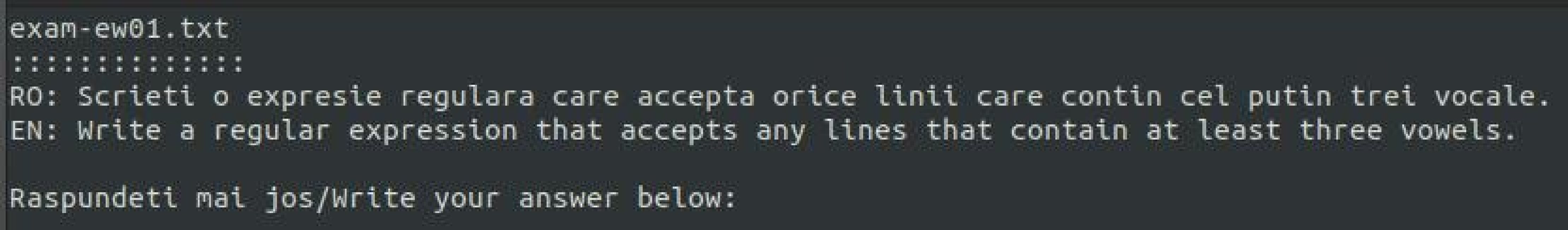
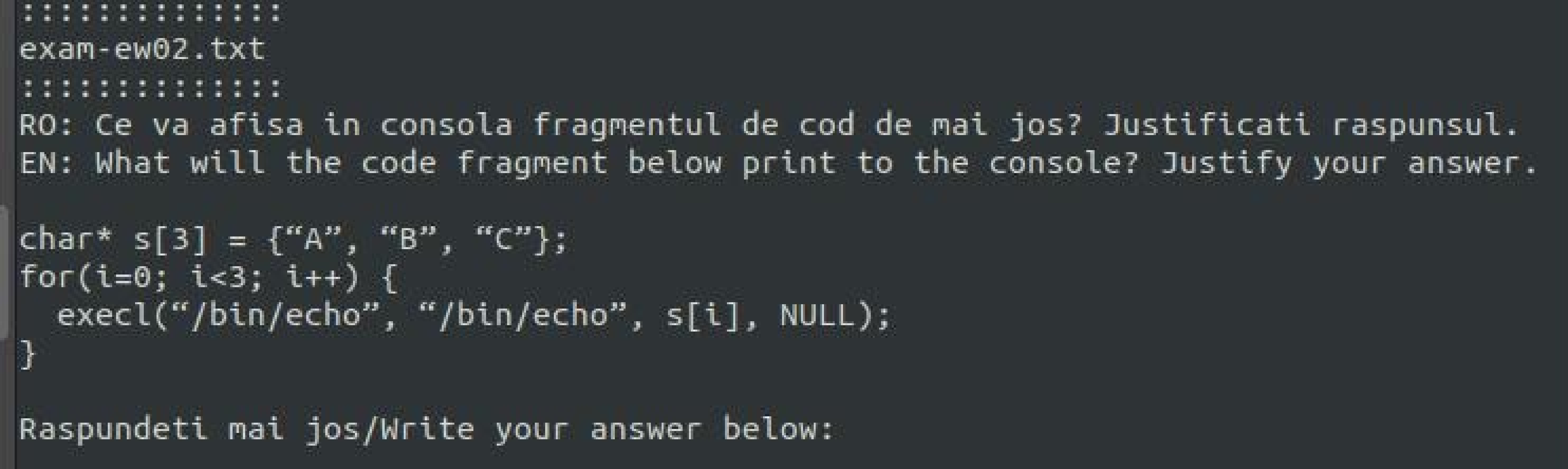
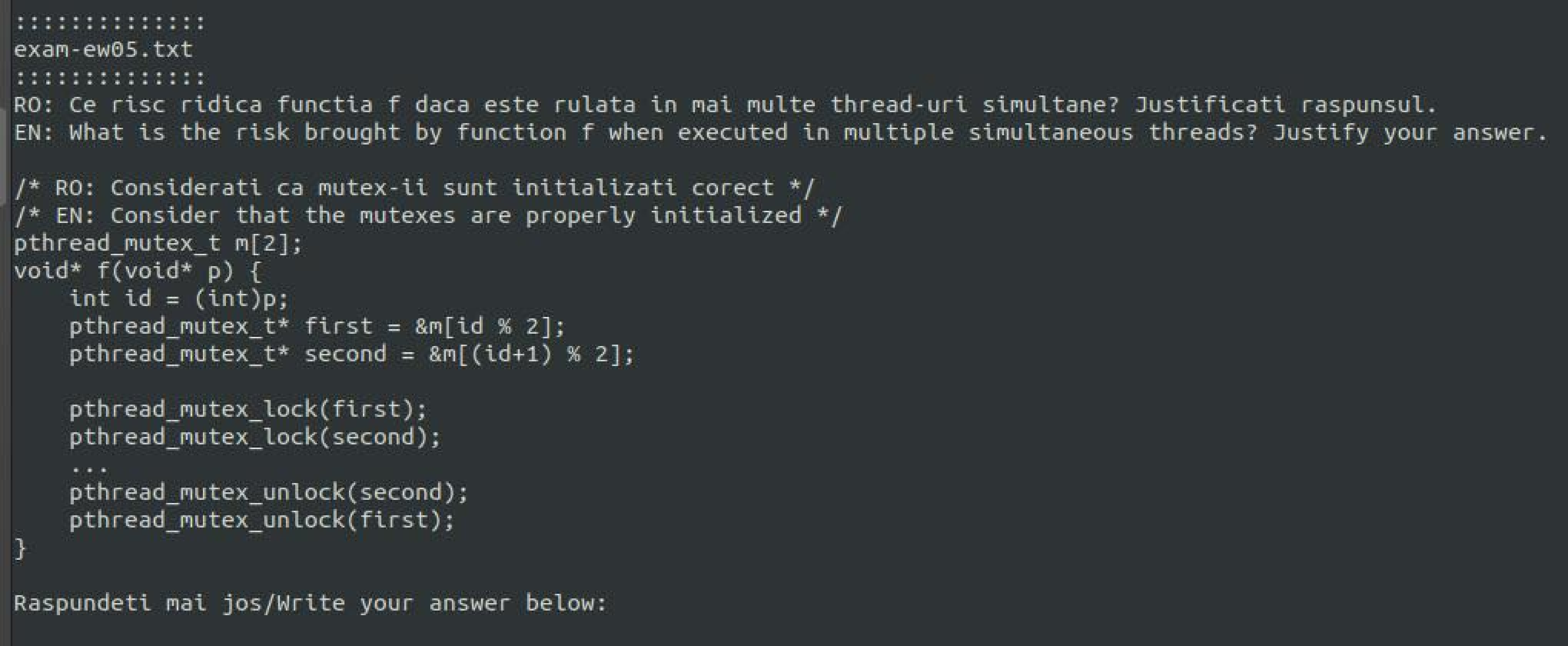


