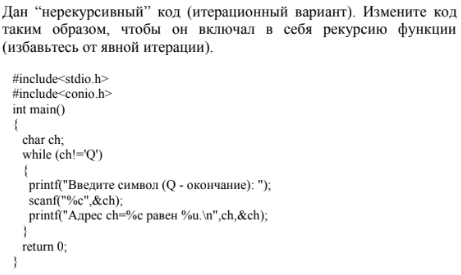
Лабораторная работа №10.

Задание 1.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<windows.h>

**int** main(**void**)

{

**char** ch;

setlocale(LC\_ALL, "Russian");

printf("Введите символ(Q - окончание):");

scanf("%c",&ch);

**if** (ch!='Q'){

printf("Адрес ch=%c равен %u**\n**",ch,&ch);

fflush(stdin);

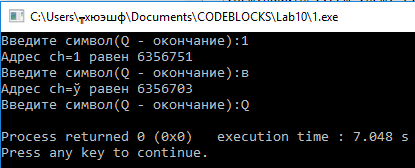
**return** main();

}

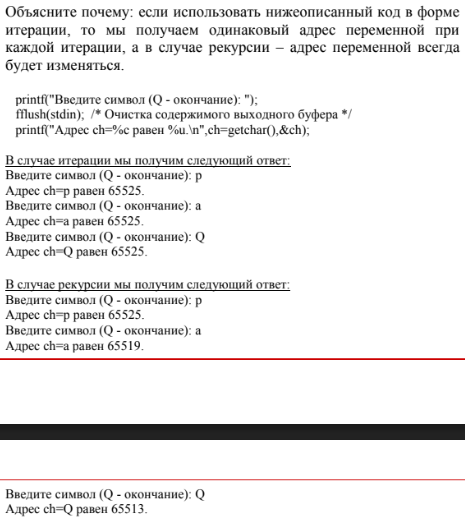
**return** 0;

}

Результат:



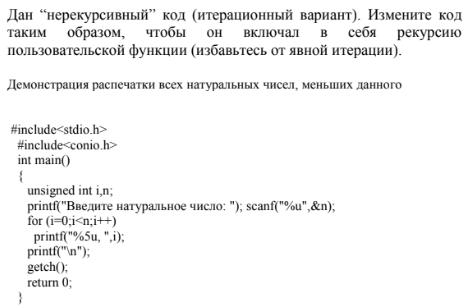
Задание 2.



Объяснение:

Рекурсивная функция при каждом вызове резервирует новую область памяти для переменных. Таким образом, в памяти сохраняются все объявляемые функцией переменные при каждом вызове, и становятся локальными для своего уровня в стеке.

Задание 3.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

**unsigned** **int** printing(**unsigned** **int**);

**int** main()

{

**unsigned** **int** i,n;

setlocale(LC\_ALL,"Russian");

printf("Введите натуральное число: ");

scanf("%u",&n);

printing(n);

getch();

**return** 0;

}

**unsigned** **int** printing(**unsigned** **int** x){

**if** (x==1);

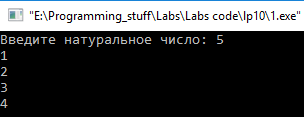
**else**

printf("%d**\n**",printing(x-1));

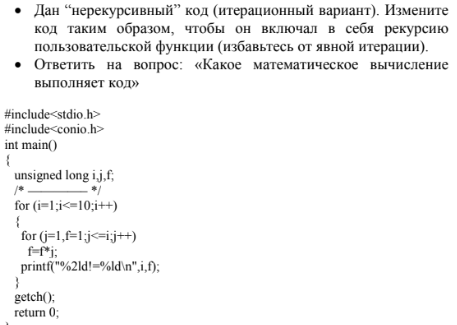
**return** x;

}

Результат:



Задание 4.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

#include<windows.h>

**unsigned** **long** fact(**unsigned** **long**);

**int** main(**void**)

{

setlocale(LC\_ALL,"Russian");

**unsigned** **long** i;

printf("Введите число:**\n**");

scanf("%d",&i);

fact(i);

getch();

**return** 0;

}

**unsigned** **long** fact(**unsigned** **long** x){

**unsigned** **long** a;

**if** (x==1) {

printf("%d!=%d**\n**",x,x);

**return** 1;

}

**else**

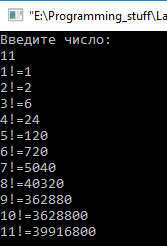
a=fact(x-1);

printf("%d!=%ld**\n**",x,a\*x);

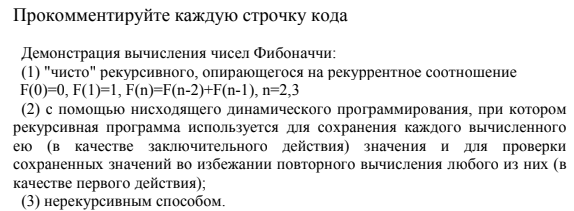
**return** a\*x;

}

Результат:



Задание 5.



