

Enunțul problemei:

Creați o aplicație pentru gestiunea concurenților de la un concurs de programare. Programul înregistrează scorul obținut de fiecare concurent la 10 probe diferite, fiecare probă este notată cu un scor de la 1 la 10. Fiecare participant este identificat printr-un număr de concurs, iar scorul este ținut într-o listă unde concurentul 3 are scorul pe poziția 3 în listă.

Lista de funcționalități ale aplicației:

1. **Adaugă un scor la un participant.**
 - a. Adaugă scor pentru un nou participant (ultimul participant)
2. **Modificare scor.**
 - a. Șterge scorul pentru un participant dat.
 - b. Șterge scorul pentru un interval de participanți.
 - c. Înlocuiește scorul de la un participant.
3. **Tipărește lista de participanți.**
 - a. Tipărește participanții care au scor mai mic decât un scor dat.
 - b. Tipărește participanții ordonat după scor
 - c. Tipărește participanții cu scor mai mare decât un scor dat, ordonat după scor.
4. **Operații pe un subset de participanți.**
 - a. Calculează media scorurilor pentru un interval dat (ex. Se da 1 și 5 se tipărește media scorurilor participanților 1,2,3,4 și 5
 - b. Calculează scorul minim pentru un interval de participanți dat.
 - c. Tipărește participanții dintr-un interval dat care au scorul multiplu de 10.
5. **Filtrare.**
 - a. Filtrare participanți care au scorul multiplu unui număr dat. Ex. Se da numărul 10, se elimină scorul de la toți participanții care **nu** au scorul multiplu de 10.
 - b. Filtrare participanți care au scorul mai mic decât un scor dat.
6. **Undo**
 - a. Reface ultima operație (lista de scoruri revine la numerele ce existau înainte de ultima operație care a modificat lista) – **Nu** folosiți funcția **deepCopy**

Planul de iterații:

Iterația	Funcționalități planificate
I1 (laboratorul 4)	1.a. 3.a. 4.a., 4.c. 5.b.
I2 (laboratorul 5)	2.a., 2.b., 2.c. 3.b., 3.c.
I3 (laboratorul 6)	4.b. 5.a. 6.a + batch mode

Scenarii de rulare:

➤ Pentru funcționalitatea 1

	Utilizator	Program	Descriere
A		<p><code>lista_toti = []</code></p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p><code>>>></code></p>	<p>Se inițializează o listă, vidă la început (nu se tipărește)</p> <p>Este tipărit meniul aplicației</p>
B	1	<ol style="list-style-type: none"> a. Creeaza un nou concurent b. Adauga scorurile pentru cele 10 probe 	Este tipărit „submeniul” corespunzător punctului 1 din meniu
C	a	Introduceti id-ul noului participant (numar natural nenul):	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	137	<p><code>id_participant = 137</code></p> <p>Introdu 10 valori reale (reprezentand scorurile pentru cele 10 probe), cuprinse intre 1 si 10: Scor proba 1 :</p>	<p>Programul pune în variabila <code>id_participant</code> id-ul citit</p> <p>Se tipărește ceea ce trebuie să introducă utilizatorul – scorurile pentru probele de concurs</p>
E	6	<code>s = 6</code>	Se creeaza o variabila in care se aduna scorurile introduse
F	8.5	<code>s = 14.5</code>	- // -
G	7	<code>s = 21.5</code>	- // -
H	5.5	<code>s = 27</code>	- // -
I	9	<code>s = 36</code>	- // -
J	9.4	<code>s = 45.4</code>	- // -
K	7.3	<code>s = 52.7</code>	- // -
L	6.8	<code>s = 59.5</code>	- // -
M	10	<code>s = 69.5</code>	- // -
N	7.7	<p><code>s = 77.3</code></p> <p><code>scor = 77.3</code></p> <p><code>participant = [137,77.3]</code></p>	<p>- // -</p> <p>Pune în scor suma calculata</p>

		<p>lista_toti = [[137,77.3]]</p>	<p>Creează participantul (sub forma unei liste), cu id_participant și scor</p> <p>Pune în lista participanților primul participant</p> <p>În acest moment, dat fiindcă s-a introdus a 10-a valoare validă, participantul a fost creat cu succes</p>
O		<p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>>></p>	<p>Este retipărit meniul, iar utilizatorul este nevoit să introducă o nouă comandă</p>
P	1	<p>a. Creeaza un nou concurent</p> <p>b. Adauga scorurile pentru cele 10 probe</p>	<p>Este tipărit „submeniul” corespunzător punctului 1 din meniu</p>
Q	a	<p>Introduceti id-ul noului participant (numar natural nenul):</p>	<p>Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului</p>
R	79	<p>id_participant = 79</p> <p>Introdu 10 valori reale (reprezentand scorurile pentru cele 10 probe), cuprinse intre 1 si 10:</p> <p>Scor proba 1 :</p>	<p>Programul pune în variabila id_participant id-ul citit</p> <p>Se tipărește ceea ce trebuie să introducă utilizatorul – scorurile pentru probele de concurs</p>
S – B1	9, 8.7, 7.9, 9.2, 10, 6.8, 5.2, 9.7, 7, 8.2	<p>s = 9, 17.7, 25.6, 34.8, 44.8, 51.6, 56.8, 66.5, 73.5, 81.7</p> <p>scor = 81.7</p> <p>participant = [79, 81.7]</p> <p>lista_toti = [[137,77.3], [79, 81.7]]</p>	<p>Pune în scor suma calculata</p> <p>Creează participantul (sub forma unei liste), cu id_participant și scor</p> <p>Pune în lista participanților al doilea participant</p> <p>În acest moment, dat fiindcă s-a introdus a 10-a valoare validă, participantul a fost creat cu succes</p>