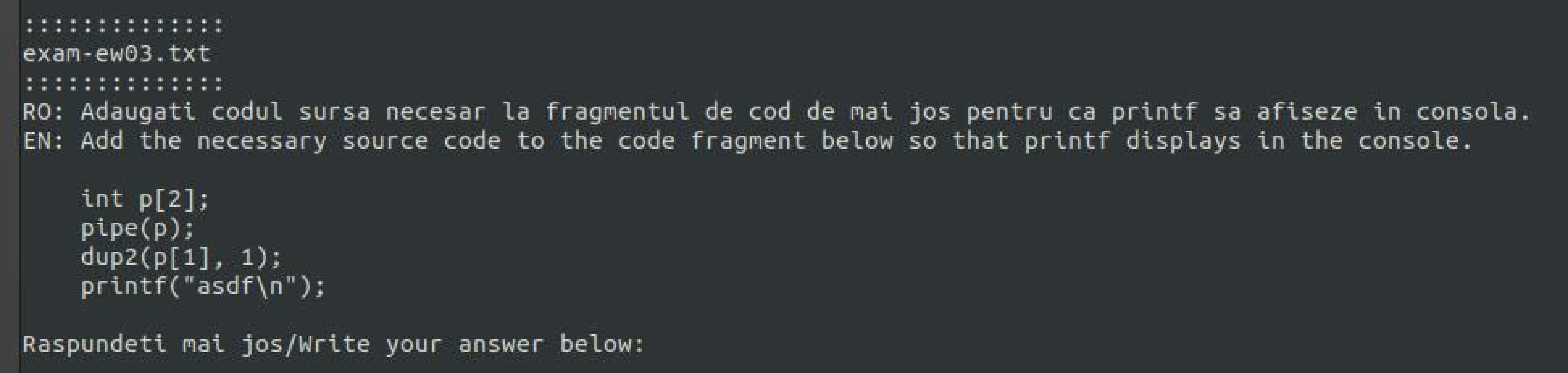
```
exam-ew01.txt
::::::::::::::
RO: Scrieti o expresie regulara care accepta orice linii care contin cel putin trei vocale.
