ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Лекція 17.
Функції у мові Сі

```
void srand(int)
// функція з ім'ям srand, приймає ціле число, нічого не повертає

float sqrt(float)
// функція з ім'ям sqrt, приймає дійсне число типу float, повертає дійсне число типу float

int rand(void)
// функція з ім'ям rand, яка не приймає аргументів, повертає ціле число

double pow(double, double)
// функція з ім'ям pow, приймає два аргументи типу double, повертає дійсне число типу double
```

Підключення заголовочних файлів

Прототипи функцій

Функція main

Опис функцій користувача

ПРИКЛАД №1

Визначити максимальне з двох чисел

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

```c
int max_num(int, int);
```

```c
int main() {
int x = 0, y = 0;
int  m = 0;

scanf_s("%d %d", &x, &y);
m = max_num(x, y);
printf("max(%d,%d) = %d\n", x, y, m);
return 0;
}
```

```c
int max_num(int a, int b) {
int max = b;
if (a > b)
max = a;
return max;
}
```

ПРИКЛАД №2

Знайти площу окружності

```
#define _USE_MATH_DEFINES
#include <stdio.h>
#include <math.h>

double CircleSquare(double);

int main() {
double r;
scanf_s("%lf",&r);
printf("CircleSquare = %f\n", CircleSquare(r));
return 0;
}

double CircleSquare(double a) {
printf("PI = %f\n", M_PI);
return M_PI*a*a;
}
```

ПРИКЛАД №3

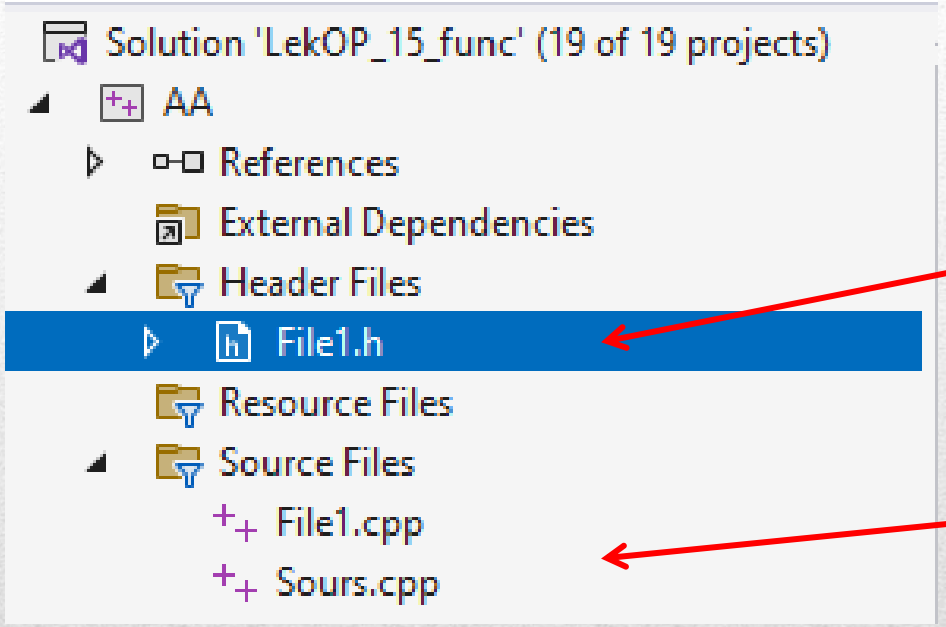
Описати функцію Fact (N) дійсного типу, яка обчислює значення факторіала

$$N! = 1 * 2 * ... * N$$

(N> 0 - параметр цілого типу; дійсний повертається, значення використовується для того, щоб уникнути цілочисельного переповнення при великих значеннях N). За допомогою цієї функції знайти факторіали п'яти заданих цілих чисел.