C1		<p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	Este retipărit meniul, iar utilizatorul este nevoit să introducă o nouă comandă
D1	1	<ol style="list-style-type: none"> a. Creeaza un nou concurent b. Adauga scorurile pentru cele 10 probe 	Este tipărit „submeniul” corespunzător punctului 1 din meniu
E1	a	Introduceti id-ul noului participant (numar natural nenul):	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
F1	288	<p>$id_participant = 288$</p> <p>Introdu 10 valori reale (reprezentand scorurile pentru cele 10 probe), cuprinse intre 1 si 10: Scor proba 1 :</p>	<p>Programul pune în variabila $id_participant$ id-ul citit</p> <p>Se tipărește ceea ce trebuie să introducă utilizatorul – scorurile pentru probele de concurs</p>
G1	7.3, 8, 7.1, 9.9, 6, 8.8, 7.2, 9, 8, 7.8	<p>$s = 7.3, 15.3, 22.4, 32.3, 38.3, 47.1, 54.3, 63.3, 71.3, 79.1$ $scor = 79.1$ $participant = [288, 79.1]$</p> <p>$lista_toti = [[137, 77.3], [79, 81.7], [288, 79.1]]$</p>	<p>Pune în scor suma calculata</p> <p>Creează participantul (sub forma unei liste), cu $id_participant$ și scor</p> <p>Pune în lista participanților al doilea participant</p> <p>În acest moment, dat fiindcă s-a introdus a 10-a valoare validă, participantul a fost creat cu succes</p>
H1		<p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p>	Este retipărit meniul, iar utilizatorul este nevoit să introducă o nouă comandă

		Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	
--	--	--	--

➤ Pentru funcționalitatea 3 (a)

	Utilizator	Program	Descriere
A		<p>lista_toti = []</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>>></p>	<p>Se inițializează o listă, vidă la început (nu se tipărește)</p> <p>Este tipărit meniul aplicației</p>
B	3	<ol style="list-style-type: none"> a. Tipareste participantii care au un scor mai mic decat unul dat b. Tipareste participantii ordonat crescator c. Tipareste participantii care au un scor mai mare decat unul dat, ordonati dupa scor 	Este tipărit „submeniul” corespunzător punctului 3 din meniu
C	a	Dati un scor, pentru a afisa participantii cu scor mai mic decat respectivul ($0 \leq \text{scor} \leq 100$):	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	80	<p>137 288</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 	<p>Folosind exemplul de la scenariul anterior, va afișa codurile de identificare ale concurenților</p> <p>Se tipărește ceea ce trebuie să introducă utilizatorul – scorurile pentru probele de concurs</p>

		5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>	
--	--	--	--

➤ Pentru funcționalitatea 4 (a)

	Utilizator	Program	Descriere
A		<p><code>lista_toti = []</code></p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	<p>Se inițializează o listă, vidă la început (nu se tipărește)</p> <p>Este tipărit meniul aplicației</p>
B	4	<ol style="list-style-type: none"> a. Calculeaza media scorurilor pentru un interval dat b. Calculeaza scorul minim pentru un interval de participanti dat c. Tipareste participantii dintr-un interval dat care au scorul multiplu de 10 	Este tipărit „submeniul” corespunzător punctului 4 din meniu
C	a	<p>Pentru a introduce intervalul, dati, pe rand, doua valori numere naturale, a si b, cu $a < b$:</p> <p>Introduceti unde incepe intervalul:</p>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	2	<p><code>poz1 = 2</code></p> <p>Introduceti unde se termina intervalul:</p>	<p>Programul pune în variabila <code>poz1</code> valoarea 2</p> <p>Se tipărește ceea ce trebuie să introducă utilizatorul – poziția de final</p>
E	3	<p><code>poz2 = 3</code></p>	Unde se încheie intervalul (corect definit)
F		Media scorurilor participantilor cuprinsi in intervalul dat este: 80.4	E tipărit rezultatul

		Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Retipărirea meniului
--	--	--	----------------------

➤ Pentru funcționalitatea 4 (b)

