

Name: grep.py	ID:1	Difficulty: B	Propose: GDT
<p>Creați un utilitar similar lui “grep” dar în limbajul python care sa suporte următoarele opțiuni ale utilitarului “grep” din Linux:</p> <ul style="list-style-type: none"> - Căutare după expresii regulate într-un fișier - Cautare recursiva într-un folder - Opțiune de tipul “NOT” ⇒ să verifice ca o anumită expresie regulată NU face match într-un fișier - Opțiune de tipul “COUNT” care sa spuna de cate ori se face sau match la o expresie regulată într-un fișier. - Opțiune de ignore case 			
<p>INPUT:</p> <ul style="list-style-type: none"> - grep.py “.*abc{3,5}.*” a.txt ⇒ <i>afișează toate liniile din a.txt care au un substring care respecta expresia regulată “.*abc{3,5}.*”</i> - grep.py “test” a.txt -ignoreCase ⇒ <i>afiseaza toate liniile din a.txt care conțin textul “test” fără sa țină cont de case</i> - grep.py “test” C:\MyFolder -ignoreCase -count ⇒ <i>afiseaza toate fișierele parcurgand recursiv “C:\MyFolder” care NU conțin textul “test” fără sa țină cont de case. Pentru fiecare fișier este precizat de cate ori s-a găsit textul “text” în acel fișier.</i> 			
<p>OUTPUT: Verificati ce afișează utilitarul grep din linux si incercati sa obtineti un output similar.</p>			

Name: Connectivity Checker	ID:2	Difficulty: B	Propose: CGI
<p>Creati un utilitar care sa verifice statusul conectivitatii . Se va realiza un script ce va putea primi la linia de comanda un URL (pagina web), FTP sau un URI (MongoDB , PostgreSQL, Elastic search) si care va face un request HTTP / va verifica daca se poate conecta la baza de date. Monitorizarea se va face fie one-time fie se va face la intervale timp x (1h, 1zi etc.) in functie de un parametru dat la linia de comanda.</p>			
<p>INPUT:</p> <ul style="list-style-type: none"> - check_connectivity.py -url <URL> - check_connectivity.py -mongodb <URI> - check_connectivity.py -postgresql <URI> 			
<p>OUTPUT: Orice formă de logging este OK (se poate loga si response time)</p>			

Name: CMD	ID:3	Difficulty: B	Propose: CGI
<p>Creati un utilitar asemanator “cmd-ului”, dar realizat in python care sa suporte urmatoare operatii:</p> <ul style="list-style-type: none"> - copiere/mutare/sterge fisiere si directoare 			

<ul style="list-style-type: none"> - listare/create/modificare/sterge chei registri - listare/kill procese running <p>Scriptul va realiza operatiile in functie de parametri primiti la linia de comanda si va putea afisa si mesaje de tipul "help" pentru fiecare functionalitate.</p> <p>SFAT: Folosirea unui VM pentru a testa functionalitatile de modificare a registrilor</p>
<p>INPUT:</p> <ul style="list-style-type: none"> - my_cmd.py -list_dir <dir> - my_cmd.py -move_dir <dir1> <dir2> - my_cmd.py -kill <PID> <p>Se pot folosi orice nume de parametri atat timp cat sunt sugestivi.</p>
<p>OUTPUT: Orice formă de logging este OK (operatia s-a realizat cu succes / a dat fail)</p>

Name: Currency Converter	ID:4	Difficulty: B	Propose: CGI
<p>Creati o aplicatie care va avea o interfata grafica si va avea functia de convertor valutar. Utilizatorul va putea selecta suma si moneda din care/ in care se va face conversia. Cursul folosit va fi unul, oficial updatat automat de pe bnr.ro (crawl pe BNR sau RSS)</p>			
<p>INPUT:</p> <ul style="list-style-type: none"> - Suma de bani, moneda intrare, moneda iesire 			
<p>OUTPUT: Conversia sumei dintr-o moneda in alta.</p>			

Name: Calculator	ID:5	Difficulty: B	Propose: MLA
<p>Sa se scrie un script care va primi de la linia de comanda o expresie matematica si returneaza rezultatul expresiei. Calculatorul va avea urmatoarele operatii: adunare, scadere, inmultire, impartire, putere, radical, logaritm, functii trigonometrice. Implementati un lexer pentru a rezolva problema. Nu aveti voie sa folositi exec.</p>			
<p>INPUT: $2 * (3 + 5 / 4) - (5 ^ 2 + 8) / \text{rad}(9)$</p>			
<p>OUTPUT: -2.5</p>			

