­­­­Facultad de Informática – Universidad Complutense  
1º curso de los Grados

Fundamentos de la programación – Grupos DG

Curso 2017–2018

Examen de junio de 2018

Tiempo disponible: 3 horas

Tenemos varias plantaciones de plátanos en la Aldea de San Nicolás en la isla de Gran Canaria. En todas ellas, los plataneros están situados en filas formando una cuadrícula perfecta. Para mejorar el sistema de recogida, hemos comprado un robot que recorre las plantaciones apuntando los plátanos listos para ser recogidos en cada árbol. Estos datos son trasladados a una matriz de valores enteros donde cada posición representa el número de plátanos listos para recoger en ese árbol. Posteriormente, se divide la plantación en parcelas cuadradas o rectangulares para ser recogidas por las cuadrillas.

En la plantación contamos con una serie de cuadrillas, cada una de las cuales puede recoger un número máximo de plátanos en una jornada. El número de cuadrillas disponibles varía según la temporada.

Vamos a realizar un programa informático que asigne a cada cuadrilla la parcela que debe ir a recoger. Para ello, debes implementar los siguientes módulos:

***Módulo Plantación (2 puntos)***

En este módulo se declara un tipo tPlantación con información sobre el identificador de la plantación y el número de plátanos a recoger de cada árbol de la plantación. El identificador de la plantación es una cadena de caracteres sin blancos. La información sobre los plátanos se almacena en una matriz de números enteros de tamaño num\_f \* num\_col con un tamaño máximo de 100 × 100.

Las operaciones que se realizan sobre la plantación son:

cargar. Carga desde un fichero la información de la plantación: identificador y matriz con el número de plátanos que hay que recoger en cada punto. El fichero comienza con una línea con el identificador y dos valores que indican el número de filas, num\_f, y el número de columnas, num\_col, de la matriz. En las num\_f filas siguientes se dan num\_col valores correspondientes a cada columna de la matriz.

El fichero tiene el siguiente formato:

Plantacion1 5 8

4 0 3 2 0 6 4 5

3 2 1 4 0 8 5 6

0 3 5 2 1 4 3 2

0 0 1 1 1 7 9 7

3 2 1 1 0 0 1 6

calcularPlatanosParcela. Dada una plantación, una fila x, una columna y, y un valor l, calcula el número de plátanos de la parcela de lado l y cuya esquina superior izquierda es la dada por la fila y la columna. Los datos de entrada cumplen: 0 <= x < num\_f y 0 <= y < num\_col y 0 < l < max(num\_f, num\_col). Si la parcela pedida sobrepasa los límites de la plantación se calculará el número de plátanos que se encuentran en la parte de la plantación.

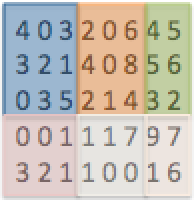
***Módulo Parcelas (3.5 puntos)***

Se declara un tipo tParcela con información sobre el identificador de la parcela que coincide con el identificador de la plantación en que se sitúa la parcela, la fila y columna donde se encuentra la esquina superior izquierda de la parcela, y el número de plátanos a recoger en esa parcela. Se declaran operaciones para escribir los datos de una parcela y para comparar dos parcelas. Una parcela es menor que otra si es menor su número de plátanos.

Se declara un tipo tListaParcelas como una lista estática de punteros a parcelas. El tamaño máximo será 1000.

Las operaciones sobre este tipo son:

crearParcelas. Dadas una plantación y la longitud de las parcelas, crea la lista de parcelas calculando los plátanos que se deben recoger en cada una de ellas. El número de elementos de la lista será el número de parcelas que se forman en la plantación con el lado dado. Si el número de árboles de las filas o columnas no es divisible entre el lado de la parcela considerado las parcelas más alejadas del (0,0) se formarán con los árboles disponibles y por lo tanto serán más pequeñas. La lista se mantendrá ordenada, en orden decreciente, por el número de plátanos. Por ejemplo, en una plantación de 8 × 5 con un lado de 3 se forman 6 parcelas:



mezclar. Recibe dos listas ordenadas de parcelas y devuelve una lista ordenada con los datos de todas las parcelas.

mostrarListaParcelas. Muestra la información de una lista de parcelas. La información se muestra separada por blancos. La implementación debe realizarse utilizando una **función recursiva**.

liberar. Libera la memoria dinámica de una lista de parcelas.

***Módulo Cuadrillas (3.5 puntos)***

Se declara un tipo estructurado tCuadrilla con datos sobre la identificación de la cuadrilla (cadena de caracteres sin blancos), el número máximo de plátanos que puede recoger en un día, y un puntero a la parcela asignada a la cuadrilla. Se declaran funciones para leer, escribir y comparar cuadrillas. Una cuadrilla es menor que otra si recoge menos plátanos.

Se declara un tipo tListaCuadrillas como una lista dinámica de cuadrillas. Esta lista no tiene orden.