	Utilizator	Program	Descriere
A		lista_toti = [] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Se inițializează o listă, vidă la început (nu se tipărește) Este tipărit meniul aplicației
B	4	a. Calculeaza media scorurilor pentru un interval dat b. Calculeaza scorul minim pentru un interval de participanti dat c. Tipareste participantii dintr-un interval dat care au scorul multiplu de 10	Este tipărit „submeniul” corespunzător punctului 4 din meniu
C	a	Pentru a introduce intervalul, dati, pe rand, doua valori numere naturale, a si b, cu $a < b$: Introduceti unde incepe intervalul:	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	2	poz1 = 2 Introduceti unde se termina intervalul:	Programul pune în variabila poz1 valoarea 2 Se tipărește ceea ce trebuie să introducă utilizatorul – poziția de final

E	3	poz2 = 3	Unde se încheie intervalul (corect definit)
F		<p>Media scorurilor participantilor cuprinsi in intervalul dat este: 80.4</p> <p>Aplicatia pentru laboratoare 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participant 4. Operatii pe un subset de participant 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	<p>E tipărit rezultatul</p> <p>Retipărirea meniului</p>

➤ **Pentru funcționalitatea 4 (c)**

	Utilizator	Program	Descriere
A		<p>lista_toti = []</p> <p>Aplicatia pentru laboratoare 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participant 4. Operatii pe un subset de participant 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	<p>Se inițializează o listă, vidă la început (nu se tipărește)</p> <p>Este tipărit meniul aplicației</p>
B	4	<ol style="list-style-type: none"> a. Calculeaza media scorurilor pentru un interval dat b. Calculeaza scorul minim pentru un interval de participant dat c. Tipareste participantii dintr-un interval dat care au scorul multiplu de 10 	Este tipărit „submeniul” corespunzător punctului 4 din meniu
C	c	<p>Pentru a introduce intervalul, dati, pe rand, doua valori numere naturale, a si b, cu a < b:</p> <p>Introduceti unde incepe intervalul:</p>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului

D	2	poz1 = 2 Introduceti unde se termina intervalul:	Programul pune în variabila poz1 valoarea 2 Se tipărește ceea ce trebuie să introducă utilizatorul – poziția de final
E	3	poz2 = 3	Unde se încheie intervalul (corect definit)
F		Nu exista concurenti cu scor multiplu de 10. Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>	E tipărit rezultatul Retipărirea meniului

➤ **Pentru funcționalitatea 2 (a)**

	Utilizator	Program	Descriere
A		lista_toti = [[137,77.3], [79, 81.7], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>	Presupunem că suntem în interiorul programului și avem deja 3 participanți. Este tipărit meniul aplicației

B	2	a. Sterge scorul pentru un participant dat b. Sterge scorul pentru un interval de participanti c. Inlocuieste scorul unui participant	Este tipărit „submeniul” corespunzător punctului 2 din meniu
C	a	Alegeti pentru care (unul) dintre urmatorii participanti (id) doriti sa stergeti scorul: 137 79 288 >>>>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	79	lista_toti = [[137,77.3], [79, 0], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Programul pune la scorul participantului cu id-ul introdus valoarea 0 Retipărirea meniului

➤ Pentru funcționalitatea 2 (b)

	Utilizator	Program	Descriere
A		lista_toti = [[137,77.3], [79, 81.7], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit.	Presupunem că suntem în interiorul programului și avem deja 3 participanți. Este tipărit meniul aplicației

		Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	
B	2	a. Sterge scorul pentru un participant dat b. Sterge scorul pentru un interval de participanti c. Inlocuieste scorul unui participant	Este tipărit „submeniul” corespunzător punctului 2 din meniu
C	b	Alegeti pentru care (unul) dintre urmatorii participanti (id) doriti sa stergeti scorul: 137 79 288 >>>>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	79	lista_toti = [[137,77.3], [79, 0], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Programul pune la scorul participantului cu id-ul introdus valoarea 0 Retipărirea meniului

➤ **Pentru funcționalitatea 2 (c)**

	Utilizator	Program	Descriere
A		lista_toti = [[137,77.3], [79, 81.7], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare)	Presupunem că suntem în interiorul programului și avem deja 3 participanți. Este tipărit meniul aplicației

		Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	
B	2	a. Sterge scorul pentru un participant dat b. Sterge scorul pentru un interval de participanti c. Inlocuieste scorul unui participant	Este tipărit „submeniul” corespunzător punctului 2 din meniu
C	c	Alegeti pentru care (unul) dintre urmatorii participanti (id) doriti sa stergeti scorul: 137 79 288 >>>>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id- ul participantului
D	79	lista_toti = [[137,77.3], [79, 0], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Programul pune la scorul participantului cu id-ul introdus valoarea 0 Retipărirea meniului

➤ **Pentru funcționalitatea 3 (b)**

	Utilizator	Program	Descriere
A		lista_toti = [[137,77.3], [79, 81.7], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie)	Presupunem că suntem în interiorul programului și avem deja 3 participanți. Este tipărit meniul aplicației