EN: Write a regular expression that accepts any lines that contain at least three vowels.

Raspundeti mai jos/Write your answer below:
```



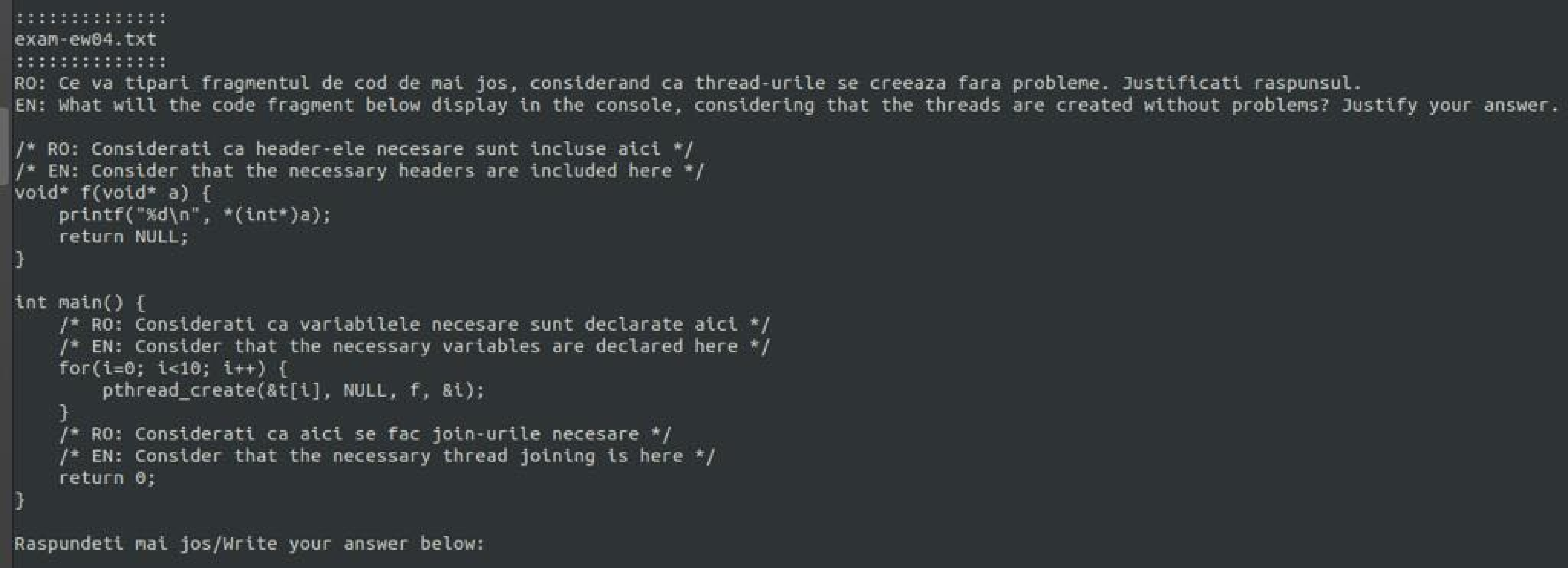
```
::::::::::::::
exam-ew02.txt
::::::::::::::
RO: Ce va afisa in consola fragmentul de cod de mai jos? Justificati raspunsul.
EN: What will the code fragment below print to the console? Justify your answer.

