МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Основы алгоритмизации и программирования»

Лабораторная работа №13

Рекурсия

Выполнил:

студент гр. ПИбд-12

Дозорова Алена

Ульяновск, 2021 г

Задание:

1. Собрать и отладить (т.е. заставить работать) код рекурсивного вычисления факториала. И выполнить трассировку его для n=5 используя встроенный отладчик VS
2. Используя простейшие рекурсивные функции с предыдущего слайда в качестве вдохновения и основы, сделайте собственные рекурсивные функции (f1(), f2(), f3()), которые выводят в консоль последовательность чисел:
   1. **Задача 2.1.**

Вызов функции: f1(11)

Формируемый вывод: 11 9 7 5 3 1

* 1. **Задача 2.2.**

Вызов функции: f2(11)

Формируемый вывод: 1 3 5 7 9 11

* 1. **Задача 2.3.**

Вызов функции: f3(11)

Формируемый вывод: 11 9 7 5 3 1 3 5 7 9 11

1. Выполнить трассировку только что созданных функций
   1. Задача 3.1.

Выполнить трассировку для n=7 используя встроенный отладчик VS. Выполнить трассировку по очереди для всех функций f1(), f2(), f3()

* 1. Задача 3.2.

Выполнить трассировку для n=7 используя встроенный бумагу и ручку/карандаш. Выполнить трассировку по очереди для всех функций f1(), f2(), f3()

1. Нужно выполнить трассировку представленного кода в отладчике VS.
2. Нужно выполнить трассировку представленного кода в отладчике VS.
3. Нужно выполнить трассировку представленного кода в отладчике VS.
4. Нужно выполнить трассировку представленного кода в отладчике VS.
5. В игру, реализованную в лаб работах 8, 9 и т.д. добавить «руку Мидаса» - прикосновение к стене превращают всю стену в набор золотых элементов. В золото превращаются ВСЕ ЭЛЕМЕНТЫ связанные друг с другом – в не зависимости от конфигурации стены.
   1. Выполнить трассировку в отладчике VS заливки небольшой стены при помощи добавленной функции
6. Реализовать бинарный поиск в отсортированном массиве. Необходимо реализовать его двумя способами – итерационно и рекурсивно.
7. Реализовать от двух до четырех рекурсивных функций не упомянутых в лекции и/или в лабораторной работе.
8. Для двух рисунков, реализованных самостоятельно в рамках лаб работы №4 (тема «Фракталы») выполнить трассировку – в отладчике VS и вручную.

Трассировка:

|  |  |
| --- | --- |
| n | res |
| 5 | 120 |
| 4 | 24 |
| 3 | 6 |
| 2 | 2 |
| 1 | 1 |

f1(11)

|  |  |
| --- | --- |
| Вывел | Вызвал |
| 11 | 9 |
| 9 | 7 |
| 7 | 5 |
| 5 | 3 |
| 3 | 1 |
| 1 |  |

f2(11)

|  |  |
| --- | --- |
| Вызвал | Вывел (снизу вверх) |
| 11 | 9 11 |
| 9 | 7 |
| 7 | 5 |
| 5 | 4 |
| 3 | 3 |
|  | 1 |

f3(11)

|  |  |  |
| --- | --- | --- |
| Вывел | Вызвал | Вывел(снизу вверх) |
|  | 11 |  |
| 11 | 9 | 11 |
| 9 | 7 | 9 |
| 7 | 5 | 7 |
| 5 | 3 | 5 |
| 3 | 1 | 3 |
| 1 |  |  |

F1(7)

|  |  |
| --- | --- |
| Вывел | Вызвал |
| 7 | 5 |
| 5 | 3 |
| 3 | 1 |
| 1 |  |

F2(7)

|  |  |
| --- | --- |
| Вызвал | Вывел (снизу вверх) |
| 7 | 7 |
| 5 | 5 |
| 3 | 3 |
| 1 | 1 |

