1- Write a program that dynamically allocates memory for a character array of user-defined size, takes user input for the string, checks if it is a palindrome, and displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم ل string ، يتحقق مما إذا كانت palindrome ويعرض النتيجة.

Input

```
Enter the size of the string: 4
Enter a string: mmmm
```

Output

```
The string is a palindrome.
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {
    char *str;
    int size, isPalindrome = 1;

    // Get string size from user
    printf("Enter the size of the string: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    str = (char*)malloc((size + 1) * sizeof(char));

    // Take user input for string
    printf("Enter a string: ");
    scanf("%s", str);

    // Check if the string is a palindrome
    for (int i = 0, j = size - 1; i < j; i++, j--) {
        if (str[i] != str[j]) {
            isPalindrome = 0;
            break;
        }
    }

    // Display the result
    if (isPalindrome) {
        printf("The string is a palindrome.\n");
    } else {
        printf("The string is not a palindrome.\n");
    }


    return 0;
}
```

2- Write a program that dynamically allocates memory for a float array of user-defined size, takes user input for each element (including negative numbers), calculates the average of positive numbers, and displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا float array ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر (بما في ذلك الأرقام السالبة)، ويحسب متوسط الأرقام الموجبة، ويعرض النتيجة.

Input

```
Enter the size of the array: 4
Enter 4 float numbers:
-4.5 6.3 5.3 -9
```

Output

```
Average of positive numbers in the array: 5.80
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    float *arr;
    int size, countPositive = 0;
    float sumPositive = 0;

    // Get array size from user
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    arr = (float*)malloc(size * sizeof(float));

    // Take user input for array elements
    printf("Enter %d float numbers:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%f", &arr[i]);
        if (arr[i] > 0) {
            countPositive++;
            sumPositive += arr[i];
        }
    }

    // Calculate average of positive numbers
    float averagePositive = (countPositive > 0) ? (sumPositive / countPositive) : 0;

    // Display the average of positive numbers
    printf("Average of positive numbers in the array: %.2f\n", averagePositive);


    return 0;
}
```

3- Write a program that dynamically allocates memory for a character array of user-defined size, takes user input for the string, converts it to uppercase, and displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا ل array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لstring ، ويحولها إلى أحرف كبيرة، ويعرض النتيجة.

Input

```
Enter the size of the string: 7
Enter a string: dynamic
```

Output

```
Uppercase string: DYNAMIC
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

int main() {
    char *str;
    int size;

    // Get string size from user
    printf("Enter the size of the string: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    str = (char*)malloc((size + 1) * sizeof(char));

    // Take user input for string
    printf("Enter a string: ");
    scanf("%s", str);

    // Convert string to uppercase
    for (int i = 0; i < size; i++) {
        str[i] = toupper(str[i]);
    }

    // Display the result
    printf("Uppercase string: %s\n", str);



    return 0;
}
```

4- Write a program that dynamically allocates memory for an integer array of user-defined size, takes user input for each element, counts the number of even numbers, and displays the count.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا array أعداد صحيحة ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويحسب عدد الأرقام الزوجية، ويعرض العدد.

Input

```
Enter the size of the array: 3
Enter 3 integer numbers:
1 3 5
```

Output

```
Number of even numbers in the array: 0
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    int *arr, size, countEven = 0;

    // Get array size from user
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    arr = (int*)malloc(size * sizeof(int));

    // Take user input for array elements
    printf("Enter %d integer numbers:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%d", &arr[i]);
        if (arr[i] % 2 == 0) {
            countEven++;
        }
    }

    // Display the count of even numbers
    printf("Number of even numbers in the array: %d\n", countEven);



    return 0;
}
```

5- Write a program that dynamically allocates memory for a character array of user-defined size, takes user input for the string, counts the occurrences of a specific character, and displays the count.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم ل string ، ويحسب تكرارات حرف معين، ويعرض العدد.

Input

```
Enter the size of the string: 8
Enter a string: elephant
Enter the character to count: e
```

Output

```
Number of occurrences of 'e' in the string: 2
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    char *str;
    int size, count = 0;
    char target;

    // Get string size from user
    printf("Enter the size of the string: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    str = (char*)malloc((size + 1) * sizeof(char));

    // Take user input for string
    printf("Enter a string: ");
    scanf("%s", str);

    // Take user input for the target character
    printf("Enter the character to count: ");
    scanf(" %c", &target);

    // Count occurrences of the target character
    for (int i = 0; i < size; i++) {
        if (str[i] == target) {
            count++;
        }
    }

    // Display the count
    printf("Number of occurrences of '%c' in the string: %d\n", target, count);

    // Free allocated memory
    free(str);

    return 0;
}
```

6- Write a program that dynamically allocates memory for a float array of user-defined size, takes user input for each element, searches for a specific element, and displays whether it is present or not.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا floating array ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويبحث عن عنصر معين، ويعرض ما إذا كان موجودًا أم لا.