Код:

#include<stdio.h> *// Подключение стандартного заголовочного файла ввода-вывода*

#include<conio.h> *// Подключение библиотеки для работы с консолью*

#include<time.h> *// Подключение заголовочного файла стандартной библиотеки, содержащего функции для работы с датой и временем*

#include<locale.h> *// Подключение заголовочного файла стандартной библиотеки языка программирования С, который используется для задач, связанных с локализацией*

#define BOUND 34 *// Задание символической константы*

**long** FibRec(**long**); *// Объявление функции FibRec, принимающей аргумент типа long, возвращающей значение типа long*

**long** FibDinRec(**long**); *// Объявление функции FibDinRec, принимающей аргумент типа long, возвращающей значение типа long*

**long** FibNRec(**long**); *// Объявление функции FibNRec, принимающей аргумент типа long, возвращающей значение типа long*

*/\* —————– \*/*

**int** main() *// Определение функции main с возвратом значения типа int*

{

setlocale(LC\_ALL,"Russian"); *// Функция, задающая языковой станларт для корректного отображения и преобразования текста в операциях ввода-вывода*

**long** k; *// Объявление переменной k типа long*

**time\_t** Before,After; *// Объявление переменных Before, After типа time\_t*

**clock\_t** start,end; *// Объявление переменных start,end типа clock\_t*

printf("Числа Фибоначчи (рекурсивное вычисление):**\n**"); *// Вывод текста, заключенного в кавычки на экран*

**for** (k=28;k<=BOUND;k++) *// Инициализация цикла for, внутри него - переменной k=28, исполнение цикла до равенства k и константы BOUND, шаг - инкрементация k*

{

Before=time(NULL); *// Запись в переменную Before текущего календарного времени*

start=clock(); *// Запись в переменную start времени, прошедшего с начала выполнения программы, измеряемого в тактах процессора*

*/\* —————————– \*/*

printf("%3ld %9ld ",k,FibRec(k)); *// Вывод на экран порядкового номера числа Фибоначчи и самого числа, возвращаемого функцией FibRec*

*/\* —————————– \*/*

After=time(NULL); *// Запись в переменную After текущего календарного времени*

end=clock(); *// Запись в переменную End времени прошедшего с начала выполнения программы, измеряемого в тактах процессора*

printf("%4.1f сек, %6.3f сек**\n**",

difftime(After,Before),

(**double**)(end-start)/(**double**)CLOCKS\_PER\_SEC); *// Вывод в консоль времени, затраченного на вычисления на данной итерации, расчитанного двумя методами:через календарное и процессорное время*

printf("Все ли полученные результаты верны?**\n**"); *// Вывод указанного текста*

*/\* ———————————————— \*/*

printf("Числа Фибоначчи (рекурсивное вычисление):**\n**"); *// Вывод указанного текста*

**for** (k=28;k<=BOUND;k++) *// Инициализация цикла for, внутри него - переменной k=28, исполнение цикла до равенства k и константы BOUND, шаг - инкрементация k*

{

Before=time(NULL); *// Запись в переменную Before текущего календарного времени*

start=clock(); *// Запись в переменную start времени, прошедшего с начала выполнения программы, измеряемого в тактах процессора*

*/\* ——————————– \*/*

printf("%3ld %9ld ",k,FibDinRec(k)); *// Вывод на экран порядкового номера числа Фибоначчи и самого числа, возвращаемого функцией FibRec*

*/\* ——————————– \*/*

After=time(NULL); *// Запись в переменную After текущего календарного времени*

end=clock(); *// Запись в переменную End времени прошедшего с начала выполнения программы, измеряемого в тактах процессора*

printf("%4.1f сек, %6.3f сек**\n**",

difftime(After,Before),

(**double**)(end-start)/(**double**)CLOCKS\_PER\_SEC); *// Вывод в консоль времени, затраченного на вычисления на данной итерации, расчитанного двумя методами:через календарное и процессорное время*

