Algortimos y programacion II:
Catedra Rosita
TP1.

Alumno: Juan Cruz Opizzi.
Padron: 99807.
DNI: 40226864.
Ayudante: Agustina Mendez.

Solución Paste: para el pase use la función getline sacada de la librería posix la cual me lee una linea de un archivo y me devuelve su longitud, entonces usando esta función leo ambos archivos al mismo tiempo y voi imprimiendo sus lineas una al lado de la otra, si alguna de los 2 me devuelve que tiene longitud igual a -1 (que es cuando llego al final del archivo) imprimo el error de que ambos archivos tienen distintos tamaños. Después de haber leído todas las lineas de ambos archivos, libero la memoria que pidió getline. La apertura y cierre de archivos, al igual que cuando verifico que la cantidad de archivos que recibo es la correcta la hago en una función main donde le paso los archivos abiertos a la función paste.
Esta función tiene un orden de O(n) siendo “n” el largo que tienen en común ambos archivos.

Solución More: use un main similar al del paste donde abro el archivo que me pasan y convierto a integer el numero que me pasan (que era un char por ingresarse por pantalla) utilizando la función atoi, luego también leo las lineas del archivo usando getline e imprimiendo las lineas del archivo hasta la cantidad de lineas que me pasaron por parámetro. Luego para imprimir una linea cada vez que se apreta enter use la función getchar la cual lee lo que te están pasando por pantalla y pregunto que si lo que me están pasando por pantalla es enter, imprimo una linea extra. Libero la memoria pedida por getline una vez recorrí todo el archivo hasta donde se quiso imprimir.
El orden de esta función es O(n) siendo “n” el largo del archivo hasta donde se quiso imprimir.

Solución DC: Para DC lo que hago es usar getline para leer la linea del archivo que me pasan, luego llamar a split pasandole como parámetro la linea del archivo, luego llamo a una función auxiliar que se fija si lo que me pasaron es un calculo o un numero, para ver si es un numero utilizo la función isdigit, y si la posición que veo tiene varios caracteres adentro, voy recorriendola preguntando si cada elemento es un numero, de no serlo devuelvo error. Luego con lo que me devuelve mi función auxiliar, si lo que me pasaron es un numero lo apilo en una pila que cree, si lo que me pasaron es un calculo, desapilo las 2 ultimas posiciones de la pila, aplico ese calculo a esos valores y al resultado lo apilo devuelta en la pila. Y asi hasta haber hecho todos los calculos que me pasaron. De haber mas de una linea en el archivo cada linea se le aplica los calculos por separado.

Esta función tiene un orden de O(nlogn) ya que tengo que recorrer todo el archivo y apilar y desapilar elementos en la pila.

Solución Join: la idea propuesta del Join es para poder pedir la cantidad exacta de memoria para la cadena a devolver es recorrer cada posición del arreglo, medir el largo de cada posición del arreglo (con strlen) e ir guardando en una variable el largo de cada posición del arreglo mas 1 espacio mas, el espacio extra es para poder meter en la cadena lo que en el arreglo eran los separadores. Una vez pedida la memoria para la cadena tengo que recorrer cada posición del arreglo y recorrer también cada posición por dentro e ir guardándolo en la cadena, después de haber terminado de leer una posición del arreglo tengo que guardar en la cadena el separador del arreglo y seguir así hasta haber terminado de recorrer el arreglo. Una vez que termine de recorrer el arreglo le asigno a la cadena como ultima posición el nulo (es decir '\0').
En esta función si bien recorro 2 veces el arreglo, por convencion de la cátedra esto sigue siendo O(n) siendo "n" el largo que tiene el arreglo y cada posición dentro del arreglo.

Solución Free_strv: Para esta función simplemente recorro todo el arreglo dinámico de cadenas hasta llegar al final, liberando el espacio de memoria pedido para cada posición, y una vez que termine de recorrer el arreglo, libero el espacio de memoria pedido para el arreglo en si.
Esta función es O(n) siendo "n" el largo del arreglo dinámico.

Solución split: para el split uso las funciones de strdup, strsep, strcpy y strlen. Copio con strdup la cadena que me pasan y la primer posición de la cadena, luego uso strcmp para recorrer la cadena hasta el separador y que en esa posición me inserte un "/0", luego copio hasta esa posición en el arreglo a devolver, piso como mi nuevo inicio donde esta ahora el /0 y me sigo moviendo hasta copiar toda la cadena en el arreglo.
Esta función tarda O(n) siendo n el largo del arreglo.