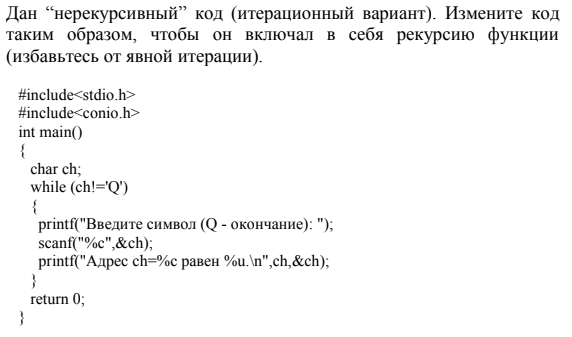
# Лабораторная работа 10

## Задание 1



## Код программы

#include<stdio.h>

#include<conio.h>

#include<locale.h>

void funct (char t) {

if (t == 'Q') return 0;

scanf("%c",&t);

printf("Aдрес ch=%c равен %u.\n",t,&t);

funct (t);

}

int main()

{

char \*locale = setlocale(LC\_ALL, "");

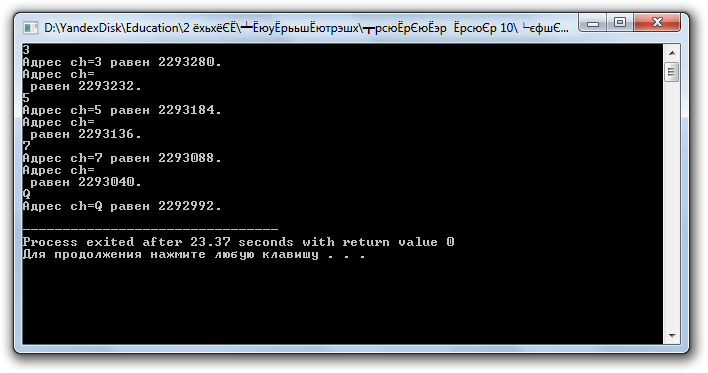
char ch;

funct (ch);

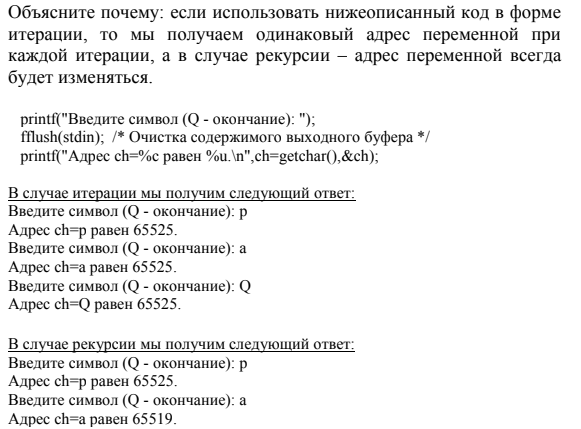
return 0;

}

## Результат проделанной работы

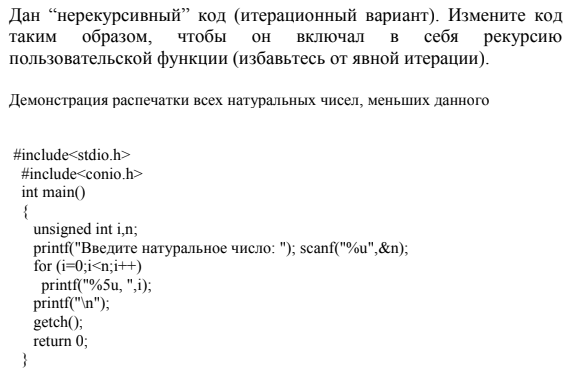


## Задание 2



Вся причина в том, что рекурсивные функции используют так называемый «Стек вызовов». Когда программа вызывает функцию, функция отправляется на верх стека вызовов. Это похоже на стопку книг, мы добавляем одну вещь за одни раз. Затем, когда вы готовы снять что-то обратно, мы всегда снимаем верхний элемент.

## Задание 3



## Код программы

#include<stdio.h>

#include<locale.h>

int funct(int t) {

if (t == 0) return 0; //Ноль к натуральным числам не относим

printf ("%d\n",t);

funct (t-1);

}

void main() {

char \*locale = setlocale(LC\_ALL, "");

int n;

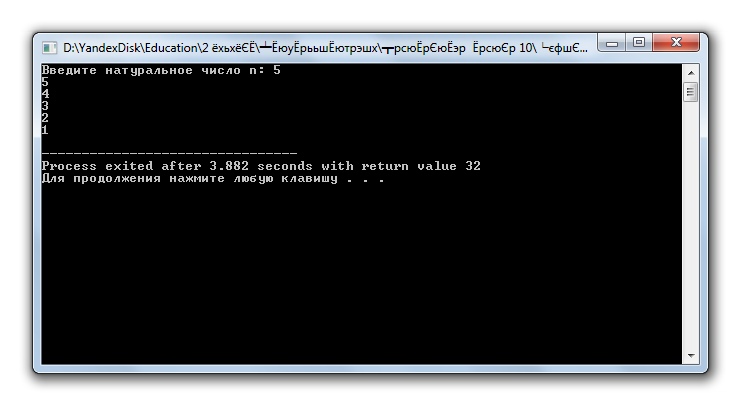
printf("Введите натуральное число n: "); scanf("%d", &n);

funct(n);

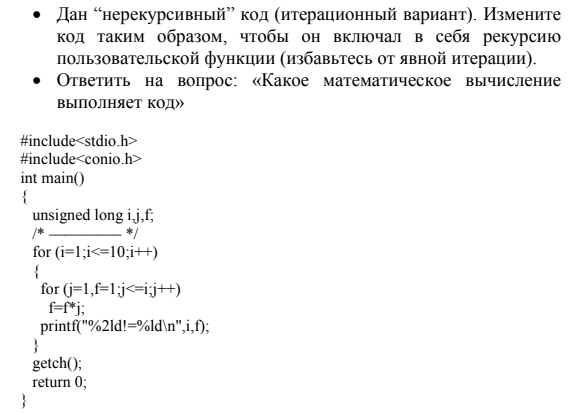
getch();

}

## Результат работы программы



## Задание 4



## Код программы

#include<stdio.h>

#include<locale.h>

#include<math.h>

long int factorial(long int t) {

if (t == 0 || t == 1) return 1;

return t \* factorial(t - 1);

}

void main() {

char \*locale = setlocale(LC\_ALL, "");

int i=1;

for (i;i<=10;i++) {

printf("%ld!=%ld\n",i,factorial(i));

}

getch();

}

## Результат проделанной работы

Программа вычисляет факториал чисел от 1 до 10