		<p>7. Afiseaza toti participantii (de verificare)</p> <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>>></p>	
B	3	<p>a. Tipareste participantii care au un scor mai mic decat unul dat</p> <p>b. Tipareste participantii ordonat crescator</p> <p>c. Tipareste participantii care au un scor mai mare decat unul dat, ordonati dupa scor</p>	Este tipărit „submeniul” corespunzător punctului 3 din meniu
C	b	<p>Alegeti pentru care (unul) dintre urmatorii participantii (id) doriti sa stergeti scorul:</p> <p>137 79 288</p> <p>>>>></p>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	79	<p>lista_toti = [[137,77.3], [79, 0], [288,79.1]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participantii 4. Operatii pe un subset de participantii 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>>></p>	<p>Programul pune la scorul participantului cu id-ul introdus valoarea 0</p> <p>Retipărirea meniului</p>

➤ Pentru funcționalitatea 3 (c)

	Utilizator	Program	Descriere
A		<p>lista_toti = [[137,77.3], [79, 81.7], [288,79.1]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 	<p>Presupunem că suntem în interiorul programului și avem deja 3 participanți.</p> <p>Este tipărit meniul aplicației</p>

		3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	
B	3	a. Tipareste participantii care au un scor mai mic decat unul dat b. Tipareste participantii ordonat crescator c. Tipareste participantii care au un scor mai mare decat unul dat, ordonati dupa scor >>>>	Este tipărit „submeniul” corespunzător punctului 3 din meniu
C	c	Alegeti pentru care (unul) dintre urmatorii participanti (id) doriti sa stergeti scorul: 137 79 288 >>>>	Este tipărit mesajul prin care utilizatorului i se comunica sa introduca id-ul participantului
D	79	lista_toti = [[137,77.3], [79, 0], [288,79.1]] Aplicatia pentru laboratoarele 4-6 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>>	Programul pune la scorul participantului cu id-ul introdus valoarea 0 Retipărirea meniului

➤ Pentru funcționalitatea 5 (a)

	Utilizator	Program	Descriere
A		<p>lista_toti = [[137,54.0], [79,44.0], [288,66.0]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>>></p>	<p>Presupunem că suntem în interiorul programului și avem deja 3 participanți.</p> <p>Este tipărit meniul aplicației</p>
B	5	<p>a. Filtrare participanti care au scorul multiplu unui numar dat.</p> <p>b. Filtrare participanti care au scorul mai mic decat un scor dat</p>	Este tipărit „submeniul” corespunzător punctului 5 din meniu
C	a	Dati un numar > 0 (de asemenea <=100), pentru a filtra participantii care nu au scorul multiplu al acelui numar:	Este tipărit mesajul prin care utilizatorului i se comunica sa introducă id-ul participantului
D	6	<p>Participantul 137 <-> 54.0.</p> <p>Participantul 79 <-> 0.0.</p> <p>Participantul 288 <-> 66.0.</p> <p>lista_toti = [[137,54.0], [79,44.0], [288,66.0]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p>	<p>Sunt afișați participanții, iar cei care nu au scor multiplu de 6 apar cu <-> 0.0, iar ceilalți cu scorurile lor, fără a fi modificată lista cu participanți.</p> <p>Retipărirea meniului</p>

		<p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	
--	--	---	--

➤ **Pentru funcționalitatea 5 (b)**

	Utilizator	Program	Descriere
A		<p>lista_toti = [[137,54.0], [79,44.0], [288,66.0]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	<p>Presupunem că suntem în interiorul programului și avem deja 3 participanți.</p> <p>Este tipărit meniul aplicației</p>
B	5	<p>a. Filtrare participanti care au scorul multiplu unui numar dat.</p> <p>b. Filtrare participanti care au scorul mai mic decat un scor dat</p>	<p>Este tipărit „submeniul” corespunzător punctului 5 din meniu</p>
C	b	<p>Dati un scor, pentru a filtra participantii cu scor mai mic decat respectivul ($10 \leq \text{scor} \leq 100$):</p>	<p>Este tipărit mesajul prin care utilizatorului i se comunica sa introducă id-ul participantului</p>
D	55	<p>Participantul 137 \leftrightarrow 0.0.</p> <p>Participantul 79 \leftrightarrow 0.0.</p> <p>Participantul 288 \leftrightarrow 66.0.</p> <p>lista_toti = [[137,54.0], [79,44.0], [288,66.0]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 	<p>Sunt afișați participanții, iar cei care au scor mai mic decât 55 apar cu \leftrightarrow 0.0, iar ceilalți cu scorurile lor, fără a fi modificată lista cu participanți.</p> <p>Retipărirea meniului</p>