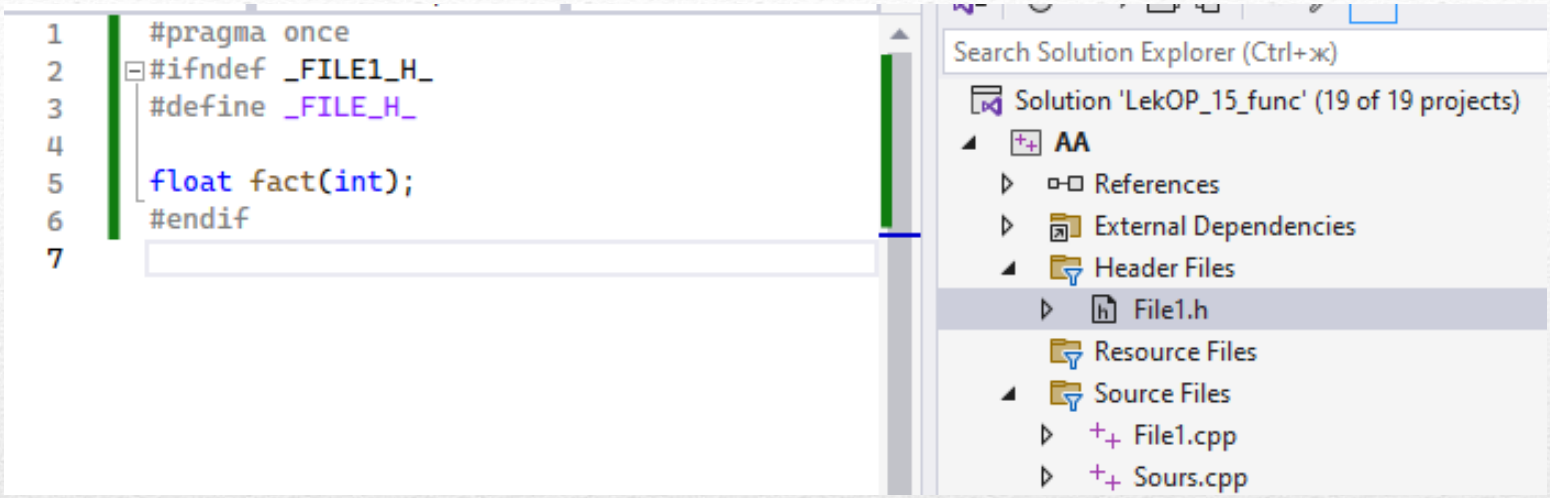
```c
#include <stdio.h>

float fact(int n) {
int temp = 1, i;
for (i = 1; i <= n; ++i) temp *= i;
return temp;
}

int main(void)
{
int i, n;
for (i = 1; i <= 5; ++i) {
printf("N:");
scanf_s("%i", &n);
printf("fact: %g\n", fact(n));
}
return 0;
}
```

Створення бібліотеки



Для цього потрібно створити два файли - один з розширенням **.h** і помістити туди прототипи функції, а інший з розширенням **.cpp** і помістити туди функцію.

.h файл необхідно створювати в папці ***Header Files***, а файли коду в папці ***Soursce Files***.
Нехай файли називаються File1.h і File1.cpp