Name: Generate structure	ID:6	Difficulty: B	Propose: TDY
<p>Creati un script care primeste de la linia de comanda un path catre un director si un fisier JSON. In fisierul JSON se afla un dictionar in care se afla o structura de directoare si fisiere astfel: fiecare cheie care are ca valoare un dictionar este un director iar dictionarul continutul, iar fiecare cheie care are ca valoare un string reprezinta un fisier iar string-ul respectiv este continutul fisierului. Scriptul va crea in folderul dat ca argument directoarele si fisierele</p>			

conform dictionarului din JSON.

INPUT: create_structure.py root_folder_path structure_json_file_path

Exemplu de dictionar:

```
{"dir1": {"dir2": {"file1": "continut1", "file2": "continut2"}, "file3": "continut3"}, "file4": "continut4"}
```

OUTPUT:

root_folder

---dir1

-----dir2

-----file1: continut1

-----file2: continut2

-----file3: continut3

---file4: continut4

Name: X si 0

ID:7

Difficulty: B

Propose: TDY

Creati un joc de X si 0. Utilizatorul primeste de fiecare data un string pe mai multe linii cu situatia actuala a table / exista o interfata grafica si este intrebat care patratel vrea sa fie completat / da el click pe patratel. La final programul va spune cine a castigat si va intreba daca mai joaca o data. Scorul total se va afisa la finalul fiecarui meci. Vor fi 3 nivele de dificultate:

1. AI-ul muta random
2. AI-ul muta o data random, alta data folosind cea mai buna mutare posibila
3. AI-ul muta de fiecare data folosind cea mai buna mutare posibila

INPUT:

OUTPUT:

Name: ArchiveCracker

ID:8

Difficulty: B

Propose: MKY

Scrieti un tool care primeste ca paramtru o arhiva parolata ZIP si care ghiceste parola arhivei. Parola va fi alfanumerica si va avea maxim 10 caractere. Nu se pot folosi tool-uri externe.

INPUT:

Calea catre arhiva

OUTPUT:

Parola gasita

Name: Spider

ID:9

Difficulty: B

Propose: MKY

Scrieti un tool care pornind de la pagina web si un tag primiti ca parametri listeaza traseul parcurs de tool atunci cand acceseaza toate paginile ce au acel tag.
Tool-ul va considera un tag valid daca link-ul contine acel tag in interiorul campului <a ...>

Daca intr-o pagina sunt mai multe pagini ce contin acel tag vor fi accesate toate

TODO: de clarificat cu un exemplu mai bun

INPUT:

Link de pornire

Tag

OUTPUT:

Traseul parcurs de tool: Nume link si dimensiune pagina accesata

Logul de rulare si erorile aparute

Name: LocalCalendar

ID:10

Difficulty: B

Propose: NIP

Pentru un set de evenimente stocate local, creati un tool care sa genereze alerte inainte de un anumit eveniment. La fiecare rulare tool-ul va parsa evenimentele stocate si va genera alerte pentru evenimentele ce urmeaza (in functie de campul alert)

INPUT:

Evenimente (sedinte / meetups / zile de nastere / etc) ce pot fi in minim doua formate: formatul ICS si intr-un format stabilit de dezvoltator (ex: ceva json custom)

OUTPUT:

Alerte (pe ecran sau intr-un fisier) pentru evenimentele ce urmeaza.

Logurile rularii si eventualele erori

Name: ArchiveSlice

ID:11

Difficulty: B

Propose: MKY

Sa se creeze un tool slicer.py care sa aiba urmatoarele optiuni:

- Creare arhiva : pentru un director primit ca parametru si o lista de extensii, creaza o arhiva cu toate fisierele ce au acele extensii
- Slice : pentru o arhiva primita ca parametru va genera intr-un folder de output o feliere a arhivei - feliile vor avea dimensiune variabila si ca nume hash pe continut
- Restore: pentru un folder cu felii primit ca parametru, tool-ul va regenera arhiva

Arhiva este un format gandit de voi (si nu ceva standardizat - zip, 7zip, etc). Deasemenea, nu trebuiei si compresie ci doar stocare a datelor in arhiva.