		4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>	
--	--	--	--

➤ **Pentru funcționalitatea 6**

	Utilizator	Program	Descriere
A		<p>lista_toti = [[137,54.0], [79,44.0], [288,66.0]]</p> <p>Aplicatia pentru laboratoarele 4-6</p> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit. Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7) >>>	<p>Presupunem că suntem în interiorul programului și avem deja 3 participanți.</p> <p>Este tipărit meniul aplicației</p>
B	6	<p>Concurentul identificat cu id-ul 137 are scorul 54.0.</p> <p>Concurentul identificat cu id-ul 79 are scorul 44.0.</p> <p>Aplicatia pentru laboratoarele 4-6</p> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) Pentru a parasi aplicatia, scrie exit.	<p>Se face undo, presupunând ca ultima data s-a făcut adăugarea participantului 288.</p> <p>Retipărirea meniului</p>

		<p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p> <p>lista_toti = [[137,54.0], [79,44.0]]</p>	
C	6	<p>Concurentul identificat cu id-ul 137 are scorul 54.0.</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p> <p>lista_toti = [[137,54.0]]</p>	<p>Se face undo, presupunând ca ultima data s-a făcut adăugarea participantului 79.</p> <p>Retipărirea meniului</p>
D	6	<p>Nu exista participanti!</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 2. Modificare scor 3. Tipareste lista de participanti 4. Operatii pe un subset de participanti 5. Filtrare 6. Undo (reface ultima operatie) 7. Afiseaza toti participantii (de verificare) <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p> <p>lista_toti = []</p>	<p>Se face undo, presupunând ca ultima data s-a făcut adăugarea participantului 137.</p> <p>Retipărirea meniului</p>
E	6	<p>Nu mai sunt operatii pentru care sa se faca undo!</p> <p>Aplicatia pentru laboratoarele 4-6</p> <ol style="list-style-type: none"> 1. Adauga un scor la un participant 	<p>Retipărirea meniului</p>

		<p>2. Modificare scor</p> <p>3. Tipareste lista de participanti</p> <p>4. Operatii pe un subset de participanti</p> <p>5. Filtrare</p> <p>6. Undo (reface ultima operatie)</p> <p>7. Afiseaza toti participantii (de verificare)</p> <p>Pentru a parasi aplicatia, scrie exit.</p> <p>Alege una dintre optiunile de mai sus pentru a incepe/continua! (se va scrie doar cifra, de la 1 la 7)</p> <p>>>></p>	
--	--	--	--

Cazuri de testare:

Date de intrare (meniu, submeniu, chestiuni aferente)	Date de ieșire (sau ce este în spate)
1 a 37 1 2 3 4 5 6 7 8 9 1	lista_toti = [{"id_participant": 37, "scor": 46.0}]
1 a 122 1 2 3 6 5 6 7 8 9 4	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
1 a 37 10 2 3 6.4 5 6 7 8 9 4	Participant existent! Lista e nemodificată
1 a 22 F 1 2 3 4 5 23 6 7 8 -1 9 10	Nu ati introdus o valoare reala, cuprinsa intre 1 si 10! Nu ati introdus o valoare reala, cuprinsa intre 1 si 10! Nu ati introdus o valoare reala, cuprinsa intre 1 si 10! lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}, {"id_participant": 22, "scor": 55.0}]
1 b	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o litera (aici doar litera a) sau exit, daca doriti sa parasiti aplicatia.
2 a 122	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 0.0}, {"id_participant": 22, "scor": 55.0}]
2 a 555	Valoare invalida pentru id! Lista e nemodificată
2 b 2 3	

	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 0.0}, {"id_participant": 22, "scor": 0.0}]
2 b 2 4	Nu ati introdus valori corecte pentru un interval, adica a < b si b <= numarul total de participanti.
2 c 22 3 3 3 3 3 3 3 3 3 3	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 0.0}, {"id_participant": 22, "scor": 30.0}]
2 c 11	Valoare invalida pentru id! Lista e nemodificată
2 c gads	Valoare invalida pentru id! Lista e nemodificată
2 r	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o litera (aici a,b sau c) sau exit, daca doriti sa parasiti aplicatia.
3 a 30	Participantii cautati sunt: 122
2 c 122 4 4 4 4 4 4 4 4 4 6	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 42.0}, {"id_participant": 22, "scor": 30.0}]
3 a 25	Nu exista participanti cu un scor mai mic decat cel introdus!
3 a 555	Scorul introdus nu este corespunzator!
3 a frate	Scorul introdus nu este corespunzator!
3 b	Participantul 22, cu scorul: 30.0 Participantul 122, cu scorul: 42.0

	Participantul 37, cu scorul: 46.0
3 c 35	Participantul 122, cu scorul: 42.0 Participantul 37, cu scorul: 46.0
3 c 2	Scorul introdus nu este corespunzator!
3 nimic	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o litera (aici a sau c) sau exit, daca doriti sa parasiti aplicatia.
4 a 1 2	Media pentru intervalul dat este: 44.0
4 a 2 2	Nu ati introdus valori corecte pentru un interval, adica $a < b$ si $b \leq$ numarul total de participanti.
4 a f	Valoare invalida!
4 a 2 re	Valoare invalida!
4 a 2 4	Nu ati introdus valori corecte pentru un interval, adica $a < b$ si $b \leq$ numarul total de participanti.
4 b 1 3	Lista participantilor care au scorul minim din intervalul dat: Participantul cu id-ul: 22 si cu scorul minim, de 30.0
1 a 18 3 3 3 4 2 3 3 3 3	<code>lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 42.0}, {"id_participant": 22, "scor": 30.0}, {"id_participant": 18, "scor": 30.0}]</code>
4 b 2 4	Lista participantilor care au scorul minim din intervalul dat: Participantul cu id-ul: 22 si cu scorul minim, de 30.0

