

1. Defining and Using Functions in C

- هي مجموعة أوامر برمجية تؤدي مهمة معينة، ويمكنك استدعاؤها بدل ما C في (Function) الدالة: **التعريف**.
تكرر نفس الكود.
- **الاستخدام:**
 - كتابة تعريف الدالة.
 - أو داخل دوال أخرى (main) استدعاء الدالة في البرنامج الرئيسي.
- تسهل إعادة استخدام الكود، وتنظيم البرنامج، وتقليل الأخطاء: **الفائدة**.

2. Performing Arithmetic Operations with Numeric Variables in C

- العمليات الحسابية على المتغيرات الرقمية مثل:
 - **(+) الجمع**
 - **(-) الطرح**
 - **(*) الضرب**
 - **(/) القسمة**
 - **(%) الباقي من القسمة**
- (float, double) وأعداد عشرية (int) العمليات على أعداد صحيحة C تدعم.

What is Recursion?

- هو عندما تستدعي الدالة نفسها أثناء تنفيذها (Recursion) الاستدعاء الذاتي: **التعريف**.
- لازم يكون فيه شرط **إيقاف** علشان الدالة ما تشتغلش بلا نهاية: **الشرط المهم**.

4. Repeating Tasks with Recursion in C

- (loops) باستخدام الاستدعاء الذاتي، يمكنك تكرار مهمة معينة بدون استخدام حلقات.

5. Types of Recursion

هناك عدة أنواع للاستدعاء الذاتي

Direct Recursion: الدالة تستدعي نفسها مباشرة.

Indirect Recursion: دالة تستدعي دالة أخرى، والدالة الثانية ترجع وتستدعي الأولى.

Tail Recursion: الاستدعاء الذاتي يحدث في نهاية الدالة.

```
1 #include <stdio.h>
2
3 int greaterNumber(int a, int b) {
4     if (a > b)
5         return a;
6     else
7         return b;
8 }
9
10 int main() {
11     int num1, num2;
12     scanf("%d %d", &num1, &num2);
13     printf("The greater number is: %d\n", greaterNumber(num1, num2));
14     return 0;
15 }
16
```

Logs & others

Code::Blocks Search results

Checking for existence: C:\Users\HP\Documents\Task2 c\bin\Debug\Task2 c.exe
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files (x86)\Common Files\Oracle\Java\java8path;C:

"C:\Users\HP\Documents\Task2 c\bin\Debug\Task2 c.exe"
2
3
The greater number is: 3
Process returned 0 (0x0) execution time : 10.272 s
Press any key to continue.

```
1 #include <stdio.h>
2
3 const char* evenOrOdd(int num) {
4     if (num % 2 == 0)
5         return "even";
6     else
7         return "odd";
8 }
9
10 int main() {
11     int number;
12     scanf("%d", &number);
13     printf("%d is %s\n", number, evenOrOdd(number));
14     return 0;
15 }
16
```

s & others

Code::Blocks Search results

Checking for existence: C:\Users\HP\Documents\Task2 c\bin\Debug\Task2 c.exe

"C:\Users\HP\Documents\Task2 c\bin\Debug\Task2 c.exe"
25
25 is odd
Process returned 0 (0x0) execution time : 7.706 s
Press any key to continue.

The screenshot shows a code editor with a file named `main.c`. The code defines an `add` function and a `main` function. The `main` function takes two integers as input and prints their sum. A terminal window is open, showing the input `434` and the output `Sum = 466`. The terminal also displays the process return status and execution time.

```
main.c X
1  #include <stdio.h>
2
3  int add(int a, int b) {
4      return a + b;
5  }
6
7  int main() {
8      int num1, num2;
9      scanf("%d %d", &num1, &num2);
10     printf("Sum = %d\n", add(num1, num2));
11     return 0;
12 }
13
```

Terminal Output:

```
"C:\Users\HP\Documents\Task2 c\bin\Debug\Tas...
434
Sum = 466

Process returned 0 (0x0)   execution time : 5.554 s
Press any key to continue.
```

The screenshot shows a code editor with a file named `main.c`. The code defines a `factorial` function and a `main` function. The `factorial` function uses a recursive approach to calculate the factorial of a given number. The `main` function takes an integer as input and prints its factorial. A terminal window is open, showing the input `2345` and the output `Factorial = 0`. The terminal also displays the process return status and execution time.

```
main.c X
1  #include <stdio.h>
2
3  int factorial(int n) {
4      if (n == 0)
5          return 1;
6      else
7          return n * factorial(n - 1);
8  }
9
10 int main() {
11     int number;
12     scanf("%d", &number);
13     printf("Factorial = %d\n", factorial(number));
14     return 0;
15 }
16
```

Terminal Output:

```
Select "C:\Users\HP\Documents\Task2 c\bin\Debug...
2345
Factorial = 0

Process returned 0 (0x0)   execution time : 3.074 s
Press any key to continue.
```

```
c X
1  #include <stdio.h>
2
3  const char* checkNumber(int num) {
4      if (num > 0)
5          return "positive";
6      else if (num < 0)
7          return "negative";
8      else
9          return "zero";
10 }
11
12 int main() {
13     int number;
14     scanf("%d", &number);
15     printf("%d is %s\n", number, checkNumber(number));
16     return 0;
17 }
18
```

"C:\Users\HP\Documents\Task2 c\bin\Debug..."

234 is positive

Process returned 0 (0x0) execution time : 2.555 s

Press any key to continue.

Head Recursion: الاستدعاء الذاتي يحدث في بداية الدالة.