F3(7)

|  |  |  |
| --- | --- | --- |
| Вывел | Вызвал | Вывел(снизу вверх) |
|  | 7 |  |
| 7 | 5 | 7 |
| 5 | 3 | 5 |
| 3 | 1 | 3 |
| 1 |  |  |

recEGE1(3)

|  |  |
| --- | --- |
| Вывел | Вызвал |
|  | 3 |
| 3 | 2 |
| 2 | 1 |
| 1 | 0 |
|  | 0 |
| 1 | 1 |
|  | 0 |
|  | 0 |
| 2 | 2 |
| 1 | 1 |
|  | 0 |
|  | 0 |
| 1 | 1 |
|  | 0 |
|  | 0 |

F1(10)

|  |  |
| --- | --- |
| Вывел | Вызвал |
|  | 10 |
| 10 | 7 |
| 7 | 4 |
| 4 | 1 |
|  | 0 |
|  | 3 |
| 3 | 0 |
|  | -1 |
|  | 6 |
| 6 | 3 |
| 3 | 0 |
|  | -1 |
|  | 2 |

F2(1)

|  |  |
| --- | --- |
| Вывел | Вызвал |
|  | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
|  | 7 |
|  | 6 |
|  | 5 |
|  | 4 |
| 4 | 5 |
|  | 7 |

F3(11)

|  |  |
| --- | --- |
| Вывел | Вызвал |
|  | F(11) |
|  | G(10) |
| \* | F(7) |
|  | G(6) |
| \* | F(3) |
|  | G(2) |
| \* | F(-1) |

doMidasHand

|  |  |
| --- | --- |
| I, j | Вызвал |
| 3, 11 | 2, 11 |
|  | 4, 11 |
| 4, 11 | 3, 11 |
|  | 5, 11 |
| 5, 11 | 4, 11 |
|  | 6, 11 |
|  | 5, 10 |
|  | 5, 12 |
|  | 4, 10 |
|  | 4, 12 |
| 3, 11 | 3, 10 |
|  | 3, 12 |

Image7

|  |  |
| --- | --- |
| Нарисовал | Вызвал |
| 500, 200, 60 | 590, 200, 30 |
| 590, 200, 30 | 635, 200, 15 |
| 635, 200, 15 | 560, 200, 15 |
| 560, 200, 15 | 440, 200, 30 |
| 440, 200, 30 | 485, 200, 15 |
| 485, 200, 15 | 410, 200, 15 |
| 410, 200, 15 |  |

Image9

|  |  |
| --- | --- |
| Нарисовал | Вызвал |
| 800, 200, 60 | 860, 200, 30 |
| 860, 200, 30 | 890, 200, 15 |
| 890, 200, 15 | 830, 200, 15 |
| 830, 200, 15 | 860, 230, 15 |
| 860, 230, 15 | 860, 170, 15 |
| 860, 170, 15 | 740, 200, 30 |
| 740, 200, 30 | 770, 200, 15 |
| 770, 200, 15 | 710, 200, 15 |
| 710, 200, 15 | 740, 230, 15 |
| 740, 230, 15 | 740, 170, 15 |
| 740, 170, 15 | 800, 260, 30 |
| 800, 260, 30 | 830, 260, 15 |
| 830, 260, 15 | 770, 260, 15 |
| 770, 260, 15 | 800, 290, 15 |
| 800, 290, 15 | 800, 230, 15 |
| 800, 230, 15 | 800, 140, 30 |
| 800, 140, 30 | 830, 140, 15 |
| 830, 140, 15 |  |

Код:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include"stdio.h"

#define NUM = 10

long fuct2(int n) {

if (n == 0) {

return 1;

}

long res = fuct2(n - 1) \* n;

return res;

}

void f1(int n) {

printf(" %d", n);

if (n > 1) {

f1(n - 2);

}

}

void f2(int n) {

if (n > 1) {

f2(n - 2);