	Participantul cu id-ul: 18 si cu scorul minim, de 30.0
4 b 5 2	Nu ati introdus valori corecte pentru un interval, adica $a < b$ si $b \leq$ numarul total de participanti.
4 b gre	Valoare invalida!
4 b 2 Gre2	Valoare invalida!
4 c 1 4	Lista participantilor care au scorul multiplu de zece: 22 18
4 c 4 4	Nu ati introdus valori corecte pentru un interval, adica $a < b$ si $b \leq$ numarul total de participanti.
4 c 1 2	Nu exista concurenti cu scor multiplu de 10.
4 c gre3	Valoare invalida!
4 c 2 Gre4	Valoare invalida!
4 cuvant	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o litera (aici a sau c) sau exit, daca doriti sa parasiti aplicatia.
5 a 3	Participantul 37 \leftrightarrow 0.0. Participantul 122 \leftrightarrow 42.0. Participantul 22 \leftrightarrow 30.0. Participantul 18 \leftrightarrow 30.0.

5 a 7	Participantul 37 <-> 0.0. Participantul 122 <-> 42.0. Participantul 22 <-> 0.0. Participantul 18 <-> 0.0.
5 a 2	Nu sunt asemenea cazuri de filtrare!
5 a 31	Participantul 37 <-> 0.0. Participantul 122 <-> 0.0. Participantul 22 <-> 0.0. Participantul 18 <-> 0.0.
5 a 0	Scorul introdus nu este corespunzator!
5 a 101	Scorul introdus nu este corespunzator!
5 a cuv	Scorul introdus nu este corespunzator!
5 b 20	Nu sunt asemenea cazuri de filtrare!
5 b 40	Participantul 37 <-> 46.0. Participantul 122 <-> 42.0. Participantul 22 <-> 0.0. Participantul 18 <-> 0.0.
5 b 7	Scorul introdus nu este corespunzator!
5 ceva	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o litera (aici a sau c) sau exit, daca doriti sa parasiti aplicatia.
daram	Ce ati introdus nu corespunde niciuneia dintre comenzile aplicatiei! Introduceti o valoare numerica intreaga, cuprinsa intre 1 si 7 sau exit, daca doriti sa parasiti aplicatia.\n")
exit	SE IESE DIN PROGRAM

Consideram ca facem totul din nou, pentru a putea vedea cum funcționează undo	
1 a 37 1 2 3 4 5 6 7 8 9 1	lista_toti = [{"id_participant": 37, "scor": 46.0}]
1 a 122 1 2 3 6 5 6 7 8 9 4	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
6	lista_toti = [{"id_participant": 37, "scor": 46.0}]
6	Nu exista participant! lista_toti = []
1 a 37 1 2 3 4 5 6 7 8 9 1	lista_toti = [{"id_participant": 37, "scor": 46.0}]
1 a 122 1 2 3 6 5 6 7 8 9 4	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
2 a 122	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 0.0}]
6	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
6	lista_toti = [{"id_participant": 37, "scor": 46.0}]
1 a 122 1 2 3 6 5 6 7 8 9 4	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
2 b 1 2	lista_toti = [{"id_participant": 37, "scor": 0.0}, {"id_participant": 122, "scor": 0.0}]

6	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
2 c 37 3 3 3 3 3 3 3 3 3	lista_toti = [{"id_participant": 37, "scor": 30.0}, {"id_participant": 122, "scor": 51.0}]
6	lista_toti = [{"id_participant": 37, "scor": 46.0}, {"id_participant": 122, "scor": 51.0}]
6	lista_toti = [{"id_participant": 37, "scor": 46.0}]
6	Nu exista participanti! lista_toti = []
6	Nu mai sunt operatii pentru care sa se faca undo!

```

def test_creeaza_participant():
    id_participant = 18
    scor = 32.9
    participant = creeaza_participant(id_participant, scor)
    assert(get_id_participant(participant)==id_participant)
    assert(get_scor(participant)==scor)

def test_valideaza_participant():
    id_participant = 18
    scor = 32.9
    participant = creeaza_participant(id_participant, scor)
    valideaza_participant(participant)
    id_participant_invalid = "-19"
    scor_invalid = -8
    participant_invalid1 = creeaza_participant(id_participant_invalid, scor)
    try:
        valideaza_participant(participant_invalid1)
        assert(False)
    except ValueError as ve:
        assert(str(ve)=="invalid id participant!\n")
    participant_invalid2 = creeaza_participant(id_participant, scor_invalid)
    try:
        valideaza_participant(participant_invalid2)
        assert(False)
    except ValueError as ve:
        assert(str(ve)=="invalid scor participant!\n")
    participant_invalid3 = creeaza_participant(id_participant_invalid, scor_invalid)
    try:
        valideaza_participant(participant_invalid3)
        assert(False)
    except ValueError as ve:
        assert(str(ve)=="invalid id participant!\ninvalid scor participant!\n")

def test_adauga_participant_lista():
    lista = []
    id_participant = 18
    scor = 32.9
    participant = creeaza_participant(id_participant, scor)
    adauga_participant_lista(lista, participant)
    assert(len(lista)==1)
    assert(get_id_participant(participant)==get_id_participant(lista[0]))
    assert(get_scor(participant)==get_scor(lista[0]))
    participant_acelasi_id = creeaza_participant(id_participant, 72)
    try:
        adauga_participant_lista(lista, participant_acelasi_id)
        assert(False)
    except RepoError as re:
        assert(str(re)=="\nParticipant existent!\n")

