

CSE 113 Structured Programming Language

Recursion

Tasnim Zahan
Assistant Professor
Dept. of CSE
NEUB

Recursion

- The process in which a function calls itself is called recursion and the corresponding function is called as recursive function
- A recursive function calls itself until a *base condition* is true, and execution stops

```
void recurse()
{
    ... ..
    recurse();
    ... ..
}

int main()
{
    ... ..
    recurse();
    ... ..
}
```

The diagram illustrates the recursive process. It shows two function definitions: `void recurse()` and `int main()`. Inside `recurse()`, there is a call to `recurse();`. Inside `main()`, there is a call to `recurse();`. Arrows indicate the flow of execution: one arrow points from the `recurse();` line in `main()` to the `recurse()` function definition, and another arrow points from the `recurse();` line inside `recurse()` back to the `recurse()` function definition. A bracket on the right side of the diagram, labeled "recursive call", spans the vertical distance between these two call points.

```

#include <stdio.h>

int sum(int n) {
    if (n <= 1)
        return n;

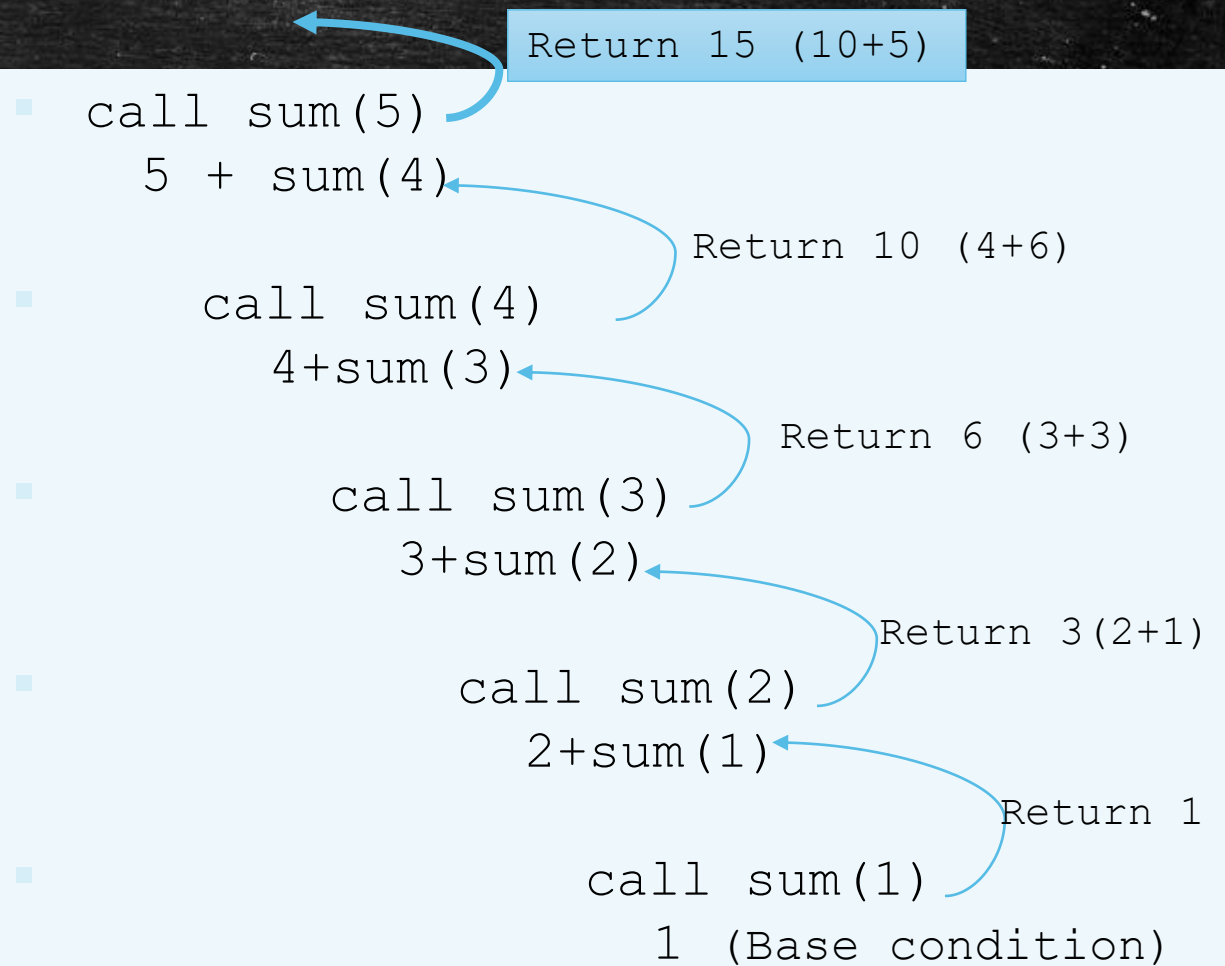
    return (n + sum(n-1));
}

int main() {
    int n, result;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    result = sum(n);
    printf("sum = %d", result);
    return 0;
}