}

printf(" %d", n);

}

void f3(int n) {

printf(" %d", n);

if (n > 1) {

f3(n - 2);

}

if (n!=1)

printf(" %d", n);

}

void recEGE1(int n) {

if (n >= 1) {

printf(" %d", n);

recEGE1(n - 1);

recEGE1(n - 1);

}

}

void F1(int n)

{

if (n > 2)

{

printf("%d", n);

F1(n - 3);

F1(n - 4);

}

}

void F2(int n)

{

if (n < 5)

{

printf("%d", n);

F2(n + 1);

F2(n + 3);

}

}

void G3(int n);

void F3(int n)

{

if (n > 0)

{

G3(n - 1);

}

}

void G3(int n)

{

printf("\*");

if (n > 1)

{

F3(n - 3);

}

}

int array[10] = { 1, 3, 4, 8, 9, 10, 11, 15, 16, 18 };

void printArray()

{

for (int i = 0; i < 10; i++)

{

printf("%d ", array[i]);

}

printf("\n\n");

}

int contains(int v)

{

int left = 0;

int right = 10 - 1;

while (left <= right)

{

int middle = (left + right) / 2;

if (array[middle] == v)

{

return 1;

}

if (array[middle] < v)

{

left = middle + 1;

}

else

{

right = middle - 1;

}

}

return 0;

}

int containsR(int left, int right, int v)

{

if (left > right)

{

return 0;

}

int middle = (right + left) / 2;

if (array[middle] == v)

{

return 1;

}

if (array[middle] < v)

{

return containsR(middle + 1, right, v);

}

else

{

return containsR(left, middle - 1, v);

}

}

void F1dz(int n) {

if (n > 0) {

F1dz(n - 1);

printf("%d", n);

F1dz(n / 4);

}

}

void F2dz(int n)

{

if (n < 8) {

F2dz(2 \* n);

printf("%d", n);

F2dz(n + 3);

}

}

void F3dz(int n)

{

if (n > 0)

{

printf("\*");

F3dz(n - 1);

F3dz(n / 3);

}

}

int main() {

int k;

while (k != -1)

{

printf("task 1, 2, 4, 5, 6, 9, 10\n");

scanf\_s("%d", &k);

switch (k)

{

case 1:

int n;

scanf\_s("%d", &n);

long f = fuct2(n);

printf("%d! = %ld", n, f);

break;

case 2:

f1(7);

printf(" f1 FINISH\n");

f2(7);

printf(" f2 FINISH\n");

f3(7);

printf(" f3 FINISH\n");

break;

case 4:

recEGE1(3);

break;

case 5:

F1(10);

break;

case 6:

F2(1);

break;

case 7:

F3(11);

break;

case 9:

printArray();

printf("contains (1) = %d\n", contains(1));

printf("contains (2) = %d\n", contains(2));

printf("contains (3) = %d\n", contains(3));

printf("contains (4) = %d\n", contains(4));

printf("contains (5) = %d\n", contains(5));

printf("contains (6) = %d\n", contains(6));

printf("contains (10) = %d\n", contains(10));

printf("contains (15) = %d\n", contains(15));

printf("\n\n");

printf("containsR (1) = %d\n", containsR(0, 9, 1));

printf("containsR (2) = %d\n", containsR(0, 9, 2));

printf("containsR (3) = %d\n", containsR(0, 9, 3));

printf("containsR (4) = %d\n", containsR(0, 9, 4));

printf("containsR (5) = %d\n", containsR(0, 9, 5));

printf("containsR (6) = %d\n", containsR(0, 9, 6));

printf("containsR (10) = %d\n", containsR(0, 9, 10));

printf("containsR (15) = %d\n", containsR(0, 9, 15));

break;

case 10:

F1dz(5);

printf("\n");

F2dz(1);

printf("\n");

F3dz(6);

break;

}

}

}