

1- Write a program that dynamically allocates memory for an integer array of size 5, initializes it, and then displays the elements.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array أعداد صحيحة بحجم 5، ثم يقوم بتهيئتها، ثم يعرض العناصر.

Output

```
Array elements: 1 2 3 4 5
```

Solution

```
// www.gammal.tech

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

int main() {
    int *arr = (int*)malloc(5 * sizeof(int));

    // Initialize the array
    for (int i = 0; i < 5; i++) {
        arr[i] = i + 1;
    }

    // Display the elements
    printf("Array elements: ");
    for (int i = 0; i < 5; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

2- Write a program that dynamically allocates memory for a string, reads the string from the user, and then displays it.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لسلسلة ما، ويقرأ String من المستخدم، ثم يعرضها.

Input

```
Enter a string: GammalTech
```

Output

```
You entered: GammalTech
```

Solution

```
// www.gammal.tech

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

int main() {
    char *str;
    str = (char*)malloc(100 * sizeof(char));

    // Read string from user
    printf("Enter a string: ");
    scanf("%s", str);

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

    return 0;
}
```

3- Write a program that dynamically allocates memory for an integer array of user-defined size, initializes it, calculates the sum, and then displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا ل array أعداد صحيحة ذات حجم محدد من قبل المستخدم، ثم يقوم بتهيئتها، ثم يحسب المجموع، ثم يعرض النتيجة.

Input

```
Enter the size of the array: 3
Enter 3 elements:
1 2 3
```

Output

```
Sum of array elements: 6
```

Solution

```
// www.gammal.tech

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

int main() {
    int *arr, n, sum = 0;

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

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

    // Initialize and calculate sum
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        sum += arr[i];
    }

    // Display the sum
    printf("Sum of array elements: %d\n", sum);

    return 0;
}
```

4- Write a program that dynamically allocates memory for two strings, takes user input for each, concatenates them, and displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لسلسلتين، ويأخذ مدخلات المستخدم لكل منهما، ويربطها، ويعرض النتيجة.

Input

```
Enter the first string: Gammal
Enter the second string: Tech
```

Output

```
Concatenated String: GammalTech
```

Solution

```
// www.gammal.tech

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

int main() {
    char *str1, *str2;

    // Allocate memory for strings
    str1 = (char*)malloc(50 * sizeof(char));
    str2 = (char*)malloc(50 * sizeof(char));

    // Take user input for strings
    printf("Enter the first string: ");
    scanf("%s", str1);
    printf("Enter the second string: ");
    scanf("%s", str2);

    // Concatenate strings
    strcat(str1, str2);

    // Display the concatenated string
    printf("Concatenated String: %s\n", str1);

    return 0;
}
```

5- Write a program that dynamically allocates memory for an array of two structures (representing books), takes user input for each book, and displays the information.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array من two structures (تمثلان الكتب)، ويأخذ مدخلات المستخدم لكل كتاب، ويعرض المعلومات.

Input

```
Enter title for book 1: Programing
Enter number of pages for book 1: 300
Enter title for book 2: Algorith
Enter number of pages for book 2: 250
```

Output

```
Book 1 Information:
Title: Programing
Pages: 300
Book 2 Information:
Title: Algorith
Pages: 250
```

Solution

```
// www.gammal.tech

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

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

int main() {
    struct Book *books;

    // Allocate memory for an array of structures
    books = (struct Book*)malloc(2 * sizeof(struct Book));

    // Take user input for each book
    for (int i = 0; i < 2; i++) {
        printf("Enter title for book %d: ", i + 1);
        scanf("%s", books[i].title);
        printf("Enter number of pages for book %d: ", i + 1);
        scanf("%d", &books[i].pages);
    }

    // Display book information
    for (int i = 0; i < 2; i++) {
        printf("Book %d Information:\n", i + 1);
        printf("Title: %s\n", books[i].title);
        printf("Pages: %d\n", books[i].pages);
    }

    return 0;
}
```

6- Write a program that dynamically allocates memory for a float array of user-defined size, takes user input for each element, calculates the average, and displays it.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ float array ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويحسب المتوسط، ويعرضه.

Input

```
Enter the size of the array: 3
Enter 3 float numbers:
1.2 0.5 3.6
```

Output

```
Average of array elements: 1.77
```

Solution

```
// www.gammal.tech

#include <stdio.h>
#include <stdlib.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 average
    float sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];
    }
    float average = sum / size;

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

    // Free allocated memory
    free(arr);

    return 0;
}
```

7- Write a program that dynamically allocates memory for a character array of user-defined size, takes user input for the string, reverses it, and displays the reversed string.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لـ string ، ويعكسها، ويعرض string المعكوسة.

Input

```
Enter the size of the string: 6
Enter a string: gammal
```

Output

```
Reversed String: lammag
```

Solution

```
// www.gammal.tech

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

    // Reverse the string
    int i, j;
    char temp;
    for (i = 0, j = size - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }

    // Display the reversed string
    printf("Reversed String: %s\n", str);

    return 0;
}
```

8- Write a program that dynamically allocates memory for an integer array of user-defined size, takes user input for each element, finds the maximum value, and displays it.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ `array` أعداد صحيحة ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويجد الحد الأقصى للقيمة، ويعرضها.

Input

```
Enter the size of the array: 4
Enter 4 integer numbers:
1 15 2 20
```

Output

```
Maximum value in the array: 20
```

Solution

```
// www.gammal.tech

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

int main() {
    int *arr, size;

    // 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]);
    }

    // Find the maximum value
    int max = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] > max) {
            max = arr[i];
        }
    }

    // Display the maximum value
    printf("Maximum value in the array: %d\n", max);

    // Free allocated memory
    free(arr);

    return 0;
}
```


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

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array أحرف ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لـ string ، ويحسب عدد حروف vowels ويعرض العدد.

Input

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

Output

```
Number of vowels in the string: 2
```

Solution

```
// www.gammal.tech

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

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

    // 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);

    // Count vowels in the string
    for (int i = 0; i < size; i++) {
        char ch = str[i];
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
            vowelCount++;
        }
    }

    // Display the count of vowels
    printf("Number of vowels in the string: %d\n", vowelCount);

    return 0;
}
```

10- Write a program that dynamically allocates memory for an integer array of user-defined size, takes user input for each element, calculates the sum of squares, and displays the result.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا لـ array أعداد صحيحة ذات حجم محدد من قبل المستخدم، ويأخذ مدخلات المستخدم لكل عنصر، ويحسب مجموع المربعات، ويعرض النتيجة.

Input

```
Enter the size of the array: 4
Enter 4 integer numbers:
1 2 3 4
```

Output

```
Sum of squares of array elements: 30
```

Solution

```
// www.gammal.tech

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

int main() {
    int *arr, size, sumSquares = 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]);
        sumSquares += arr[i] * arr[i];
    }

    // Display the sum of squares
    printf("Sum of squares of array elements: %d\n", sumSquares);

    // Free allocated memory
    free(arr);

    return 0;
}
```