```




```
#include <stdio.h>
```

Factorial with recursion

```
long factorial(long n) {  
    if(n<=1)  
        return 1;
```

```
    return (n*factorial(n-1));  
}
```

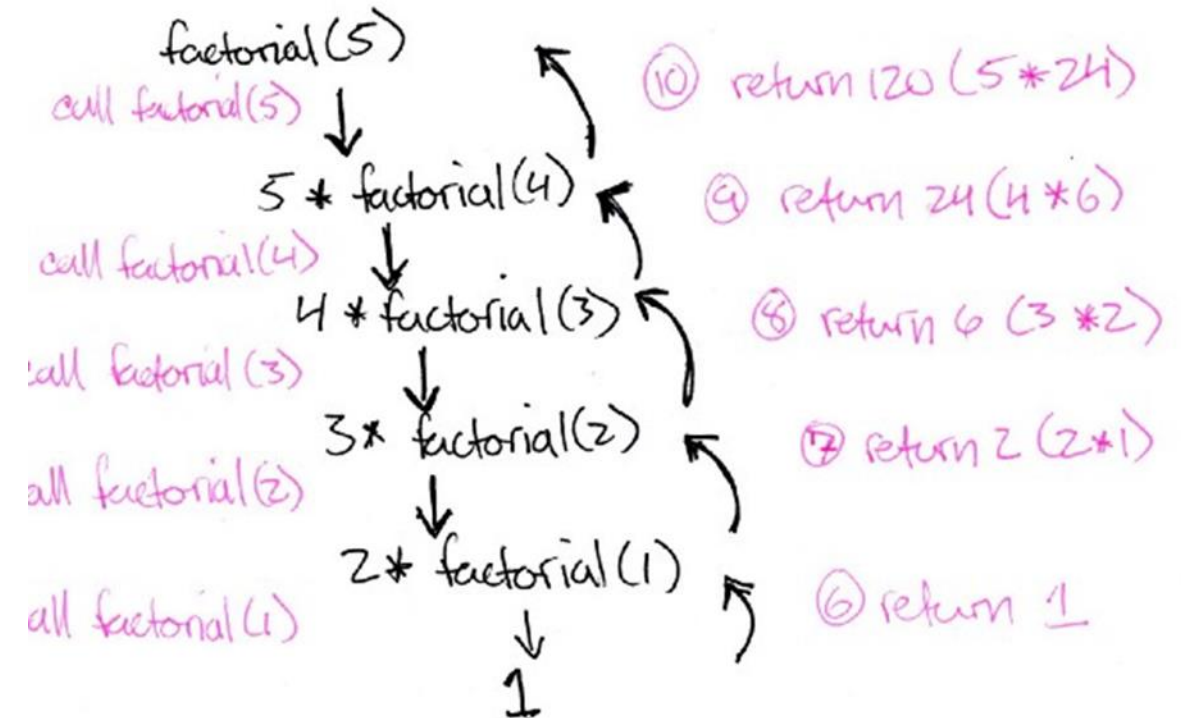
```
int main(){  
    long n, result;
```

```
    printf("Enter a positive integer: ");  
    scanf("%ld", &n);
```

```
    result = factorial(n);  
    printf("factorial of %ld is %ld", n, result);
```

```
    return 0;
```

```
}
```



String Reverse

```
#include <stdio.h>
char str[100];

void str_rev(int ind) {
    if(str[ind]=='\0')
        return;

    str_rev(ind+1);
    printf("%c",str[ind]);
    return;
}

int main() {
    printf("Enter a line of text: ");
    gets(str);

    str_rev(0);

    return 0;
}
```