



C- More malloc, free

TASK 0 – Leen Alsaleh

0. Float like a butterfly, sting like a bee

mandatory

Write a function that creates an array of chars, and initializes it with a specific char.

- Prototype: `char *create_array(unsigned int size, char c);`
- Returns `NULL` if size = 0
- Returns a pointer to the array, or `NULL` if it fails

```
julien@ubuntu:~/0x0a. malloc, free$ cat 0-main.c
#include "main.h"
#include <stdio.h>
#include <stdlib.h>

/**
 * simple_print_buffer - prints buffer in hexa
 * @buffer: the address of memory to print
 * @size: the size of the memory to print
 *
 * Return: Nothing.
 */
void simple_print_buffer(char *buffer, unsigned int size)
{
    unsigned int i;

    i = 0;
    while (i < size)
    {
        if (i % 10)
        {
            printf(" ");

```

Why should we determine the memory size?

Heap

لو حاولت احجز ذاكره بدون تحديد حجم :
`Int *arr;`
`arr=malloc(sizeof(int));`
نحجز عنصر واحد بس وين باقي الحجم ؟
ماحددنا كم عنصر احتاج
هنا الكومبايلر او المترجم بيحجز 4 بايت فقط
حجم int واحد
لكن ! لما أقول :
`arr=malloc(5*sizeof(int));`
حددنا حجم وحجزنا

Stack

`Int arr [5];`
المترجم الكومبايلر بيحسبها تلقائي
 $20=4*5$
تأخذ 4 بايت `Int`
وحجم العناصر 5
بيحجزها المساحه في
stack

Memory Allocation fails reasons

Out of memory

مافيه مساحه في الذاكره

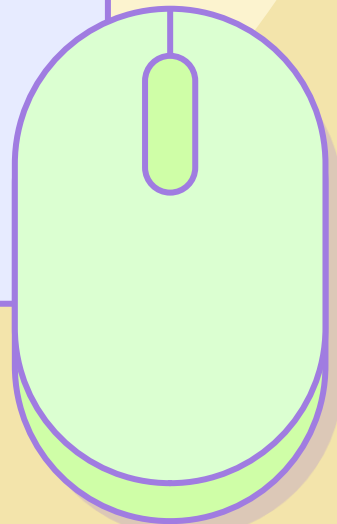
Memory fragmentation

تجزئة الذاكره
فيه مساحه لكن!
متفرقه وغير متجاوره
malloc
يحتاج مساحه متواصله متجاوره
(contiguous)

Dynamic Memory Allocation

يستخدم ال
heap
ليش؟

Enter



Why we use Dynamic Memory Allocation “heap”

01

Heap

يقدر يحتفظ بالبيانات حتى بعد ماتخلص
الداله

02

يقدر يحجز كميات كبيره من الذاكره
على حسب الحاجه

03

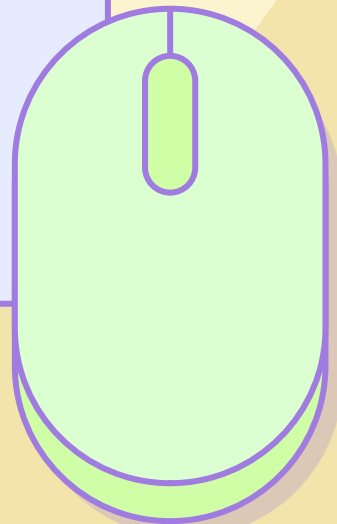
مرن اكثر لأنك تحدد الحجم وقت
التشغيل


Heap disadvantages (عيوب الheap):

memory Leak (تسريب في الذاكرة)
when ??