INPUT:

Creare: director cu fisiere, lista de extensii, cale arhiva

Slice: cale arhiva si cale director output

Restore: cale folder cu felii si cale arhiva de output

OUTPUT:

Output-ul comenzilor, logurile de rulare si eventualele erori

Name: ExpenseAlert	ID:12	Difficulty: B	Propose: NIP
<p>Creati un tool care gestioneaza chetuielile intr-o companie si genereaza alerta cand pentru o anumita categorie de cheltuieli se depaseste o anumita limita de buget prestabilita.</p> <p>Tool-ul va monitoriza un director in care apar facturi si va actualiza informatiile intr-o baza de date.</p> <p>Utilizatorul poate seta pentru anumite cheltuieli categorii generale (ex: Chirie ar fi in categoria Administrative). Daca pentru o anumite cheltuiala nu este specificata categoria (in factura sau in baza de date) se va considera automat ca face parte din categoria "Diverse". Toate cheltuielile din categoria Diverse pot fi catalogate manual. Formatul unui fisier de tip factura este ceva ce puteti decide voi (ex: un fisier JSON sau fisier XML)</p>			
<p>INPUT:</p> <p>Director cu facturi ce va fi monitorizat.</p> <p>Fisier cu lista de praguri pentru categorii de cheltuieli (formatele fisierelor vor fi stabilite de dezvoltator)</p>			
<p>OUTPUT:</p> <p>Alertele depasire limita pentru anumite categorii - in momentul in care apar</p> <p>Loguri pentru operatiuni si erorile cand apar</p>			

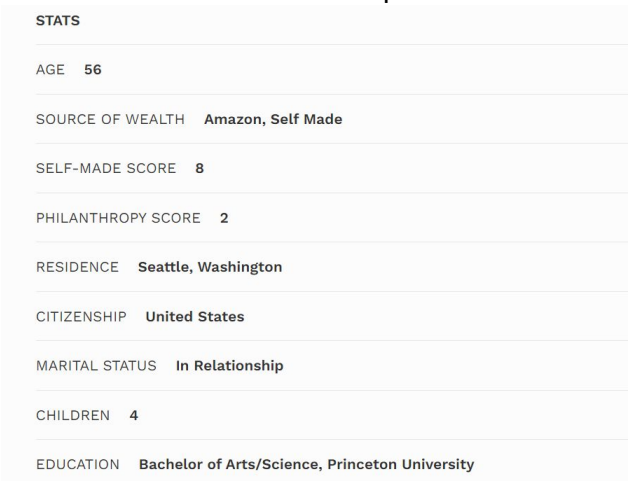
Name: GUI API Client	ID:13	Difficulty: B	Propose: ZAR
<p>Implementați o aplicație GUI cu ajutorul căruia un utilizator poate să facă requesturi HTTP către diverse API endpoints.</p> <p>Pentru fiecare request făcut de către user, se va afișa și răspunsul requestului. (într-un mod asemănător ca https://www.postman.com/product/api-client/)</p> <p>Pentru partea de GUI se poate folosi tkinter sau alte biblioteci asemănătoare.</p> <p>Trebuie implementat si un feature de tip "istoric" (requesturile și răspunsurile acestora să poată fi accesate ulterior)</p> <p>Pentru un request, utilizatorul va putea seta metoda, urlul, headers, payload</p>			
<p>INPUT:</p>			
<p>OUTPUT:</p>			

Name: GUI News Feed	ID:14	Difficulty: B	Propose: ZAR
<p>Implementați o aplicație GUI cu ajutorul căruia un utilizator poate să vizioneze cele mai recente news feeds legat de un anume subiect.</p> <p>Informațiile ar trebui să provină din diverse surse (cel puțin patru surse), de preferat folosindu-se API-urile puse la dispoziție de platformele respective (ex: Reddit, BBC, Google News, etc.)</p> <p>Pentru partea de GUI se poate folosi tkinter sau alte biblioteci asemănătoare.</p>			
INPUT:			
OUTPUT:			

