 \cdot 10^1 + 4 \cdot 10^2$ , pentru fiecare putere existând un simbol diferit.

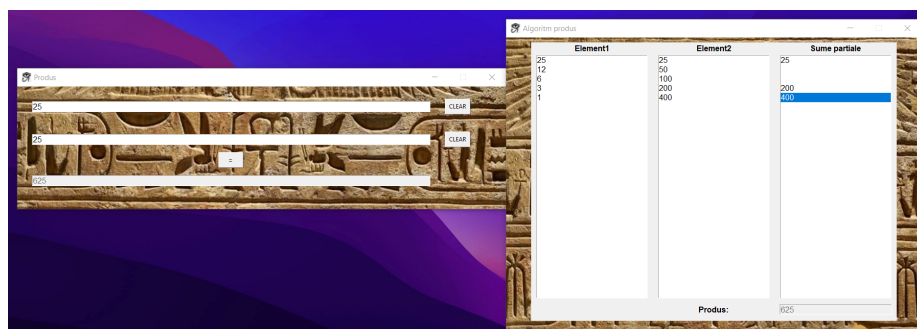


Reprezentarea este secționată astfel: prima parte, până la simbolul "ochiului", reprezintă partea întreagă a numărului, iar după acesta este reprezentată partea fracționară. Simbolurile utilizate sunt cele prezentate anterior.

## 2.3 Produsul

Această opțiunea este reprezentată de 2 ferestre: o interfață care conține cele două numere introduse și rezultatul, iar cealaltă reprezintă decurgerea algoritmului de înmulțire și rezultatul.

Algoritmul funcționează astfel: se identifică deînmulțitul și înmulțitorul. Se pun deînmulțitul și înmulțitorul pe două coloane. Pe fiecare rând, primul număr se împarte la 2 și se ia partea întreagă a acestui număr, iar al doilea se înmulțește cu 2. Atunci când deînmulțitul este impar, valoarea înmulțitorului de pe acel rând este adăugată la rezultatul final. Algoritmul se oprește când deînmulțitul devine 1. [5]

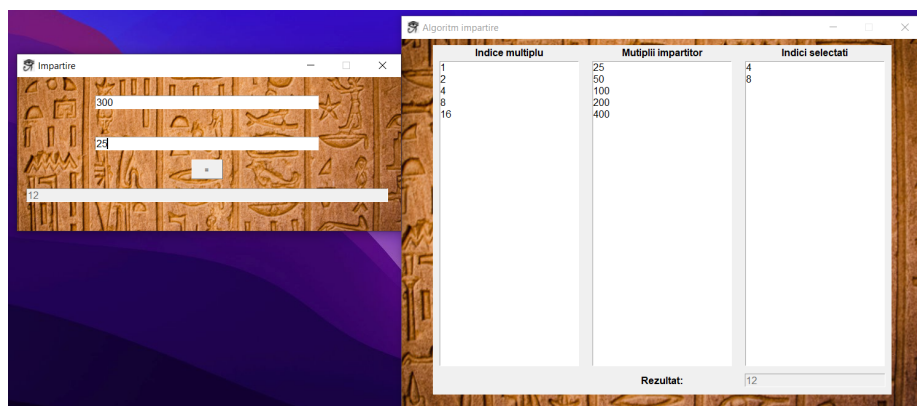


Rezultatul este reprezentat de partea întreagă a produsului, iar partea fracționară este reprezentată de o sumă de fracții a căror numărător este 1.

## 2.4 Împărțirea

Această opțiunea este reprezentată de 2 ferestre: o interfață care conține cele două numere introduse și rezultatul, iar cealaltă reprezintă decurgerea algoritmului de împărțire și rezultatul.

Algoritmul funcționează astfel: se utilizează două coloane. Pe prima se vor pune multiplii lui 2. Pe a doua se vor pune multiplii împărțitorului de rang 2 până când nu este depășită valoarea deîmpărțitului. Se va căuta o sumă a elementelor de pe coloana a doua astfel încât să fie egală cu deîmpărțitul. Dacă nu se găsește suma exactă, se va returna cea mai mare sumă mai mică decât deîmpărțitul. [6]



Rezultatul este secționat identic ca cel de la produs.



## 3 Autoevaluare

Am vrut să învățăm noutăți despre domeniul informatic, dar am vrut să îl putem îmbina și cu istoria pentru a ne ieși cât mai mult din zona de confort. Ne așteptam că va urma să îmbinăm învățatul limbajului cu implementarea practică a ideilor, iar acest gând putea părea inițial dificil și poate ușor descurajant pentru majoritatea persoanelor. Fiind iubitori de a accepta noi provocări, am fost încântați să realizăm acest proiect.

### 3.1 Rezultatul final

În ciuda faptului că aceasta este prima interacțiune a echipei noastre cu programarea în Python, suntem de părere că am realizat o implementare fidelă structurii propuse inițial.

Proiectul dispune de meniul principal și de cele 3 submeniuri: reprezentarea eficientă a numerelor, împărțirea și produsul numerelor folosind algoritmii utilizați în egiptul antic.

Reprezentarea numerelor a fost eficientizată prin simplificarea părții fracționare, deoarece, inițial, aceasta trebuia formată sub forma unei sume de fracții cu numărătorul 1. Această modalitate poate duce la o afisare foarte lungă a unui număr relativ mic, așa că am decis asupra unui model simplu: partea fracționară va fi reprezentată similar cu cea întreagă, dar va fi despărțită de aceasta prin simbolul de "ochi albastru", descris în prefața documentației.

### 3.2 Utilitate

Proiectul poate fi utilizat ca material suport pentru studiul sistemului de calcul egiptean, oferind o înțelegere aprofundată și rapidă datorită meniului grafic.

## 4 References

- [1] Nicholas Graham. Ancient egyptian math is identical to math used in modern computers.  
LINK.  
Accessed: 2023-05-26.
- [2] Jill Platts. Ancient egyptians: The original computer scientists?  
LINK.  
Accessed: 2023-05-26.
- [3] Eda Aydemir. Egyptian fractions.  
LINK.  
Accessed: 2023-05-26.
- [4] E F Robertson J J O'Connor. Egyptian numerals.  
LINK.  
Accessed: 2023-05-26.
- [5] Agnes Azzolino. Ancient egyptian multiplication, division, root extraction – computation.  
LINK.  
Accessed: 2023-05-26.
- [6] Pamini Thangarajah. Egyptian multiplication and division.  
LINK.  
Accessed: 2023-05-26.