```

```

def test_func2a():
    '''l_toti = [[32,54],[434,44],[4,65]]
    care = 434
    functionalitate2a(l_toti,care)
    assert(l_toti==[[32,54],[434,0],[4,65]])
    care = 4
    functionalitate2a(l_toti,care)
    assert(l_toti!=[[32,54],[434,0],[4,65]])'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    care = 434
    functionalitate2a(l_toti,care)
    assert(l_toti==[
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":0},
        {"id_participant":4, "scor":65}])
    care = 4
    functionalitate2a(l_toti,care)
    assert(l_toti!=[
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":0},
        {"id_participant":4, "scor":65}])

```

```

def test_func2b():
    '''l_toti = [[32,54],[434,44],[4,65]]
    poz1 = 1
    poz2 = 2
    functionalitate2b(l_toti,poz1,poz2)
    assert(l_toti==[[32,0],[434,0],[4,65]])
    poz1 = 2
    poz2 = 4
    assert(functionalitate2b(l_toti,poz1,poz2)==-1)'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    poz1 = 1
    poz2 = 2
    id_de_la_care = []
    scoruri_sterse = []
    functionalitate2b(l_toti,poz1,poz2,id_de_la_care,scoruri_sterse)
    assert(l_toti==[
        {"id_participant":32, "scor":0},
        {"id_participant":434, "scor":0},
        {"id_participant":4, "scor":65}
    ])
    poz1 = 2
    poz2 = 4
    assert(functionalitate2b(l_toti,poz1,poz2,id_de_la_care,scoruri_sterse)==-1)

```

```

def test_func2c():
    '''l_toti = [[32,54],[434,44],[4,65]]
    care = 434
    scor = 32
    functionalitate2c(l_toti,care,scor)
    assert(l_toti==[[32,54],[434,32],[4,65]])
    care = 4
    functionalitate2c(l_toti,care,59)
    assert(l_toti==[[32,54],[434,32],[4,59]])'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    care = 434
    scor = 32
    scor_initial = 0
    functionalitate2c(l_toti,care,scor,scor_initial)
    assert(l_toti==[
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":32},
        {"id_participant":4, "scor":65}
    ])
    care = 4
    functionalitate2c(l_toti,care,59,scor_initial)
    assert(l_toti==[
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":32},
        {"id_participant":4, "scor":59}
    ])

def test_func3a():
    ''' l_toti = [[32,54],[434,44],[4,65]]
    scor_dat = 55
    assert(functionalitate3a(l_toti,scor_dat)=="32 434 ")
    scor_dat = 14
    assert(functionalitate3a(l_toti,scor_dat)==-1)'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    scor_dat = 55
    assert(functionalitate3a(l_toti,scor_dat)=="32 434 ")
    scor_dat = 14
    assert(functionalitate3a(l_toti,scor_dat)==-1)

```

```

def test_func3b():
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    li_mea = functionalitate3b(l_toti)
    assert(li_mea==[{"id_participant":434, "scor":44},
        {"id_participant":32, "scor":54},
        {"id_participant":4, "scor":65}])

def test_func3c():
    '''l_toti = [[32,54],[434,44],[4,65]]
    scor_dat = 45
    li_mea = functionalitate3c(l_toti,scor_dat)
    assert(li_mea == [[32,54],[4,65]])
    scor_dat = 70
    li_mea = functionalitate3c(l_toti,scor_dat)
    assert(li_mea == [])'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":65}
    ]
    scor_dat = 45
    li_mea = functionalitate3c(l_toti,scor_dat)
    assert(li_mea == [
        {"id_participant":32, "scor":54},
        {"id_participant":4, "scor":65}
    ])
    scor_dat = 70
    li_mea = functionalitate3c(l_toti,scor_dat)
    assert(li_mea == [])