char* s[3] = {“A”, “B”, “C”};
for(i=0; i<3; i++) {
  execl(“/bin/echo”, “/bin/echo”, s[i], NULL);
}

Raspundeti mai jos/Write your answer below:
```



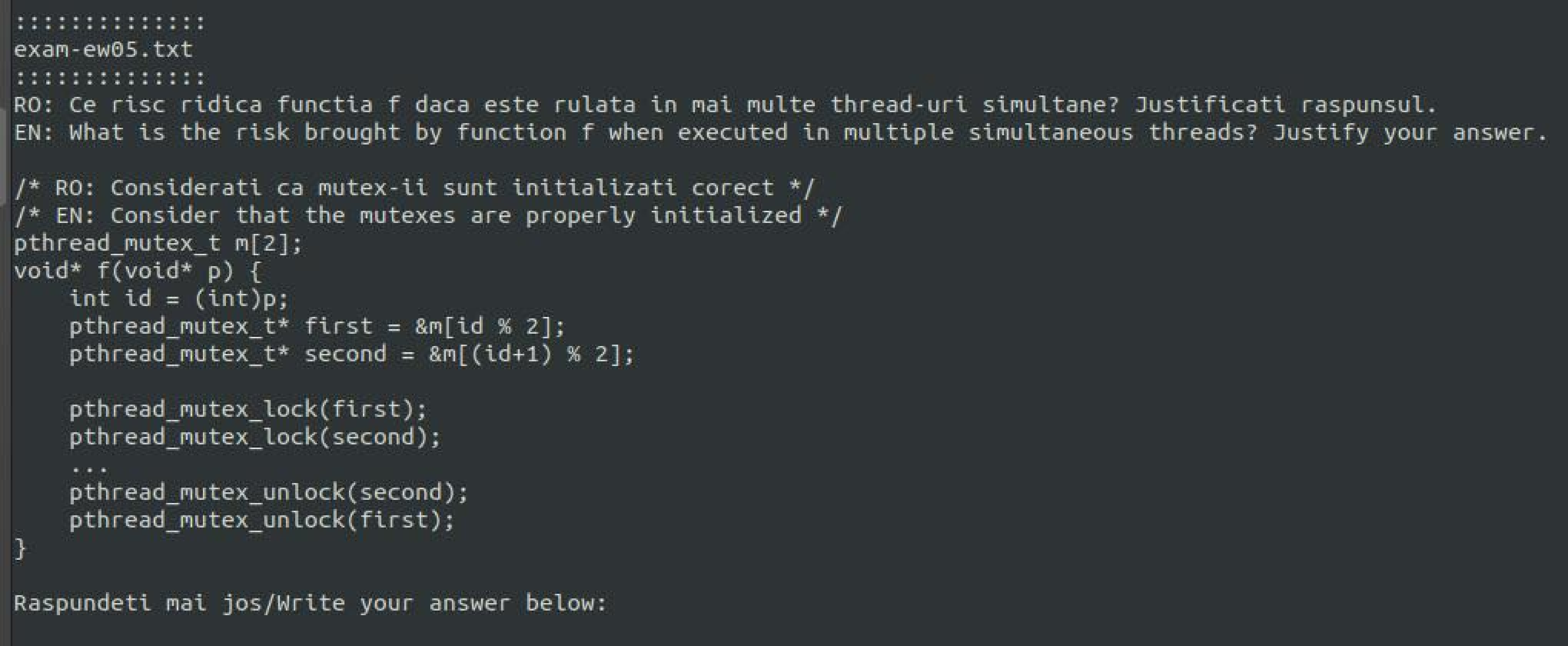
```
::::::::::::::
exam-ew05.txt
::::::::::::::
RO: Ce risc ridica functia f daca este rulata in mai multe thread-uri simultane? Justificati raspunsul.
EN: What is the risk brought by function f when executed in multiple simultaneous threads? Justify your answer.

/* RO: Considerati ca mutex-ii sunt initializati corect */
/* EN: Consider that the mutexes are properly initialized */
pthread_mutex_t m[2];
void* f(void* p) {
    int id = (int)p;
    pthread_mutex_t* first = &m[id % 2];
    pthread_mutex_t* second = &m[(id+1) % 2];

    pthread_mutex_lock(first);
    pthread_mutex_lock(second);
    ...
    pthread_mutex_unlock(second);
    pthread_mutex_unlock(first);
}

Raspundeti mai jos/Write your answer below:
```



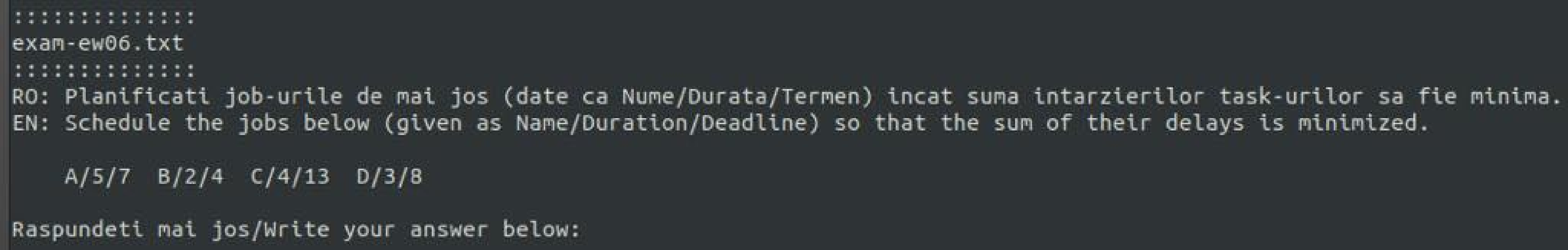
```
::::::::::::::
exam-ew03.txt
::::::::::::::
RO: Adaugati codul sursa necesar la fragmentul de cod de mai jos pentru ca printf sa afiseze in consola.
