

Лабораторная работа № 8

Функции в языке с

Задание № 2.

Постановка задачи:

Допишите код в указанных местах.

Объясните свой выбор

- 1) типа возвращаемого значения функции
- 2) списка параметров функции

Код:

```
#include<stdio.h>
#include<conio.h>
#include <locale.h>
void func()
{
    printf("Мы устали!\n");
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    int i;
    for (i=1;i<=3;i++)
        func();
    getch();
    return 0;
}
```

Результат вывода:

```
Мы устали!
Мы устали!
Мы устали!
```

Ответ:

- 1) Возвращаемое значение типа void, тк функция ничего не возвращает, а только выводит фразу
- 2) Функция не принимает в себя никакие аргументы

Задание № 3.

Постановка задачи:

Допишите программу таким образом, чтобы она соответствовала предметной области и результату.

Код:

```

#include<stdio.h>
#include<conio.h>
#include<locale.h>
void more(char a)
{
    if (a != 'Q')
        main();
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    char ch;
    printf("Введите символ (Q - признак окончания): ");
    scanf("%c",&ch);
    fflush(stdin); /* Очистка буфера клавиатуры */
    printf("Вы указали символ %c\n",ch);
    more(ch);
    getch();
    return 0;
}

```

Результат вывода:

```

Введите символ (Q - признак окончания): З
Вы указали символ З
Введите символ (Q - признак окончания): 4
Вы указали символ 4
Введите символ (Q - признак окончания): е
Вы указали символ ?
Введите символ (Q - признак окончания): Q
Вы указали символ Q

```

Задание № 4.

Постановка задачи:

Ознакомьтесь с программой.

Ответьте на вопросы:

- 1) за что отвечает библиотека assert?
- 2) какое математическое действие выполняет функция StepFast?

Код:

```

#include<stdio.h>
#include<conio.h>
#include<locale.h>
#include<assert.h>
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;
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    printf("Тестирование...\n");
    assert(StepFast(5,3)==125);
    assert(StepFast(11,4)==121*121);
    printf("Тестирование неуспешно: ошибок не обнаружено.\n");
    getch();
    return 0;
}

```

Результат вывода:

```

Тестирование...
Тестирование неуспешно: ошибок не обнаружено.

```

Ответ:

- 1) Проверка вычислений, сделанных программой
- 2) Возведение в степень

Задание № 5.

Постановка задачи:

Познакомьтесь с синтаксисом и семантикой функции `rand()`. Запустите несколько раз код на выполнение. Удостоверьтесь, что при вызове функции `rand()` генерируется последовательность, повторяющаяся при каждом новом выполнении программы. Измените код, выполните рандомизацию (функцию `srand()` без использования “семени”), чтобы достичь эффекта генерации различных последовательностей случайных чисел для каждого выполнения программы.

Код:

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<locale.h>
int main()
{
    srand(time(NULL));
    char*locale = setlocale(LC_ALL, "");
    int i;
    for (i=1;i<=20;i++)
    {
        printf("%3d",1+(rand()%6));
        if (i%5==0)
            printf("\n");
    }
    printf(" ----\n");

    getch();
    return 0;
}

```

Результат вывода:

```

 5  5  4  3  1
 2  4  5  5  1
 6  4  1  2  4
 4  4  4  6  2
----

```

Задание № 1.

Постановка задачи:

Запустите программу и ответьте на вопросы, заключенные в комментариях.

Ответ может быть дан в комментариях к коду, либо отдельным сообщением.

Код:

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<locale.h>
#include<math.h>
int main()
{
    char*locale = setlocale(LC_ALL, "");
    double z1,z2;
    printf("Введите числа z1 и z2: ");
    scanf("%le %le",&z1,&z2);
    printf(" ceil(%g)=%le\n",z1,ceil(z1)); /* Вычисляет наименьшее целое, значение которого не будет меньше, чем аргумент. */
    printf(" floor(%g)=%le\n",z1,floor(z1)); /* Эта функция вычисляет наибольшее целое, по значению не превосходящее x */
    printf(" fmod(%g,%g)=%le\n",z1,z2,fmod(z1,z2)); /* Получаем остаток от деления первого аргумента на второй */

    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)); //Синус гиперболический x
    printf(" cosh(%g)=%g\n",z1,cosh(z1)); //Косинус гиперболический x
    printf(" tanh(%g)=%g\n",z1,tanh(z1)); //Тангенс гиперболический x
    getch();
    return 0;
}

```

Результат вывода:

```
Введите числа z1 и z2: 2
4
ceil(2)=2,000000e+000
floor(2)=2,000000e+000
fmod(2,4)=2,000000e+000
sin(2)=0,909297
cos(2)=-0,416147
tan(2)=-2,18504
sinh(2)=3,62686
cosh(2)=3,7622
tanh(2)=0,964028
```

Задание № 6.

Постановка задачи:

Ознакомьтесь с кодом, при котором наблюдается эффект “невозвращения” значений параметров, переданных в функцию. Исправьте код таким образом, чтобы значения параметров передаваемые в функцию, всё же обменялись своими значениями.

Результат работы данной программы:

До обращения к функции: x=5 и y=10.

В теле функции: u=5 и v=10.

В теле функции: u=10 и v=5.

После обращения к функции: x=5 и y=10.

Необходимо получить следующий результат работы программы:

До обращения к функции: x=5 и y=10.

После обращения к функции: x=10 и y=5.

Вывод: переменные обменялись значениями!

Код:

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
void interchange(int *,int *);
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);
}
int main()
{
    char*locale = setlocale(LC_ALL, "");
    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;
}
```

Результат вывода:

До обращения к функции: $x=5$ и $y=10$.
В теле функции: $u=5$ и $v=10$.
В теле функции: $u=10$ и $v=5$.
После обращения к функции: $x=10$ и $y=5$.
Вывод: переменные обменялись значениями!