

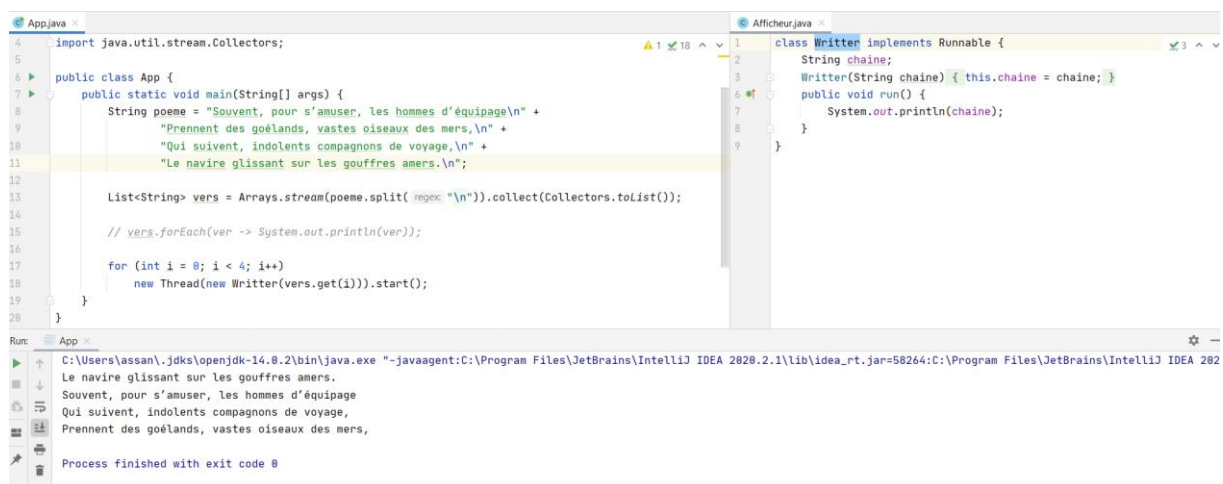
Exercice 1

Q1) Ecrire ce programme (mono tâche) qui satisfait ces spécifications.

```
public class App {  
    public static void main(String[] args) {  
        String poeme = "Souvent, pour s'amuser, les hommes d'équipage\n" +  
            "Prennent des goélands, vastes oiseaux des mers,\n" +  
            "Qui suivent, indolents compagnons de voyage,\n" +  
            "Le navire glissant sur les gouffres amers."  
        System.out.println(poeme);  
    }  
}
```

Q2) Quelles difficultés posent la répartition du travail en 4 activités autonomes

L'ordonnancement arbitraire : nous n'aurons pas les vers du poème dans l'ordre.



Q3) Proposer au moins deux solutions simples (avec ce qui a déjà été vu en cours) pour résoudre ces problèmes ; comparez ces solutions entre elles.

Il faut attendre que l'exécution du thread soit finie avec `join`.

Q4) Ecrire ce programme en Java en utilisant la solution que vous aurez jugée optimale.

```
for (int i = 0; i < 4; i++) {  
    Thread thread = new Thread(new Writer(vers.get(i)));  
    thread.start();  
    thread.join();  
}
```

Q5) Reprendre les questions précédentes en utilisant le langage C et les API et les concepts POSIX.

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <string.h>

pthread_t createTask(void *(*start_routine)(void *), void *arg)
{
    pthread_t id;
    int erreur;
    erreur = pthread_create(&id, NULL, start_routine, arg);
    if (erreur != 0)
    {
        perror("Echec creation de tâche Posix");
        exit(-1);
    }
    return id;
}

void * displaySentence(void * s)
{
    printf("%s\n", (char *)s);
}

int main()
{
    int i;
    pthread_t task;
    char poeme[] = "Souvent, pour s'amuser, les hommes d'équipage\nPrennent de  
s goélands, vastes oiseaux des mers,\nQui suivent, indolents compagnons de voy  
age,\nLe navire glissant sur les gouffres amers.";
    char delim[] = "\n";

    char *ptr = strtok(poeme, delim);

    while(ptr != NULL)
    {
        task = createTask(displaySentence, (void*)ptr);
        pthread_join( task, NULL);
        ptr = strtok(NULL, delim);
    }
    return 0;
}
```