TD – MALLOC – Partie 2 - A

1 – Commentez et expliquez le code ci-dessous

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

// verifier s'il ne manque pas des bibliothèques

// ?? - 1

#define ENTETE\_SIZE (ALIGN(sizeof(bloc\_entete)))

// ?? - 2

bloc\_entete\* ADR\_PREMIER\_BLOC=0;

// ?? - 3

typedef struct bloc\_entete{

size\_t taille ;

unsigned short libre :1 ;

} bloc\_entete;

// ?? - 4

void\* myalloc(size\_t t){

// ?? - 5

size\_t blk\_size = ALIGN(t+ENTETE\_SIZE);

// ?? - 6

if(ADR\_PREMIER\_BLOC == 0){

void\* bloc\_total=(void\*)sbrk(blk\_size);

void\* bloc\_utilisateur=(void\*)bloc\_total+ENTETE\_SIZE ;

bloc\_entete\* entete = (bloc\_entete\*)bloc\_total;

entete->taille=blk\_size;

entete->libre=0;

ADR\_PREMIER\_BLOC = (bloc\_entete\*)entete;

}

// ?? -7

else{ bloc\_entete\* current = (bloc\_entete\*)ADR\_PREMIER\_BLOC;

int alloc=0;

// ?? - 8

while(current<sbrk(0) && alloc==0){

// ?? - 9

if(current->libre==1 && current->taille >= blk\_size){

current->libre = 0;

alloc=1;

void\* bloc\_utilisateur=(void\*)current+ENTETE\_SIZE;

}

}

// ?? - 10

if(alloc==0){

void\* bloc\_total=(void\*)sbrk(blk\_size);

void\* bloc\_utilisateur=(void\*)bloc\_total+ENTETE\_SIZE ;

bloc\_entete\* entete = (bloc\_entete\*) bloc\_total;

entete->taille=blk\_size;

entete->libre=0;

}

}

// ?? -11

return bloc\_utilisateur;

}

// ?? - 12

void myfree(void\* ptr){

(bloc\_entete\*)ptr-ENTETE\_SIZE->libre=1;

}

// ?? - 13

void main(int argc, char\* argv[]){

// partie à compléter pour réserver 4 blocs et en libérer 2

}

|  |  |
| --- | --- |
| Commentaires |  |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |

2 – Complétez le main

void main(int argc, char\* argv[]){

// partie à compléter pour réserver 4 blocs et en libérer 2

}