Name: GUI Fact of the day	ID:15	Difficulty: B	Propose: ZAR
<p>Implementați o aplicație GUI cu ajutorul căruia utilizatorul primește la intervale regulate de timp (ex: 15 minute) scurte noutăți din anumite domenii definite de el (ex:astronomie, biologie, medicină, etc).</p> <p>Aplicația o să caute pe diverse site-uri de profil cele mai noi articole din domeniile respective și o să populeze în interfață câmpurile pentru Titlu, Descriere, Poza (daca feedul site-urilor respective pune la dispoziție)</p> <p>Pentru partea de GUI se poate folosi tkinter sau alte biblioteci asemănătoare.</p> <p>Obs: pentru inspirație: https://feedly.com/i/welcome</p>			
INPUT:			
OUTPUT:			

Name: GUI FTP Manager	ID:16	Difficulty: B	Propose: ZAR
<p>Implementați o aplicație GUI cu ajutorul căreia userul se poate conecta la diverse servere FTP și face operații de baza (CRUD) pe fișierele de acolo.</p> <p>Aplicația ar trebuie să permită userului să își salveze credențialele sub forma de sesiuni.</p> <p>Pentru partea de GUI se poate folosi tkinter sau alte biblioteci asemănătoare.</p>			
INPUT:			
OUTPUT:			

Name: Razboi	ID:17	Difficulty: B	Propose: RMC
<p>Se va crea un program cu o interfata grafica ce va permite utilizatorului sa joace impotriva calculatorului jocul de carti razboi. http://www.boardgames-blog.ro/board-games/regulamente-jocuri-de-carti-razboi/</p>			
INPUT: war.py			
OUTPUT: o interfata grafica care sa afiseze cele 2 carti in paralel, cea a jucatorului si a calculatorului si sa pastreze un scor pentru fiecare jucator.			

Name: TOP FORBES	ID:18	Difficulty: B	Propose: RMC
<p>Sa se creeze un crawler care sa preia informatiile de pe forbes despre cei mai bogati 200 de oameni si sa stocheze informatii despre ei intr-o baza de date. De asemenea, sa se creeze un set de functii care sa interactioneze cu baza de date.</p> <ol style="list-style-type: none"> 1. O functie care sa returneze top 10 cele mai tinere persoane din forbes 2. O functie care sa returneze cate persoane au cetatenie americana sau nu 3. O functie care sa returneze top 10 cel mai mare scor filantropic. 			
 <p>The screenshot shows a 'STATS' section for a person. It lists various attributes: AGE (56), SOURCE OF WEALTH (Amazon, Self Made), SELF-MADE SCORE (8), PHILANTHROPY SCORE (2), RESIDENCE (Seattle, Washington), CITIZENSHIP (United States), MARITAL STATUS (In Relationship), CHILDREN (4), and EDUCATION (Bachelor of Arts/Science, Princeton University).</p>			
<p>Site.ul suport este: https://www.forbes.com/billionaires/.</p>			
INPUT: forbes.py			
OUTPUT: un crawler care insereaza in baza de date informatii.			

Name: Spanzuratoarea	ID:19	Difficulty: B	Propose: RMC
<p>Se va crea o aplicatie de tip client server care va reproduce jocul de cuvinte "Spanzuratoarea". Vor exista 2 clienti, unul care va da cuvantul si o mica explicatie catre</p>			

server, iar celalalt care va raspunde cu cate o litera pana fie reuseste sa ghiceasca cuvantul, pana fie pierde.

INPUT: server.py
client1.py(care trimite cuvantul si o mica definitie)
client2.py(trimite cate o litera si va primi de la server o linie pe care va scrie:
___ c __ 5 daca litera "c" este ok, _ _ _ _ _ 4 daca litera "c" nu este ok).

OUTPUT:

Name: Rock - Paper - Scissors - Lizard - Spock	ID:20	Difficulty: B	Propose: ALM
---	--------------	----------------------	---------------------

Se va crea o aplicatie de tip server-client care va reproduce jocul de **Rock - Paper - Scissors - Lizard - Spock**. Clientul va trimite optiunea catre server, moment in care serverul va genera random si el o optiune si ii va transmite clientului daca a castigat sau nu. De asemenea, trebuie sa existe posibilitatea sa se conecteze la server pana la maxim 3 clienti care sa poata juca simultan un acel joc. Fiecare joaca independent cu serverul. (ex. Daca jucatorul 1 pierde, atunci partida se continua pentru jucatorul 2 si jucatorul 3. In continuare, daca jucatorul 3 pierde, jucatorul 2 isi continua jocul cu serverul pana pierde / castiga)

INPUT: server.py
client.py

OUTPUT: Raspunsul catre fiecare client - daca a castigat sau nu – impreuna cu toate optiunile alese.

