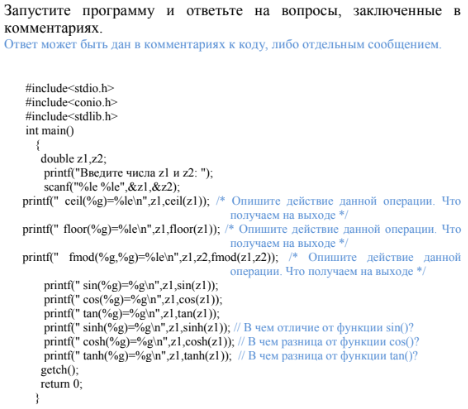
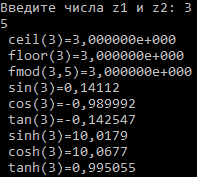
Лабораторная работа №8. Функции в языке C.

Задание 1.



Результат выполнения:



Код(комментарии):

#include<stdio.h>

#include<conio.h>

#include<locale.h>

int main()

{

setlocale(LC\_ALL,"Russian");

double z1,z2;

printf("Введите числа z1 и z2: ");

scanf("%le %le",&z1,&z2);

printf(" ceil(%g)=%le\n",z1,ceil(z1)); /\* Опишите действие данной операции. Что

получаем на выходе \*/

/\* Спецификатор вывода %le указывает на вещественное число типа double в

десятичной E-записи.

Функция ceil возвращает ближайшее целое значение к z1, но большее z1 \*/

printf(" floor(%g)=%le\n",z1,floor(z1)); /\* Опишите действие данной операции. Что

получаем на выходе \*/

/\* Спецификатор вывода %le указывает на вещественное число типа double в

десятичной E-записи.

Функция floor возвращает ближайшее целое значение к z1, не превышающее z1\*/

printf(" fmod(%g,%g)=%le\n",z1,z2,fmod(z1,z2)); /\* Опишите действие данной

операции. Что получаем на выходе \*/

/\* Спецификатор вывода %le указывает на вещественное число типа double в

десятичной E-записи.

Функция fmod возвращает остаток от деления вещественных чисел \*/

printf(" sin(%g)=%g\n",z1,sin(z1));

printf(" cos(%g)=%g\n",z1,cos(z1));

printf(" tan(%g)=%g\n",z1,tan(z1));

printf(" sinh(%g)=%g\n",z1,sinh(z1)); // В чем отличие от функции sin()?

printf(" cosh(%g)=%g\n",z1,cosh(z1)); // В чем разница от функции cos()?

printf(" tanh(%g)=%g\n",z1,tanh(z1)); // В чем разница от функции tan()?

/\* sinh, cosh, tanh - гиперболические функции, определяемые через e и тесно

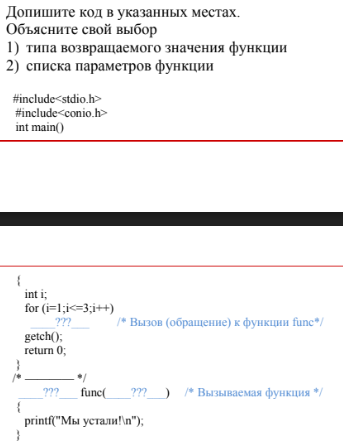
связанные с тригонометрическими функциями. \*/

getch();

return 0;

}

Задание 2.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

void func(void);

int main()

{

setlocale(LC\_ALL,"Russian");

int i;

for (i=1;i<=3;i++)

func();/\* Вызов (обращение) к функции func\*/

getch();

return 0;

}

/\* Т.к. управление циклом осуществляется непосредственно в теле main, а назначение

вызываемой функции - вывод на экран сообщения, то очевидно, что функции не передаются

параметры, а возвращаемое значение отсутствует. Поэтому используется void func(void) \*/

void func(void) /\* Вызываемая функция \*/

{

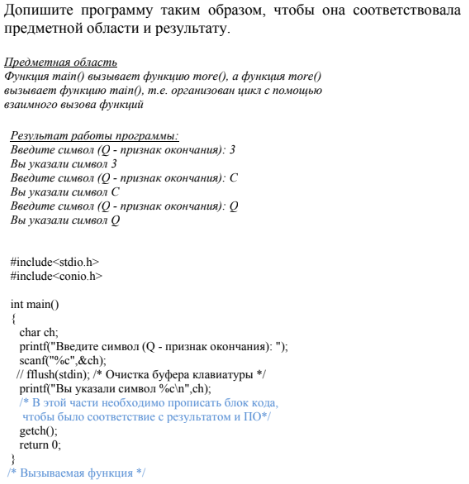
printf("Мы устали!\n");

}

Результат выполнения:



Задание 3.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

int more(void);

int main()

{

setlocale(LC\_ALL,"Russian");

char ch;

printf("Введите символ (Q - признак окончания): ");

scanf("%c",&ch);

fflush(stdin); /\* Очистка буфера клавиатуры \*/

printf("Вы указали символ %c\n",ch);

/\* В этой части необходимо прописать блок кода,

чтобы было соответствие с результатом и ПО\*/

if (ch!='Q')

more();

return 0;