}

printf("Все ли полученные результаты верны?**\n**"); *// Вывод указанного текста*

*/\* ————————————————– \*/*

printf("Числа Фибоначчи (нерекурсивное вычисление):**\n**"); *// Вывод указанного текста*

**for** (k=28;k<=BOUND;k++) *// Инициализация цикла for, внутри него - переменной k=28, исполнение цикла до равенства k и константы BOUND, шаг - инкрементация k*

{

Before=time(NULL); *// Запись в переменную Before текущего календарного времени*

start=clock(); *// Запись в переменную start времени, прошедшего с начала выполнения программы, измеряемого в тактах процессора*

*/\* —————————— \*/*

printf("%3ld %9ld ",k,FibNRec(k)); *// Вывод на экран порядкового номера числа Фибоначчи и самого числа, возвращаемого функцией FibRec*

*/\* —————————— \*/*

After=time(NULL); *// Запись в переменную After текущего календарного времени*

end=clock(); *// Запись в переменную End времени прошедшего с начала выполнения программы, измеряемого в тактах процессора*

printf("%4.1f сек, %6.3f сек**\n**",

difftime(After,Before),

(**double**)(end-start)/(**double**)CLOCKS\_PER\_SEC); *// Вывод в консоль времени, затраченного на вычисления на данной итерации, расчитанного двумя методами:через календарное и процессорное время*

}

printf("Все ли полученные результаты верны?**\n**"); *// Вывод указанного текста*

getch(); *//*

**return** 0;

}

*/\* ————— \*/*

**long** FibRec(**long** n)

*/\* Рекурсивная функция для вычисления чисел Фибоначчи. \*/*

*/\* Эта функция, хотя и выглядит компактно и изящно, не- \*/*

*/\* применима на практике, поскольку время вычисления n- \*/*

*/\* го числа Фибоначчи экспоненциально зависит от n \*/*

*/\* —————————————————- \*/*

{

**if** (n==0||n==1)

**return** n;

**else** **return** FibRec(n-2)+FibRec(n-1);

}

*/\* —————— \*/*

**long** FibDinRec(**long** i) *// Определение функции FibDinRec*

*/\* Сохранение вычисляемых значений в статическом массиве \*/*

*/\* (элементы которого инициализируются 0) явным образом \*/*

*/\* исключает любые повторные вычисления. Эта функция вы- \*/*

*/\* числяет n-е число Фибоначчи за время, пропорциональное \*/*

*/\* n \*/*

*/\* —————————————————— \*/*

{

**static** **long** knownF[40]; *// Объявление статического локального массива из 40 элементов*

**long** t=i; *// Дублирование аргумента*

**if** (knownF[i]!=0) *// Проверка на*

**return** knownF[i]; *// Возврат значения элемента массива с данным индексом*

**if** (i<0) *//*

**return** 0; *// Возврат 0 при индексе < 0*

**if** (i>1)

t=FibDinRec(i-1)+FibDinRec(i-2); *// При индексе > 1 в t записывается сумма предыдущих чисел Фибоначчи, подсчитываемая вызовами этой же функции*

**return** knownF[i]=t; *// Возвращение расчитанной суммы и её запись в элемент массива с индексом i*

}

*/\* —————- \*/*

**long** FibNRec(**long** x) *// Опрееление функции FibNRec*

*/\* Нерекурсивная функция для вычисления чисел Фибоначчи \*/*

*/\* с использованием массива \*/*

*/\* —————————————————- \*/*

{

**long** i,a[3]; *// Объявление переменной i и массива a из трех элементов типа long*

**if** (x<3) *// Логический оператор if, проверяющий условие x<3*

**return** 1; *// Возврат единицы в случае истинности вышеописанного условия*

a[1]=a[0]=1; *// Присвоение элементам массива с индексами 1 и 0 значения 1 - первые два числа Фибоначчи*

**for** (i=2;i<x;++i) *// Инициализация цикла for с начальным значением параметра i=2, условием выполнения i<x, шаг - инкрементация i*

{

a[2]=a[0]+a[1]; a[0]=a[1]; a[1]=a[2]; *// Запись в элемент массива a с индексом 2 суммы первых двух элементов, перезапись результата и предыдущего числа Фибоначчи в ячейки массива с индексами 0 и 1 соответсвтенно*

}

**return** a[2]; *// Функция возвращает второй элемент массива a - найденное число Фибоначчи*

}

Результат:

