

Поинтери и динамична памет

Задача 1: Изведете адреса на променливи от тип: int, char, double, float, bool.

Задача 2: Дефинирайте масив с N елемента на статичната и автоматичната памет. Изведете на екрана последователно адресите и срещу тях стойностите на всяка клетка в масива. Напишете функцията **void printAddress(int* array, unsigned size)**.

Задача 3: Напишете функцията **void swap(int* a, int* b)**, която разменя стойностите на две променливи по подадени адреси.

Задача 4: Напишете функцията **void readArray(int* array, unsigned size)**, която по подаден поинтер към масив с цели числа и размер size прочита от клавиатурата size на брой числа и ги записва в масива.

Задача 5: Напишете функцията **void printArray(int* array, unsigned size)**, която по подаден поинтер към масив с цели числа и размер size, изкарва на екрана size на брой числа от масива.

Динамична памет:

Задача 6: Напишете функциите

- int* alloc(unsigned size), която заделя динамично памет и връща поинтер към заделената памет.
- void free(int* array), която изтрива динамично заделената памет за масива

Задача 7: Въведете цяло положително число N. Заделете динамично масив с N елемента. Можете ли да използвате функциите от задача 4 и 5, за да запишете и изведете числата от и в масива?

Задача 8: За въведените числа в динамичния масив от задача 6, сортирайте го с любимия си сортиращ алгоритъм. Напишете функцията **void mySort(int* array, unsigned size, bool ascending)**, която сортира масива във възходящ ред, ако променливата ascending е равна на true. И в низходящ, ако е false.

Задача 9: Заделете динамично памет за N реални числа. Въведете ги. Изведете масив с N-1 числа, който представлява разликите на 2 поредни числа в първия масив.

Задача 10: Заделете динамично памет за N реални числа. Въведете ги. Заделете памет за точно толкова прости числа, колкото има в първоначално въведените. В новата памет запишете простите числа.