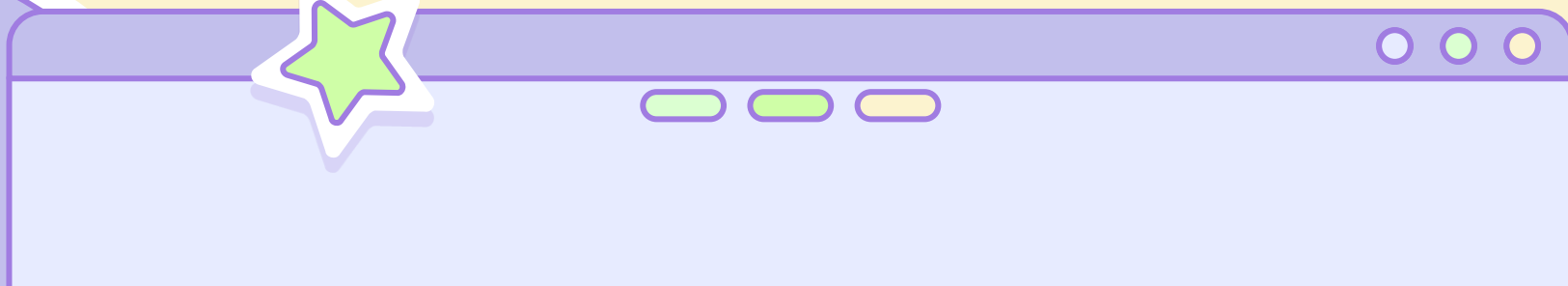
**if we don't do
free()**

Enter

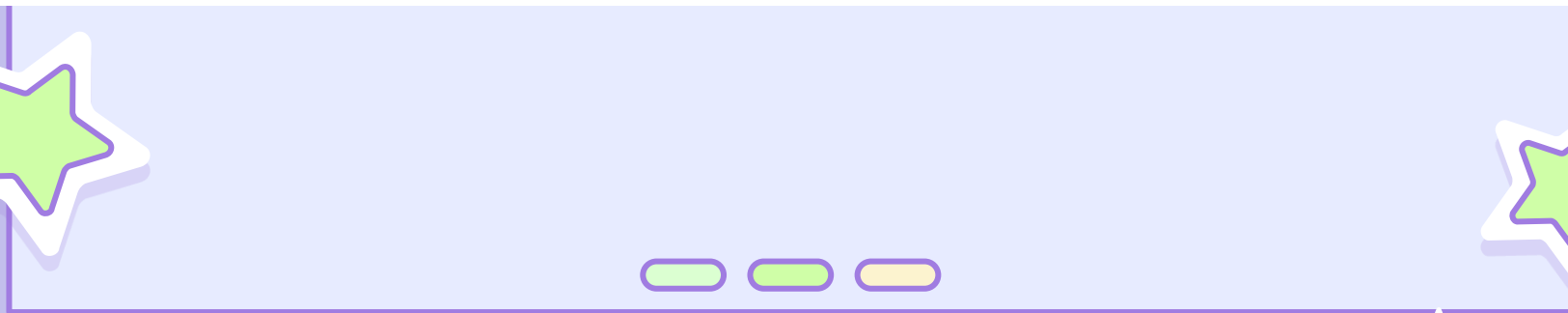




```
#include <stdlib.h>
#include "main.h"
/**
 * create_array - Creates an array of chars and initializes it
 * with a specific char.
 * @size: The size of the array.
 * @c: The character to initialize the array with.
 *
 * Return: Pointer to the array, or NULL if it fails or size is 0.
 */
char *create_array(unsigned int size, char c)
{
    char *arr;
    unsigned int i;
    if (size == 0)
        return (NULL);
    arr = malloc(size * sizeof(char));
    if (arr == NULL)
        return (NULL);
    for (i = 0; i < size; i++)
        arr[i] = c;
    return (arr);
}
```