Name: GITHUB MANAGER	ID:21	Difficulty: B	Propose: ALM
-----------------------------	--------------	----------------------	---------------------

Sa se creeze un program care sa permita posibilitatea automatizarii procesului de a incarca fisiere pe un repository.
Daca repository-ul nu exista local, se va clona in directorul dat ca parametru.
La un interval de x ore :

- Se verifica Github pentru noi modificari – pentru a se face actualizarea locala
- Se incarca fisierele modificate sau cele noi, spre a le actualiza pe Github
- Daca exista conflicte la urcare, rezolvați conflictele din fisierele afectate, ștergând porțiunile modificate remote și păstrându-le pe cele locale.

Se poate folosi modulul *git*.

INPUT:git_manager.py <repository name, directory where repository is>

OUTPUT: Se va observa in repo-ul de github de test

Name: GUESS THE NUMBER	ID:22	Difficulty: B	Propose: ALM
<p>Se va crea o aplicatie de tip server - client care va reproduce jocul Ghicește Numărul. Numărul ce trebuie ghicit se va afla în intervalul [0, 50]. Vor exista 2 posibilități: fie numărul va fi generat de server, fie va fi dat de către un alt client - în situația în care vor să joace 2 persoane. La fiecare încercare de a ghici numărul, clientul va primi unul din mesajele : numărul este corect / numărul este mai mic / mai mare decât numărul ales. Fiecare rulare a scriptului va reprezenta o sesiune de joc, formată din mai multe partide de joc. La finalul sesiunii, se va afișa scorul maxim – care va reprezenta cel mai mic număr de încercări necesar descoperirii numărului.</p>			
<p>INPUT: server.py client1.py(care trimite un număr sau nu) client2.py(este clientul care trimite câte un număr catre server, orice decizie luată de server fiind afișată ambilor participanți – ex. Client 1 ghiceste -> notificați ambii participanți că jocul s-a terminat, cu Scorul și Scorul Maxim aferent) .</p>			
<p>OUTPUT:</p>			

Name:	ID:23	Difficulty:B	Propose: CGI
<p>Creati o aplicatie grafica de tip client-server. Clientul va trimite catre server un titlu de film iar serverul va intoarce un rating. Daca titlul filmului nu se gaseste in baza de date in care va intergoa serverul , acesta va prelua rating-ul filmului (daca e posibil) de pe un site precu imdb sau rottentomatoes, adaugand filmul in baza de date pentru o interogare ulterioara. Interfata trebuie sa suporte cautare dupa: actor, nume de film si sa poata returna si link-uri la trailere de pe Youtube, respectiv o lista cu recenzii aferente acelui film.</p>			
<p>INPUT: titlu de film</p>			
<p>OUTPUT: rating-ul / scorul filmului (0-5 sau 0/100)</p>			

Name: Battleships	ID:24	Difficulty: B	Propose: ALM
--------------------------	--------------	----------------------	---------------------

Se va crea o aplicatie, cu o interfață grafică sugestivă, care va simula jocul Battleships. Descrierea jocului : https://en.wikipedia.org/wiki/Battleship_(game) Aplicatia va trebui să permită jocul între doi jucători, cu un singur vas de dimensiune fixa.
INPUT: battleships.py
OUTPUT: interfața grafică cu aferență jocului

Name: Browser ProcessViewer	ID:25	Difficulty: B	Propose: GDT
Realizati un script in python care sa permita anumite operații pe procese: <ul style="list-style-type: none"> • Vizualizare procese curente care rulează pe masina (PID, Nume, Path) • Suspend/resume pentru un process • Posibilitatea de a porni/opri un process (inclusiv cu o linie de comanda aferentă) • Informații despre procesele care rulează (cat procesor folosesc, cata memorie folosesc) 			
INPUT: process.py view ⇒ listeaza toate procesele + informatii aditionale per process process.py suspend <PID> ⇒ suspenda procesul cu un anumit PID process.py kill <PID> ⇒ opreste procesul cu un anumit PID process.py run path parametri ⇒ porneste un process			
OUTPUT:			