Input

```
Enter the size of the array: 3
Enter 3 float numbers:
1.5 5.5 0.3
Enter the element to search: 5.5
```

Output

```
Element 5.50 is present in the array.
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    float *arr;
    int size;
    float target;
    int isFound = 0;

    // Get array size from user
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    arr = (float*)malloc(size * sizeof(float));

    // Take user input for array elements
    printf("Enter %d float numbers:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%f", &arr[i]);
    }

    // Take user input for the target element
    printf("Enter the element to search: ");
    scanf("%f", &target);

    // Search for the target element
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            isFound = 1;
            break;
        }
    }

    // Display the result
    if (isFound) {
        printf("Element %.2f is present in the array.\n", target);
    } else {
        printf("Element %.2f is not present in the array.\n", target);
    }


    return 0;
}
```

7- Write a program that dynamically allocates memory for a float array of user-defined size, takes user input for each element, calculates the square root of each element, and displays the results.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا floating array ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويحسب الجذر التربيعي لكل عنصر، ويعرض النتائج.

Input

```
Enter the size of the array: 4
Enter 4 float numbers:
16 64 36 64
```

Output

```
Square roots of array elements:
4.00 8.00 6.00 8.00
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main() {
    float *arr;
    int size;

    // Get array size from user
    printf("Enter the size of the array: ");
    scanf("%d", &size);

    // Dynamically allocate memory
    arr = (float*)malloc(size * sizeof(float));

    // Take user input for array elements
    printf("Enter %d float numbers:\n", size);
    for (int i = 0; i < size; i++) {
        scanf("%f", &arr[i]);
    }

    // Calculate square root of each element and display results
    printf("Square roots of array elements:\n");
    for (int i = 0; i < size; i++) {
        printf("%.2f ", sqrt(arr[i]));
    }
    printf("\n");



    return 0;
}
```

8- Write a program that dynamically allocates memory for an integer array of size 5 using malloc and checks if the allocation is successful.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا array أعداد صحيحة بالحجم 5 باستخدام malloc ويتحقق من نجاح التخصيص.

Output

```
Memory allocation for array successful.
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>

int main() {
    // Allocate memory for an integer array of size 5
    int *arr = (int*)malloc(5 * sizeof(int));

    // Check if memory allocation is successful
    if (arr != NULL) {
        printf("Memory allocation for array successful.\n");

    } else {
        printf("Memory allocation for array failed.\n");
    }

    return 0;
}
```

9- Create a program that dynamically allocates memory for a structure representing a student with name and age.

أنشئ برنامجًا يخصص الذاكرة ديناميكيًا structure تمثل الطالب بالاسم والعمر.

Output

```
Student: John, Age: 20
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Student {
    char name[50];
    int age;
};

int main() {
    struct Student *student = (struct Student*)malloc(sizeof(struct Student));

    if (student != NULL) {
        strcpy(student->name, "John");
        student->age = 20;

        printf("Student: %s, Age: %d\n", student->name, student->age);
        free(student);
    } else {
        printf("Memory allocation failed.\n");
    }

    return 0;
}
```

10- Create a program that dynamically allocates memory for a structure representing a book with title and page count. Initialize the structure and print its contents.

أنشئ برنامجًا يخصص الذاكرة ديناميكيًا structure تمثل كتابًا له عنوان وعدد الصفحات. تهيئة structure وطباعة محتوياته.

Output

```
Book: The C Programming Language, Pages: 300
```

Solution

```c
// www.gammal.tech

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Book {
    char title[50];
    int pageCount;
};

int main() {
    struct Book *book = (struct Book*)malloc(sizeof(struct Book));

    if (book != NULL) {
        strcpy(book->title, "The C Programming Language");
        book->pageCount = 300;

        printf("Book: %s, Pages: %d\n", book->title, book->pageCount);
        // No free operation
    } else {
        printf("Memory allocation failed.\n");
    }

    return 0;
}
```