EN: Add the necessary source code to the code fragment below so that printf displays in the console.

    int p[2];
    pipe(p);
    dup2(p[1], 1);
    printf("asdf\n");

Raspundeti mai jos/Write your answer below:
```



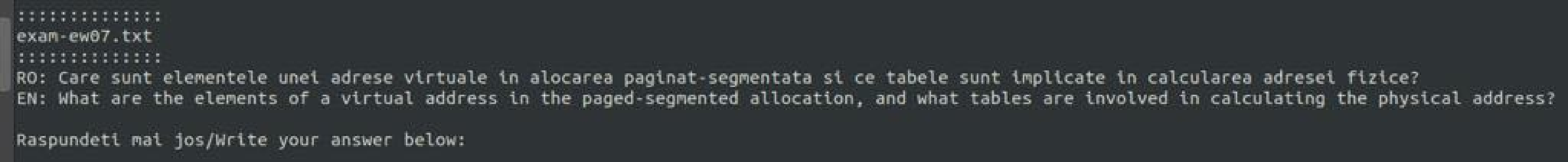
```
::::::::::::::
exam-ew04.txt
::::::::::::::
RO: Ce va tipari fragmentul de cod de mai jos, considerand ca thread-urile se creeaza fara probleme. Justificati raspunsul.
EN: What will the code fragment below display in the console, considering that the threads are created without problems? Justify your answer.

/* RO: Considerati ca header-ele necesare sunt incluse aici */
/* EN: Consider that the necessary headers are included here */
void* f(void* a) {
    printf("%d\n", *(int*)a);
    return NULL;
}

int main() {
    /* RO: Considerati ca variabilele necesare sunt declarate aici */
    /* EN: Consider that the necessary variables are declared here */
    for(i=0; i<10; i++) {
        pthread_create(&t[i], NULL, f, &i);
    }
    /* RO: Considerati ca aici se fac join-urile necesare */
    /* EN: Consider that the necessary thread joining is here */
    return 0;
}

Raspundeti mai jos/Write your answer below:
```



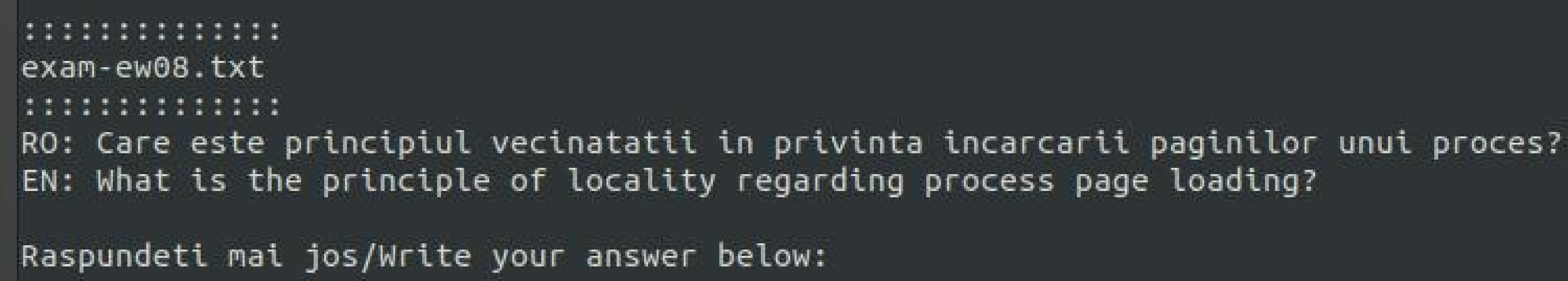
```
::::::::::::::
exam-ew05.txt
::::::::::::::
RO: Ce risc ridica functia f daca este rulata in mai multe thread-uri simultane? Justificati raspunsul.
EN: What is the risk brought by function f when executed in multiple simultaneous threads? Justify your answer.