Name: OwnSSH	ID:26	Difficulty: B	Propose: WLD
Realizați un utilitar care să simuleze funcționalitatea unei conexiuni SSH. Utilitarul va avea două componente (client / server). Serverul va fi pornit cu un set de chei publice și va aștepta inițierea conexiunilor de către clienți. Clienții se vor conecta la server și se vor autentifica folosind o cheie privată. În continuare, clientul va putea trimite comenzi de shell iar serverul va returna output-ul acestora.			
INPUT: ownssh.py -s <public_key_1, public_key_2, ...> ownssh.py -c <ip> <private_key>			
OUTPUT: Output-ul comenzilor rulate de către client, cât și salvarea logurilor pe server.			

Name: Diagrame Karnaugh	ID:27	Difficulty: B	Propose: GDT
Realizati o aplicatia care stie sa faca minimizarea unor funcții booleene folosind diagrame			

<p>Karnaugh pentru funcțiile booleene cu 3 sau 4 variable. Aplicatia va primi la intrare un string de forma "sigma(...)" sau "sigma(...)+sigma*(...)" și va face următoarele:</p> <ul style="list-style-type: none"> • Afișează tabelul de adevăr corespunzător • Afișează forma minimizată (după procesarea prin diagrama Karnaugh) • FNC și FND <p>Aplicația știe să afișeze o eroare dacă forma sigma nu e corectă (valorile implică mai mult de 4 sau sub 3 variabile, respectiv sintaxa nu e corectă).</p>
INPUT: karnaugh.py "sigma(1,5,9,13)+sigma*(3,7,11,15)"
OUTPUT: f(x,y,z,t) = t + un tabel de adevăr (scris într-un fișier text) + FNC + FND.

Name: Network mapper	ID:28	Difficulty: B	Propose: WLD
<p>Realizați un utilitar care să primească o clasă de IP-uri în format CIDR și să afișeze nodurile din rețea care sunt online, cât și porturile care acceptă conexiuni (se vor testa doar cele mai utilizate porturi, sau cele dintr-o listă dată la linia de comandă).</p>			
INPUT: network_mapper.py 192.168.0.0/24 [port1, port2, ...]			
OUTPUT: 192.168.0.1:80 (HTTP) :443 (HTTPS) 192.168.0.105:445 (SMB)			

Name: Metoda celor mai mici patrate	ID:29	Difficulty: B	Propose: TDY
<p>Realizați un utilitar pentru calcularea și afișarea ecuației de aproximare a unei funcții folosind metoda celor mai mici patrate (liniar). Exemplu: https://www.mathsisfun.com/data/least-squares-calculator.html https://www.mathsisfun.com/data/least-squares-regression.html</p>			
INPUT: coordonatele punctelor			
OUTPUT:			

Name: Password manager	ID:30	Difficulty: B	Propose: WLD
<p>Realizați un utilitar care să gestioneze o listă de parole sub forma unui fișier, care poate fi actualizat (insert / update / delete), înăuntrul căruia parolele sunt criptate (cu o parolă master ce va trebui introdusă la deschiderea fișierului).</p>			
INPUT: pwmanager.py <master_password> -<operation> <website> <username> <password>			

<p>pwmanager.py <master_password> -add gmail.com johndoe@google.com cookie123</p> <p>pwmanager.py <master_password> -get gmail.com</p> <p>pwmanager.py <master_password> -remove gmail.com</p> <p>pwmanager.py <master_password> -list</p>
OUTPUT: pwmanager.db

Name: Memo	ID:31	Difficulty: B	Propose: TDY
Realizati o aplicatie grafica de tip Memo in care utilizatorul isi poate crea notite sub forma de text sau liste (de exemplu de cumparaturi). Datele vor fi salvate de la o utilizare la alta intr-un sqllite.			
INPUT			
OUTPUT:			

Name: Partition statistics	ID:32	Difficulty: B	Propose: TDY
Realizati o aplicatie care primeste ca parametru o partitie si analizeaza toate directoarele si fisierele de acolo. La final, aceasta va afisa numarul de directoare si de fisiere, proportia fiecarui tip de fisier (.zip, .exe etc.) atat ca numar cat si ca size. Toate aceste informatii vor fi afisate sub forma de charturi (pie chart, bar chart etc.) a.i. sa fie cat mai usor de intepratat.			
INPUT analyze_partition.py D			
OUTPUT:			

