

1- Create a C++ program that defines a binary tree using a struct named node. Each node in the tree contains an integer value and pointers to its left and right children. Implement a constructor in the struct to initialize the data value, and a function print() to print the data of a node. Finally, in the main() function, create a binary tree with the following structure:

قم بإنشاء برنامج ++C يحدد binary tree باستخدام struct تسمى node. تحتوي كل node في tree على قيمة عددية ومؤشرات إلى العناصر الفرعية اليمنى واليسرى. قم بتنفيذ مُنشئ في struct لتهيئة قيمة البيانات، ووظيفة print() لطباعة بيانات العقدة. أخيرًا، في الدالة main()، أنشئ binary tree بالبنية التالية:

Output



Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

struct node {
    int data;
    node* right, * left;
    node(int i) {
        data = i;
    }
    void print() {
        cout << data << endl;
    }
};

int main() {
    // Create nodes and set up the binary tree structure
    node* root = new node(8);
    root->right = new node(10);
    root->left = new node(20);
    root->left->right = new node(30);
    root->left->left = new node(40);

    // Print the binary tree structure
    cout << "\t\t\t" << root->data << endl;
    cout << "\t" << root->left->data;
    cout << "\t\t\t" << root->right->data << endl;
    cout << root->left->left->data << "\t\t";
    cout << root->left->right->data << endl;

    return 0;
}
```

2- Write a program to print numbers from 1 to N using recursion.

اكتب برنامجاً لطباعة الأرقام من 1 إلى N باستخدام recursion.

Input

Enter a number: 5

Output

Numbers from 1 to 5: 1 2 3 4 5

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

void printNumbers(int n) {
    if (n > 0) {
        printNumbers(n - 1);
        cout << n << " ";
    }
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    cout << "Numbers from 1 to " << num << ": ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

3- Write a program that takes an integer input 'num' from the user and prints the numbers from 'num' to 1 in decreasing order. using recursion.

اكتب برنامجًا يأخذ عددًا صحيحًا "num" من المستخدم ويطبّع الأرقام من "num" إلى 1 بترتيب تنازلي. using recursion.

Input

```
Enter a number: 5
```

Output

```
Numbers from 5 to 1: 5 4 3 2 1
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print numbers from n to 1 recursively
void printNumbers(int n) {
    if (n > 0) {
        cout << n << " ";
        printNumbers(n - 1);
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display the numbers from N to 1
    cout << "Numbers from " << num << " to 1: ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

4- Write a program that takes an integer input 'num' from the user and prints the numbers from 'num' to -2 in decreasing order. using recursion

اكتب برنامجًا يأخذ عددًا صحيحًا "num" من المستخدم ويطبع الأرقام من "num" إلى -2 بترتيب تنازلي. using recursion.

Input

```
Enter a number: 5
```

Output

```
Numbers from 5 to -2: 5 4 3 2 1 0 -1 -2
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print numbers from n to -2 recursively
void printNumbers(int n) {
    if (n > -3) {
        cout << n << " ";
        printNumbers(n - 1);
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display the numbers from N to -2
    cout << "Numbers from " << num << " to -2: ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

5- Write a program that takes an integer input 'num' from the user and prints the numbers from 'num' to -2 in decreasing order, excluding zero. using recursion

اكتب برنامجًا يأخذ عددًا صحيحًا "num" من المستخدم ويطبّع الأرقام من "num" إلى -2 بترتيب تنازلي، باستثناء الصفر. using recursion

Input

```
Enter a number: 3
```

Output

```
Numbers from 3 to -2 (excluding zero): 3 2 1 -1 -2
```

Solution

```
// www.gammal.tech

#include <iostream>
using namespace std;

// Function to print numbers from n to -2 excluding zero recursively
void printNumbers(int n) {
    if (n > -3) {
        if (n != 0)
            cout << n << " ";
        printNumbers(n - 1);
    }
}

int main() {
    int num;

    // Input: Get an integer from the user
    cout << "Enter a number: ";
    cin >> num;

    // Output: Display the numbers from N to -2 excluding zero
    cout << "Numbers from " << num << " to -2 (excluding zero): ";
    printNumbers(num);
    cout << endl;

    return 0;
}
```

6- 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;
}
```

7- 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 لـ 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;
}
```

8- 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;
}
```

9- 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;
}
```

10- Create a C++ program that defines a binary tree using a struct named node. Each node in the tree contains a string value and pointers to its left and right children. Implement a constructor in the struct to initialize the data value, and a function print() to print the data of a node. Finally, in the main() function, create a binary tree with the following structure and print it:

قم بإنشاء برنامج ++C يحدد binary tree باستخدام struct تسمى node. تحتوي كل node في tree على قيمة string ومؤشرات إلى العناصر الفرعية اليمنى واليسرى. قم بتنفيذ مُنشئ في struct لتهيئة قيمة البيانات، ووظيفة print() لطباعة بيانات العقدة. أخيرًا، في الدالة main()، أنشئ binary tree بالبنية التالية:

Output

```
      A
    B   C
  D   E
```

Solution

```
// www.gammal.tech

#include <iostream>
#include <string>
using namespace std;

struct node {
    string data;
    node* right, * left;
    node(string s) {
        data = s;
    }
    void print() {
        cout << data << endl;
    }
};

int main() {
    // Create nodes and set up the binary tree structure
    node* root = new node("A");
    root->right = new node("C");
    root->left = new node("B");
    root->left->right = new node("E");
    root->left->left = new node("D");

    // Print the binary tree structure
    cout << "\t\t\t" << root->data << endl;
    cout << "\t" << root->left->data;
    cout << "\t\t\t" << root->right->data << endl;
    cout << root->left->left->data << "\t\t";
    cout << root->left->right->data << endl;

    return 0;
}
```