/* RO: Considerati ca mutex-ii sunt initializati corect */
/* EN: Consider that the mutexes are properly initialized */
pthread_mutex_t m[2];
void* f(void* p) {
    int id = (int)p;
    pthread_mutex_t* first = &m[id % 2];
    pthread_mutex_t* second = &m[(id+1) % 2];

    pthread_mutex_lock(first);
    pthread_mutex_lock(second);
    ...
    pthread_mutex_unlock(second);
    pthread_mutex_unlock(first);
}

Raspundeti mai jos/Write your answer below:
```



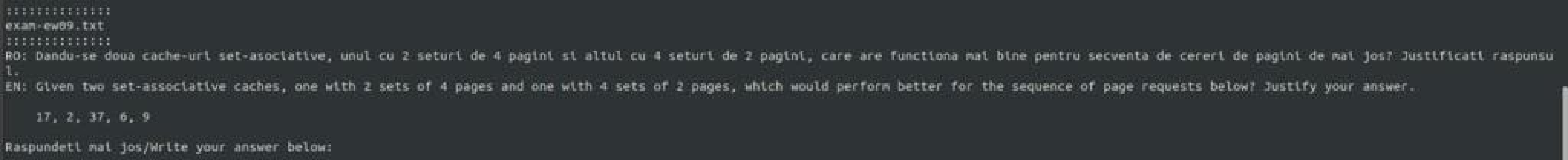
```
::::::::::::::
exam-ew06.txt
::::::::::::::
RO: Planificati job-urile de mai jos (date ca Nume/Durata/Termen) incat suma intarzierilor task-urilor sa fie minima.
EN: Schedule the jobs below (given as Name/Duration/Deadline) so that the sum of their delays is minimized.

    A/5/7  B/2/4  C/4/13  D/3/8

Raspundeti mai jos/Write your answer below:
```



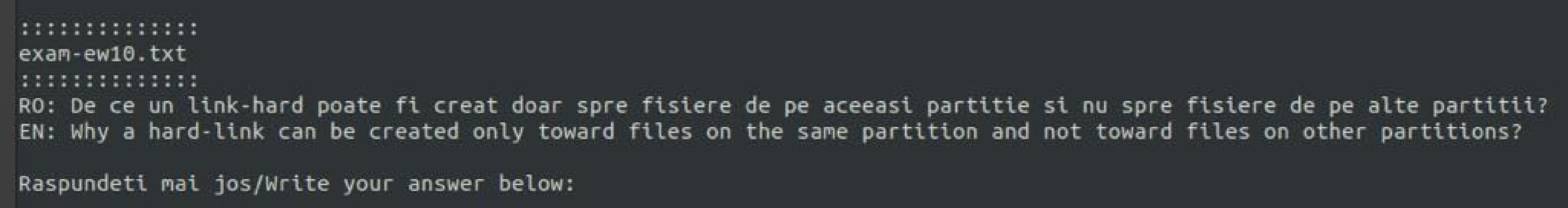
```
::::::::::::::
exam-ew07.txt
::::::::::::::
RO: Care sunt elementele unei adrese virtuale in alocarea paginat-segmentata si ce tabele sunt implicate in calcularea adresei fizice?
EN: What are the elements of a virtual address in the paged-segmented allocation, and what tables are involved in calculating the physical address?

Raspundeti mai jos/Write your answer below:
```



```
::::::::::::::
exam-ew08.txt
::::::::::::::
RO: Care este principiul vecinatatii in privinta incarcarii paginilor unui proces?
EN: What is the principle of locality regarding process page loading?

Raspundeti mai jos/Write your answer below:
```



```
::::::::::::::
exam-ew09.txt
::::::::::::::
RO: Dandu-se doua cache-uri set-asociative, unul cu 2 seturi de 4 pagini si altul cu 4 seturi de 2 pagini, care are functiona mai bine pentru secventa de cereri de pagini de mai jos? Justificati raspunsu
l.
EN: Given two set-associative caches, one with 2 sets of 4 pages and one with 4 sets of 2 pages, which would perform better for the sequence of page requests below? Justify your answer.

    17, 2, 37, 6, 9

Raspundeti mai jos/Write your answer below:
```



```
::::::::::::::
exam-ew10.txt
::::::::::::::
RO: De ce un link-hard poate fi creat doar spre fisiere de pe aceeasi partitie si nu spre fisiere de pe alte partitii?
EN: Why a hard-link can be created only toward files on the same partition and not toward files on other partitions?

Raspundeti mai jos/Write your answer below:
```