Name: Secret share	ID:33	Difficulty: B	Propose: WLD
Realizați un utilitar care să împartă un fișier în n fișiere. Pentru a recompune fișierul, va fi necesară prezența a cel puțin m dintre ele.			
INPUT: secretshare.py -split 3 2 secret.txt secretshare.py -recompose file1.secret file2.secret OR secretshare.py -recompose file2.secret file3.secret OR secretshare.py -recompose file1.secret file3.secret file1 -> part1 și part2 file2 -> part2 și part3 file3 -> part1 și part3			
OUTPUT:			

Name: Crawling	ID:34	Difficulty: B	Propose: MLA
Creati o aplicatie care permite utilizatorului sa urmareasca modificarile din anumite pagini web. Paginile web vor fi salvate intr-o baza de date. Utilizatorul va avea posibilitatea sa adauge, sa stearga sau sa modifice paginile web salvate. In momentul in care apare o modificare pe una din pagini, userul va fi notificat printr-un email. Modificarile vor fi cautate recursiv in pagina web.			
INPUT: https://en.wikipedia.org/wiki/Web_crawler			
OUTPUT: Pagina web a fost modificata.			

Name: CRAWLER PYTHON COURSE PAGE	ID: 35	Difficulty: B	Propose: RMC
Se va crea un crawler care va lua toate informatiile de pe pagina de laboratoare a site-ului de python. Pe baza problemelor extrase se vor crea subdirectoare intr-un director specificat, de forma lab{d} , ce vor contine fisiere de forma: lab{d}.py . In interiorul acestora se vor crea functiile conform informatiilor de pe pagina. Doar numele functiei in cazul in care exista va fi luat in considerare, altfel numarul exercitiului de forma: ex{d}. De exemplu: <pre>""" def ex1(parametru): pass """</pre>			
INPUT: Directorul unde se vor crea subdirectoarele cu fisierele .py Template.py <director>			
OUTPUT: Directorul completat cu cerintele de mai sus			

Name: Game of 2048	ID: 36	Difficulty: B	Propose: GDT
Se va crea o interfata grafica ce va oferi utilizatorului posibilitatea sa joace o partida 2048			
INPUT: 2048.py <tip adversar>			
OUTPUT: Interfata grafica cu tabla de 2048. După terminarea jocului se va afișa un mesaj corespunzător.			

Name: Game of the Pharaon	ID: 37	Difficulty: B	Propose: GDT
Se va crea o interfata grafica ce va oferi utilizatorului posibilitatea sa joace un joc similar cu cel de pe acest link: https://www.mathplayground.com/logic_stones_of_the_pharoah.html			

INPUT: pharaon.py <tip adversar>
OUTPUT: Interfata grafica cu tabla de de similara cu cea din linkul de mai sus.. După terminarea jocului se va afișa un mesaj corespunzător.

Name: Sudoku	ID:38	Difficulty: B	Propose: RMC
<p>Se va crea o aplicatie grafica care va simula jocul Sudoku. https://en.wikipedia.org/wiki/Sudoku De asemenea va exista si countdown pentru rezolvarea jocului. Jocul se termina fie cand toate sunt aranjate corect, fie cand timpul a expirat.</p>			
INPUT: sudoku.py			
OUTPUT: interfata grafica cu jocul			

Name: BigBrother	ID:39	Difficulty: B	Propose: NIP
<p>Creati un tool BiGBrother care primeste ca parametru un fisier cu amenintari (cuvinte cheie) si un folder in care vor aparea mesaje ale suspectilor de atacuri.. Tool-ul va monitoriza folderul de input si pentru fiecare mesaj nou aparut in folder va cauta daca apare vreunul din cuvintele cheie din fisierul de amenintari. Daca in timp ce ruleaza tool-ul detecteaza ca se modifica lista de cuvinte cheie, si-o va actualiza si el. Tool-ul va tine cont de faptul ca pe disk in folder pot aparea mesaje de dimensiune foarte mare (ex: 70 GB)</p>			
INPUT: Fisierul ce contine cuvinte cheie Folder in care apar mesajele			
OUTPUT: Pe ecran sau intr-un fisier de alerta se vor afisa fisierele cu mesaje ce contin civinte cheie, precum si locatia in fisier + textul de pe langa (+- 100 caractere)			