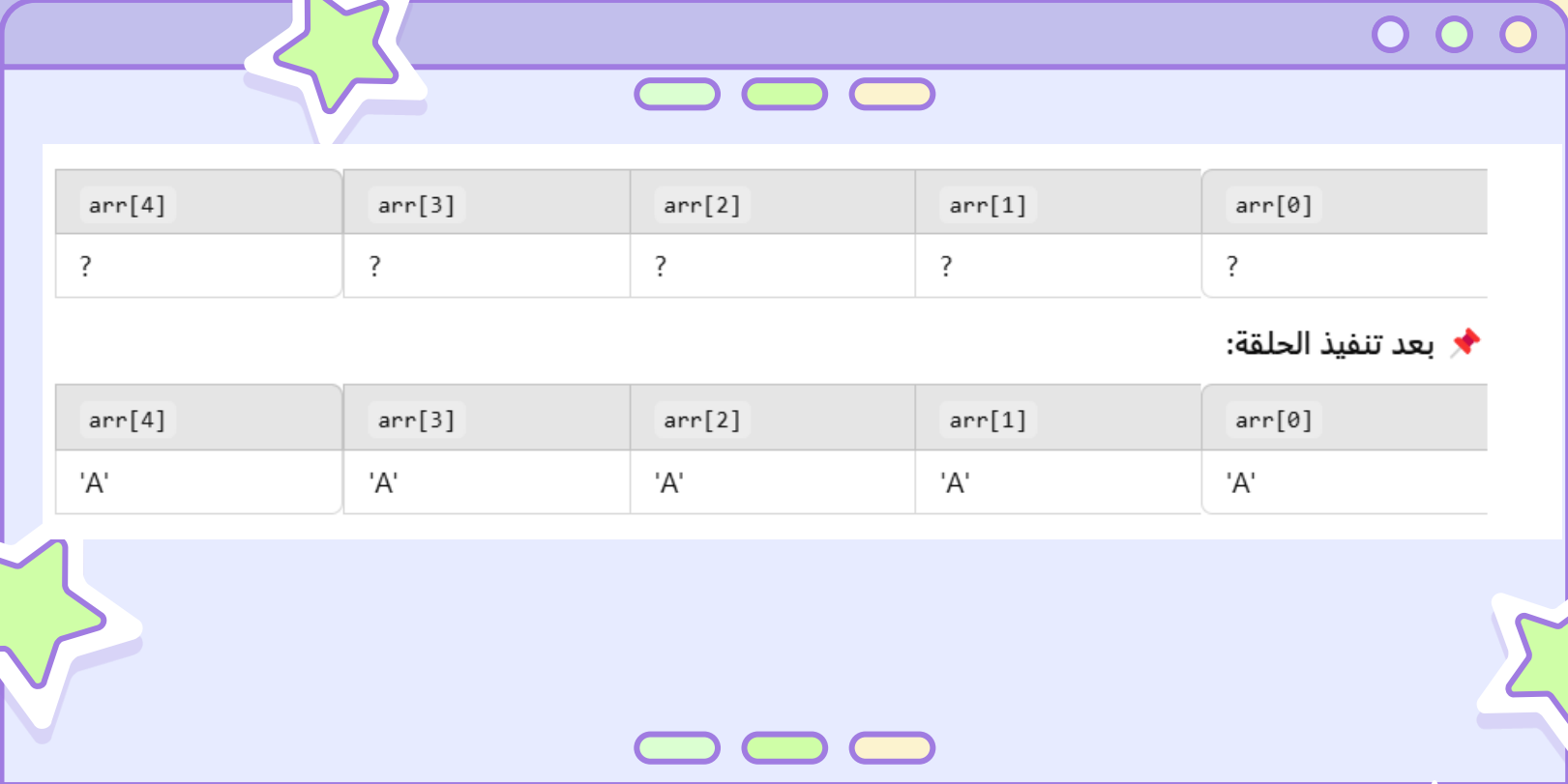



arr[4]	arr[3]	arr[2]	arr[1]	arr[0]
?	?	?	?	?





تنفيذ الجسم	تحقق الشرط؟ $i < \text{size}$	i
<code>arr[i] = c</code> يتم تعيين	$(5 > 0)$ ✓	0
<code>arr[1] = c</code> يتم تعيين	$(5 > 1)$ ✓	1
<code>arr[2] = c</code> يتم تعيين	$(5 > 2)$ ✓	2
<code>arr[3] = c</code> يتم تعيين	$(5 > 3)$ ✓	3
<code>arr[4] = c</code> يتم تعيين	$(5 > 4)$ ✓	4
لا يتم تنفيذ شيء	$(5 > 5)$ ✗ خطأ، نوقف التكرار	5



arr[4]	arr[3]	arr[2]	arr[1]	arr[0]
?	?	?	?	?

بعد تنفيذ الحلقة: 📌

arr[4]	arr[3]	arr[2]	arr[1]	arr[0]
'A'	'A'	'A'	'A'	'A'

(Address) العنوان (Value) المحتوى (Size) حجم العنصر

| 0x100 | 'A' | 1 بايت char

| 0x101 | 'A' | 1 بايت char

| 0x102 | 'A' | 1 بايت char

| 0x103 | 'A' | 1 بايت char

| 0x104 | 'A' | 1 بايت char



Thank you