

Лабораторная работа 7.

Указатели на простейшие типы.

Операции над указателями.

1.

Постановка задачи: Запустите программу и ответьте на вопросы, заключенные в комментариях.

Ответ: #include<stdio.h>

#include<math.h>

//#include<conio.h>

#include<stdlib.h>

int main()

{

double z1,z2;

printf("Введите числа z1 и z2: ");

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

printf(" ceil(%g)=%le\n",z1,ceil(z1)); // округление до верхнего значения

printf(" floor(%g)=%le\n",z1,floor(z1)); // округление до нижнего значения

printf(" fmod(%g,%g)=%le\n",z1,z2,fmod(z1,z2)); // остаток от деления z1 на z2

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)); // гиперболический синус?

printf(" cosh(%g)=%g\n",z1,cosh(z1)); // гиперболический косинус?

printf(" tanh(%g)=%g\n",z1,tanh(z1)); // гиперболический тангенс?

//getch();

return 0;

}

2.

Постановка задачи: Допишите код в указанных местах.

Объясните свой выбор

1) типа возвращаемого значения функции

2) списка параметров функции

Ответ:

```
#include<stdio.h>
//#include<conio.h>
void func() /* Вызываемая функция */
{
printf("Мы устали!\n");
}
int main()
{
int i;

for (i=1;i<=3;i++)
func();
return 0;
}
```

Мною был выбран тип void т.к. задача данной функции – вывести текст на экран, ничего более от нее не нужно, по этой же причине список параметров функции пустой.

3.

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

Предметная область

Функция main() вызывает функцию more(), а функция more() вызывает функцию main(), т.е. организован цикл с помощью взаимного вызова функций

Результат работы программы:

Введите символ (Q - признак окончания): З

Вы указали символ З

Введите символ (Q - признак окончания): С

Вы указали символ С

Введите символ (Q - признак окончания): Q

Вы указали символ Q

Ответ: #include<stdio.h>

```
//#include<conio.h>
```

```
void more(char a)
```

```
{  
    if (a!='Q')  
        main();  
}
```

```
int main()
```

```
{  
    char ch;  
    printf("Введите символ (Q - признак окончания): ");  
    scanf("%c",&ch);  
    // fflush(stdin); /* Очистка буфера клавиатуры */  
    printf("Вы указали символ %c\n",ch);  
    // getch();  
    more(ch);  
    return 0;  
}
```

4.

Постановка задачи: Ознакомьтесь с программой.

Ответьте на вопросы:

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

Ответ: 1) assert – проверка условия для своевременного выключения программы, в случае несоответствия тестовых данных.

- 2) возведение в степень.

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.

Ответ: #include<stdio.h>

//#include<conio.h>

#include<stdlib.h>

/* ———— */

int main()

{

srand(time(NULL));

int i;

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

{

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

if (i%5==0)

printf("\n");

}

printf(" ————\n");

//getch();

return 0;

}

6.

Постановка задачи:

Ответ: #include<stdio.h>

//#include<conio.h>

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()
{
    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 УРА!\n");
    //getch();
    return 0;
}
/* ----- */
```