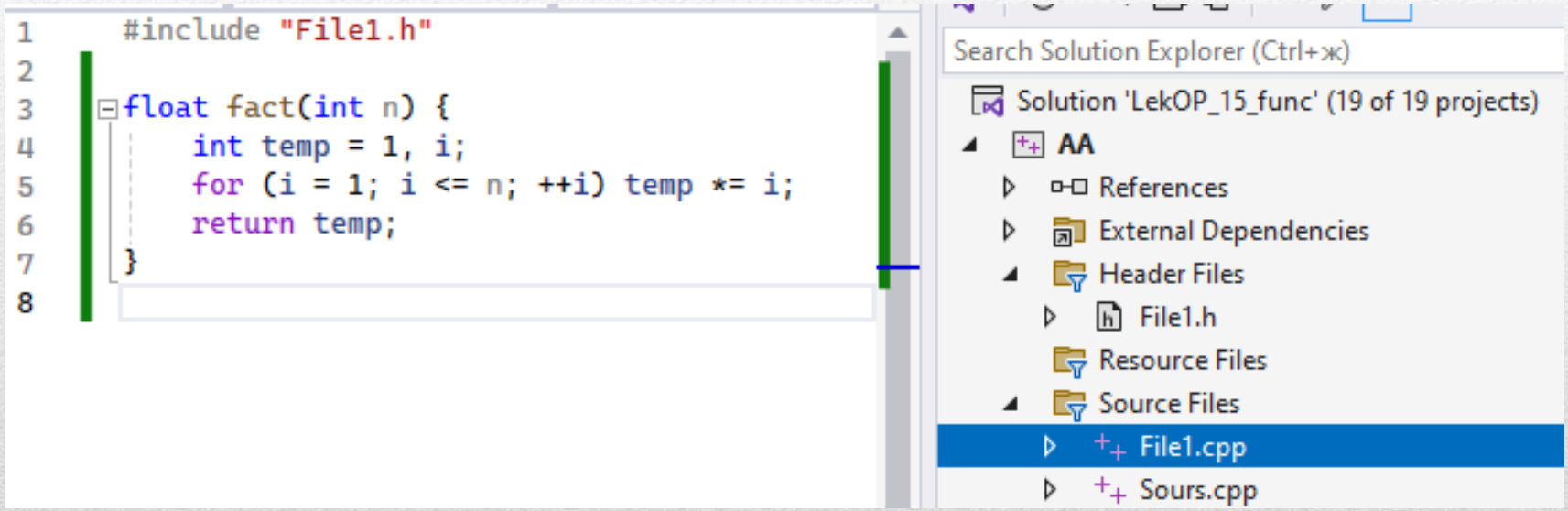


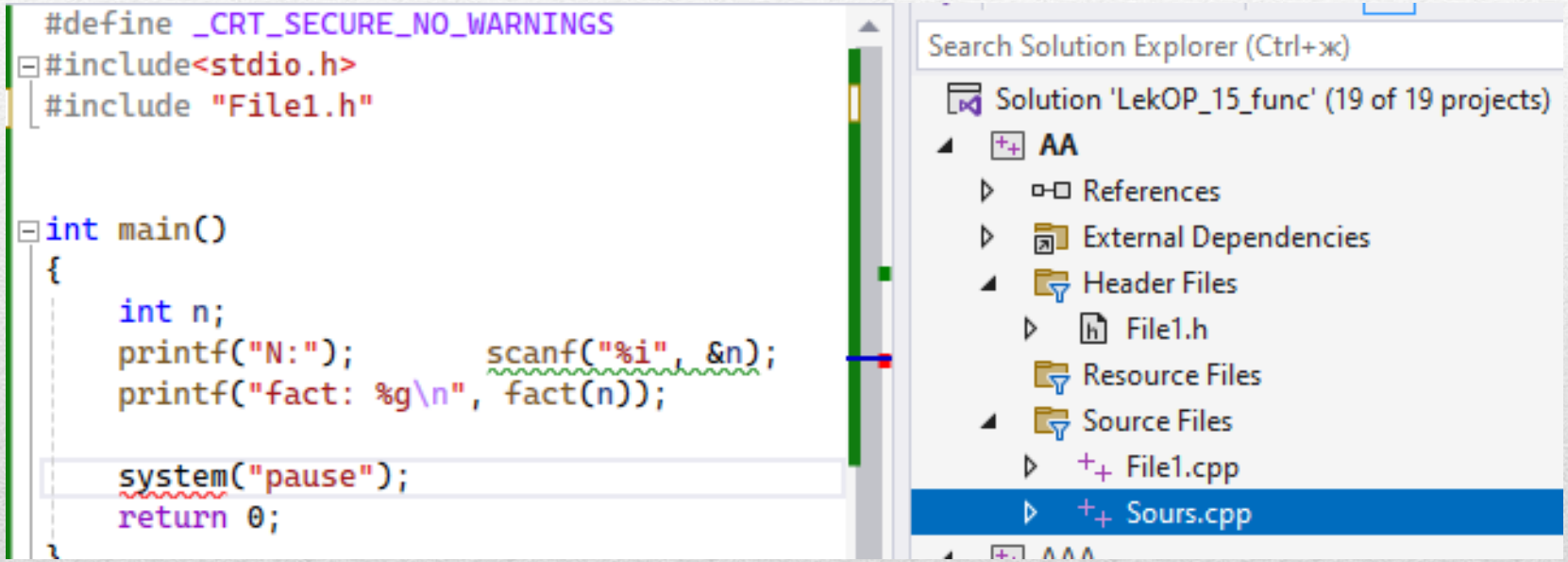
Дана препроцесорна обгортка запобігає багаторазовому включенню заголовних файлів.

Препроцесорні директиви обробляються до етапу компіляції, програмою-препроцесором. Яка, в свою чергу не допускає багаторазового визначення однієї і тієї ж функції.

Директива #ifndef перевіряє, чи визначено ім'я _FILE1_H_, якщо ні, то управління передається директиві #define.

Якщо ж ім'я _FILE1_H_ вже визначено, управління передається директиві #endif.


#include "File1.h"
float fact(int n) {
int temp = 1, i;
for (i = 1; i <= n; ++i) temp *= i;
return temp;
}
Search Solution Explorer (Ctrl+ж)
Solution 'LekOP_15_func' (19 of 19 projects)
AA
References
External Dependencies
Header Files
File1.h
Resource Files
Source Files
File1.cpp
Sours.cpp


#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include "File1.h"
int main()
{
int n;
printf("N:"); scanf("%i", &n);
printf("fact: %g\n", fact(n));
system("pause");
return 0;
Search Solution Explorer (Ctrl+ж)
Solution 'LekOP_15_func' (19 of 19 projects)
AA
References
External Dependencies
Header Files
File1.h
Resource Files
Source Files
File1.cpp
Sours.cpp

ПРИКЛАД №4

Описати функцію AddLeftDigit (D, K), що додає до цілого позитивного числа K зліва цифру D (D - вхідний параметр цілого типу, що лежить в діапазоні 1-9, K - параметр цілого типу, який є вхідним і вихідним). За допомогою цієї процедури послідовно додати до цього числа K зліва дані цифри D_1 і D_2, виводячи результат кожного додавання.

```c
#include <stdio.h>

int  addleftdigit(int d, int k) {
    int temp = 10;
    while (k > temp) temp *= 10;
        k += d * temp;
return k;
}

int main() {
    int i, k;
    printf("K:");
    scanf_s("%i", &k);

    for (i = 1; i <= 2; ++i) {
        int d;
        printf("D:");
        scanf_s("%i", &d);

        k=addleftdigit(d, k);
        printf("K: %i\n", k);
    }

return 0;
}
```

ПРИКЛАД №5

?

Описати функцію SumRange (A, B) цілого типу, яка знаходить суму всіх цілих чисел в масиві від A до B включно (A і B - цілі). Якщо A> B, то функція повертає 0.
За допомогою цієї функції знайти суми чисел від A до B і від B до C.