Las operaciones sobre este tipo son:

inicializarListaCuadrilla. Reserva la memoria de la lista de cuadrillas.

crearListaCuadrilla. Crea una lista de cuadrillas a partir de los datos de un fichero llamado "Cuadrillas.txt" y la devuelve como resultado de la función. La lista se debe inicializar con tamaño el número de cuadrillas que se lee del fichero y no se modificará su tamaño durante toda la ejecución del programa. Si el fichero no se abre correctamente se devolverá una lista vacía.

El fichero comienza con el número de cuadrillas a cargar. A continuación hay tantas líneas como número de cuadrillas. En cada línea aparece el identificador de la cuadrilla y el número de plátanos que puede recoger.

El formato del fichero es el siguiente:

5

cuad1 30

cuad2 5

cuad3 20

cuad4 10

cuad5 40

asignarParcela. Las parcelas se asignan por orden, empezando por la que tiene más plátanos para recoger. Se asignan cuadrillas empezando por la primera de la lista hasta que la parcela está completamente recogida. A continuación se pasan a asignar las siguientes cuadrillas a la siguiente parcela. El dueño de la plantación sabe que con esta asignación algunas cuadrillas recogen menos plátanos de los que podrían recoger, pero no le importa, porque se dedican a otros trabajos pendientes en el tiempo que les sobra.

mostrarAsignaciones. Muestra la parcela que le ha correspondido a cada cuadrilla., con el formato que se muestra al final del enunciado.

Liberar. Libera la memoria dinámica de una lista de cuadrillas.

***Programa principal. (1 punto)***

El programa principal, comienza pidiendo el lado de las parcelas a construir. Supondremos que todos los valores introducidos por el usuario son correctos. A continuación pedirá el nombre del primer fichero de las plantaciones; cargará los datos en la matriz, y formará la lista con las parcelas de esta plantación. La lista de parcelas se debe mantener ordenada (decreciente) por el número de plátanos que hay que recoger. A continuación realiza el mismo proceso con la segunda plantación y procede a mezclar las dos listas. En este punto se pueden mostrar los datos de las parcelas para comprobar que la lista se ha formado correctamente.

Por último, crea la lista de las cuadrillas a partir de la información del fichero "Cuadrillas.txt". Asigna a cada cuadrilla una parcela de acuerdo con la asignación explicada anteriormente y cuando acaba la asignación se escribe la parcela a la que irá cada cuadrilla en un fichero de salida "Salida.txt".

Recuerda: El comando para que se muestre la memoria no liberada es

\_CrtSetDbgFlag(\_CRTDBG\_ALLOC\_MEM\_DF | \_CRTDBG\_LEAK\_CHECK\_DF);

y se coloca al comienzo del programa principal.

**Instrucciones de entrega**

* Pon al inicio de todos los archivo de código un comentario con tu nombre completo, el laboratorio y puesto que has utilizado.
* Cuando termines el examen, sube todos los archivos .h y.cpp con el código fuente por el sistema de entrega de ftp. No los comprimas.
* Se valorará la estructura, legibilidad y uso de los esquemas de recorrido y búsqueda vistos en clase. Recuerda que no se permite el uso de variables globales ni de instrucciones de salto, salvo un *return* como última instrucción de las funciones y *break* *si* utilizas *switch*.

**// archivo checkML.h**

#ifdef \_DEBUG

#define \_CRTDBG\_MAP\_ALLOC

#include <stdlib.h>

#include <crtdbg.h>

#ifndef DBG\_NEW

#define DBG\_NEW new ( \_NORMAL\_BLOCK , \_\_FILE\_\_ , \_\_LINE\_\_ )

#define new DBG\_NEW

#endif

#endif

*Listado de las parcelas ordenadas de los dos ficheros de entrada del ejemplo*

Plantacion1 0,3 27

Plantacion2 0,3 26

Plantacion1 0,6 25

Plantacion1 3,6 23

Plantacion2 0,6 22

Plantacion1 0,0 21

Plantacion2 0,0 15

Plantacion2 0,9 14

Plantacion1 3,3 10

Plantacion1 3,0 7

*Fichero Salida.txt* para los ficheros de entrada del ejemplo con longitud del lado de la parcela.

Parcela asignada a cada cuadrilla

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cuadrilla Capacidad Plantacion Parcela Platanos en parcela

cuad1 30 Plantacion1 0,3 27

cuad2 5 Plantacion2 0,3 26

cuad3 20 Plantacion2 0,3 26

cuad4 10 Plantacion2 0,3 26

cuad5 40 Plantacion1 0,6 25

cuad6 5 Plantacion1 3,6 23

cuad7 10 Plantacion1 3,6 23

cuad8 30 Plantacion1 3,6 23

cuad9 25 Plantacion2 0,6 22

cuad10 5 Plantacion1 0,0 21

cuad11 20 Plantacion1 0,0 21

cuad12 30 Plantacion2 0,0 15

cuad13 35 Plantacion2 0,9 14

cuad14 10 Plantacion1 3,3 10

cuad15 18 Plantacion1 3,0 7

*Ejemplo de ejecución*