}

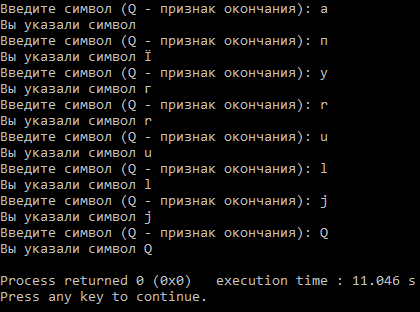
/\* Вызываемая функция \*/

int more(void){

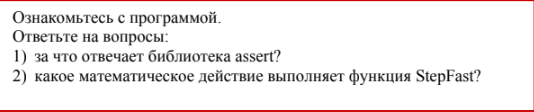
return main();

}

Результат выполнения:



Задание 4.



Код:

#include <stdio.h>

#include <conio.h>

#include <assert.h>

#include <locale.h>

int main() {

setlocale(LC\_ALL, "Russian");

printf("Тестирование...\n");

assert(StepFast(5, 3) == 125);

assert(StepFast(11, 4) == 121\*121);

printf("Тестирование неуспешно: ошибок не обнаружено.\n");

getch();

return 0;

}

int StepFast(int x, int y) {

int a = x, b = y, z = 1;

while (b != 0)

if (odd(b)) {

z \*= a;

b--;

} else {

a \*= a;

b /= 2;

}

return z;

}

int odd(int t) {

return (t % 2 == 0) ? 0 : 1;

}

/\* assert.h - заголовочный файл стандартной библиотеки,

покдлючается для использования макроса препроцессора

assert(), позволяющего выполнить проверку верности

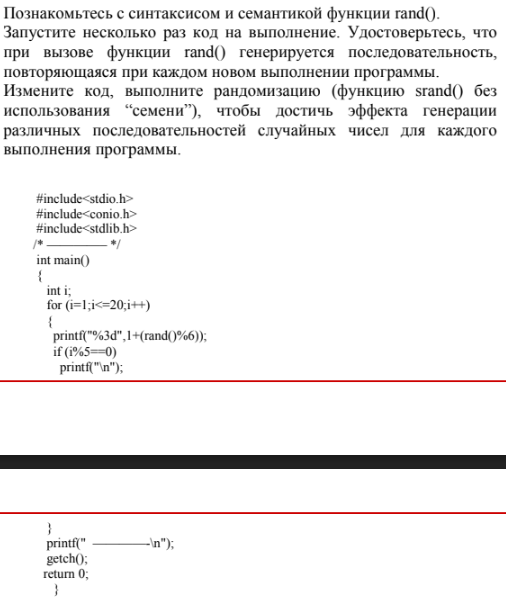
выражения. Если результат сравнения 0 - программа останавливается.\*/

/\* Функция StepFast реализует бинарный алгоритм возведения в степень \*/

Результат выполнения:



Задание 5.



Код:

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

/\* ————– \*/

int main()

{

int i;

srand(time(NULL));

for (i=1;i<=20;i++)

{

printf("%3d",1+(rand()%6));

if (i%5==0)

printf("\n");

}

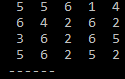
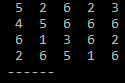
printf(" ------\n");

getch();

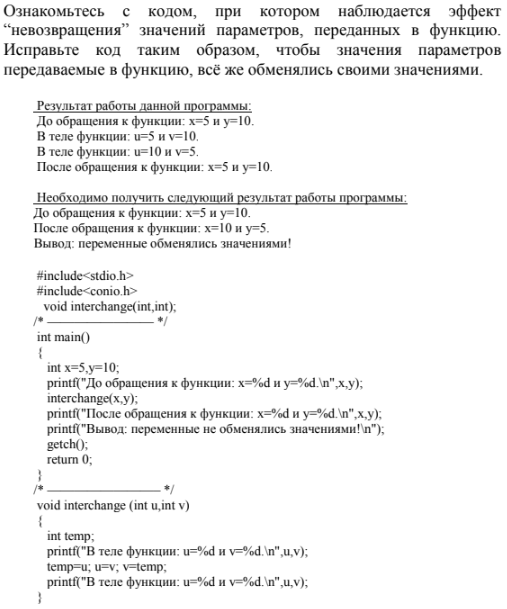
return 0;

}

Результаты выполнения:

Задание 6.



Код:

#include<stdio.h>

#include<conio.h>

#include<locale.h>

void interchange(int \*u,int \*v);

/\* ———————— \*/

int main()

{

setlocale(LC\_ALL,"Russian");

int x=5,y=10;

printf("До обращения к функции: x=%d и y=%d.\n",x,y);

interchange(&x,&y);

printf("После обращения к функции: x=%d и y=%d.\n",x,y);

printf("Вывод: переменные обменялись значениями!\n");

getch();

return 0;

}

/\* ————————– \*/

void interchange (int \*u,int \*v)

{

int temp;

printf("В теле функции: u=%d и v=%d.\n",\*u,\*v);

temp=\*u; \*u=\*v; \*v=temp;

printf("В теле функции: u=%d и v=%d.\n",\*u,\*v);

}

Результат выполнения:

