BUDDY ALLOCATOR WITH BITMAP

WHAT

We implemented a buddy allocator using a bitmap to save a part of memory and to keep track of the allocated and free blocks.
Bitmap is a buffer where each position of buffer's elements is the same of tree's nodes which represents the buddy allocator: the first element corresponds to the first node, the second one corresponds to the second node and so one...
Each element of bitmap can take 2 values:

- 0 if it's an allocated block
- 1 if it's a free block

HOW

At beginning we initialize a buddy allocator structure and a bitmap structure setting its bits to 1.
Test programs use two functions to allocate and free the memory: BuddyAllocator_malloc() and BuddyAllocator_free().

BuddyAllocator_malloc()
We use it to allocate a memory, and it returns a pointer to that memory. When you pass a size in input, it compute the level of the tree and it pass this to an auxiliary function called BuddyAllocator_GetBuddy() that searches in the right area of the bitmap a block setted to 1 and return a node. This node is written in the first part of the memory.

BuddyAllocator_free()
We use it to free a memory. It uses an auxiliary function called BuddyAllocator_ReleaseBuddy() that takes in input a node and it sets the block to 1.

Other auxiliary functions that we implemented:

set_successori	→	It sets to 0 the successors of a node recursively
set_padri	→	It sets to 0 the ancestors of a node recursively
set_successori_uno	→	It sets to 1 the successors of a node recursively
set_padri_uno	→	It sets to 1 the ancestors of a node recursively
Bitmap_setBit	→	It sets to 1 or 0 an input block
Bitmap_Bit	→	It checks the status of a block

HOW-TO-RUN
We implemented a Makefile to compile. For lauching tests you can use the command **make** and then **./test**<x> where x = 1,2,...,9 (the number of the tests).

AUTHORS:
Carlotta Guiso email: guiso.1801584@studenti.uniroma1.it
Marco Carfora email: carfora.1794568@studenti.uniroma1.it