```

```

def test_func4a():
    '''l_toti = [[32,54],[434,44],[4,66]]
    poz1 = 2
    poz2 = 3
    med = functionalitate4a(l_toti,poz1,poz2)
    assert(med == 55)
    poz1 = 2
    poz2 = 7
    med = functionalitate4a(l_toti,poz1,poz2)
    assert(med == -1)'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":66}
    ]
    poz1 = 2
    poz2 = 3
    med = functionalitate4a(l_toti,poz1,poz2)
    assert(med == 55)
    poz1 = 2
    poz2 = 7
    med = functionalitate4a(l_toti,poz1,poz2)
    assert(med == -1)

def test_func4b():
    l_toti = [
        {"id_participant":32, "scor":50},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":60}
    ]
    poz1 = 1
    poz2 = 3
    scor_minim = functionalitate4b(l_toti,poz1,poz2)
    assert(scor_minim == [{"id_participant":434, "scor":44}])
    l_toti=[
        {"id_participant":32, "scor":41},
        {"id_participant":434, "scor":41},
        {"id_participant":4, "scor":64}
    ]
    poz1 = 1
    poz2 = 3
    scor_minim = functionalitate4b(l_toti,poz1,poz2)
    assert(scor_minim == [{"id_participant":32, "scor":41}, {"id_participant":434, "scor":41}])
    poz1 = 2
    poz2 = 2
    scor_minim = functionalitate4b(l_toti,poz1,poz2)
    assert(scor_minim == -1)

```

```

def test_func4c():
    '''l_toti = [[32,50],[434,44],[4,60]]
    poz1 = 1
    poz2 = 3
    multiplu = functionalitate4a(l_toti,poz1,poz2)
    assert(multiplu == "32 4 ")
    l_toti=[[32,52],[434,44],[4,64]]
    poz1 = 1
    poz2 = 3
    multiplu = functionalitate4a(l_toti,poz1,poz2)
    assert(multiplu == -2)
    poz1 = 7
    poz2 = 2
    multiplu = functionalitate4a(l_toti,poz1,poz2)
    assert(multiplu == -1)'''
    l_toti = [
        {"id_participant":32, "scor":50},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":60}
    ]
    poz1 = 1
    poz2 = 3
    multiplu = functionalitate4c(l_toti,poz1,poz2)
    assert(multiplu == "32 4 ")
    l_toti=[
        {"id_participant":32, "scor":52},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":64}
    ]
    poz1 = 1
    poz2 = 3
    multiplu = functionalitate4c(l_toti,poz1,poz2)
    assert(multiplu == -2)
    poz1 = 7
    poz2 = 2
    multiplu = functionalitate4c(l_toti,poz1,poz2)
    assert(multiplu == -1)

```

```

def test_func5a():
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":66}
    ]
    scor_dat = 6
    li_mea = functionalitate5a(l_toti, scor_dat)
    assert(li_mea == [2])
    scor_dat = 2
    li_mea = functionalitate5a(l_toti, scor_dat)
    assert(li_mea == -1)

def test_func5b():
    '''
    l_toti = [[32,54],[434,44],[4,65]]
    scor_dat = 45
    li_mea = functionalitate3c(l_toti, scor_dat)
    assert(li_mea == [2])
    scor_dat = 22
    li_mea = functionalitate3c(l_toti, scor_dat)
    assert(li_mea == [])'''
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":66}
    ]
    scor_dat = 45
    li_mea = functionalitate5b(l_toti, scor_dat)
    assert(li_mea == [2])
    scor_dat = 22
    li_mea = functionalitate5b(l_toti, scor_dat)
    assert(li_mea == -1)

def test_interval():
    poz1 = 2
    poz2 = 4
    interval = este_interval(poz1, poz2)
    assert(interval)

```



```

def test_func6():
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
    ]
    stack = [[1,32],[1,434]]
    functionalitate6(l_toti, stack)
    assert(l_toti==[{"id_participant":32, "scor":54}])
    assert(stack == [[1,32]])
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":0},
    ]
    stack = [[1,32],[2,434,52]]
    functionalitate6(l_toti, stack)
    assert(l_toti== [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":52},
    ])
    l_toti = [
        {"id_participant":32, "scor":0},
        {"id_participant":434, "scor":0},
        {"id_participant":4, "scor":66}
    ]
    stack = [[1,32],[1,434],[1,4],[3,1,2,[32,434],[23,77]]]
    functionalitate6(l_toti, stack)
    assert(l_toti==[
        {"id_participant":32, "scor":23},
        {"id_participant":434, "scor":77},
        {"id_participant":4, "scor":66}
    ])
    assert(stack==[[1,32],[1,434],[1,4]])
    l_toti = [
        {"id_participant":32, "scor":54},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":78}
    ]
    stack= [[1,32],[1,434],[1,4],[4,32,91]]
    functionalitate6(l_toti, stack)
    assert(l_toti==[
        {"id_participant":32, "scor":91},
        {"id_participant":434, "scor":44},
        {"id_participant":4, "scor":78}
    ])

```