

Лабораторная работа 1 (0001 = 1)

Ввод-вывод при помощи libc

Операционная система ОС Windows, 64-разрядная ОС.

Задание Л1.31.

Разработайте программу, выводящую на стандартный вывод группу, номер и состав команды при помощи функции `puts()` библиотеки `libc`. При работе в ОС MS Windows возможны проблемы с кодировкой русского языка. Если они возникли — используйте транслит или любые доступные вам способы настройки.

Программный код:

```
#include <iostream>

using namespace std;

int main()
{
    char string[] = "PIN-22\nTEAM 5\nNadezhkina Anna\nTrusov Mikhail";
    puts(string);
    return 0;
}
```

Вывод:

PIN-22

TEAM 5

Nadezhkina Anna

Trusov Mikhail

Задание Л1.32.

Укажите для платформы, где выполняется работа:

- ОС и разрядность ОС;
- компилятор (должен относиться к коллекции GCC/MinGW) и его версию;
- разрядность сборки (собираемая программа может работать в 32-битном режиме даже под 64-битной ОС — в режиме совместимости);

– архитектуру процессора, назначение платформы. Компьютер с процессором x86/x86-64 под управлением GNU/Linux, BSD (в том числе Mac OS X) или MS Windows — платформа общего назначения

При помощи оператора `sizeof` языка C/C++ выясните, сколько байтов занимают на выбранной платформе переменные следующих типов: `char`, `signed char`, `unsigned char`, `char*`, `bool`, `wchar_t`, `short`, `int`, `long`, `long long`, `float`, `double`, `long double`, `size_t`, `ptrdiff_t`, `void*`

Бонус +2 балла, если при помощи макроса пояснения выводятся так, что в коде каждое имя типа в Л1.32 встречается единожды.

ОС и разрядность ОС: Windows 64 разрядная

Компилятор и его версия: GCC 13.2.0

Разрядность сборки: 64 бит

Архитектура процессора: x64 и MS Windows — платформа общего назначения

Программный код:

```
#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#define print_sizeof(type) printf("%s: %d\n", #type, sizeof(type));

using namespace std;

int main()
{
    print_sizeof(char);
    print_sizeof(signed char);
    print_sizeof(unsigned char);
    print_sizeof(char*);
    print_sizeof(bool);
    print_sizeof(wchar_t);
    print_sizeof(short);
    print_sizeof(int);
    print_sizeof(long);
    print_sizeof(long long);
    print_sizeof(float);
    print_sizeof(double);
    print_sizeof(long double);
```

```

    print_sizeof(size_t);
    print_sizeof(ptrdiff_t);
    print_sizeof(void*);

    return 0;
}

```

Вывод:

```

char: 1
signed char: 1
unsigned char: 1
char*: 8
bool: 1
wchar_t: 2
short: 2
int: 4
long: 8
long long: 8
float: 4
double: 8
long double: 16
size_t: 8
ptrdiff_t: 8
void*: 8

```

Задание Л1.34. Вариант 1 ($x = 5/3$).

Разработайте программу на языке C/C++, создающую массивы из $N = 5$ чисел и инициализирующую их N одинаковыми значениями x :

- Ms из 16-битных целых чисел ($x = 0xFADE$);
- Ml из 32-битных целых чисел ($x = 0xADE1A1DA$);
- Mq из 64-битных целых чисел ($x = 0xC1A551F1AB1E$);
- Mfs из 32-битных чисел с плавающей запятой (x из таблицы Л1.2);
- Mfl из 64-битных чисел с плавающей запятой (x из таблицы Л1.2). Тип элементов каждого из массивов определите по результатам заданий Л1.32/Л1.33

Выведите каждый из массивов на экран при помощи функции *libc printf()*

Бонус +1 балл, если вывод массива в двух формах описан как функция и в последующих заданиях используется вызов этой функции, а не копирование и вставка; **+2 балла**, если эта функция описана как единый для всех массивов шаблон и принимает тип как параметр шаблона, а адрес начала *M*, длину *N* и форматы с модификатором размера как параметры функции

Программный код:

```
#include <iostream>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    const int N = 5;
```

```
    // 16-битные целые числа
```

```
    short Ms[N];
```

```
    for (int i = 0; i < N; i++) {
```

```
        Ms[i] = 0xFADE;
```

```
    }
```

```
    // 32-битные целые числа
```

```
    int MI[N];
```

```
    for (int i = 0; i < N; i++) {
```

```
        MI[i] = 0xADEIAIDA;
```

```
    }
```

```
    // 64-битные целые числа
```

```
    long long Mq[N];
```

```
    for (int i = 0; i < N; i++) {
```

```
        Mq[i] = 0xCIA55IFIABIE;
```

```
    }
```

```
    // 32-битные целые числа с плавающей запятой
```

```

float Mfs[N];
for (int i = 0; i < N; i++) {
    Mfs[i] = 5.0/3.0;
}

// 64-битные целые числа с плавающей запятой
double Mfl[N];
for (int i = 0; i < N; i++) {
    Mfl[i] = 5.0/3.0;
}

// Вывод целочисленных массивов
template<typename T>
void printArray(T *arr, int N, const char *mod){
    for (int i = 0; i < N; i++) {
        printf(mod, arr[i], arr[i]);
    }
}

printf("16-bit short\n");
printArray(Ms, N, "%d, %04hX\n");

printf("32-bit int\n");
printArray(Ml, N, "%d, %08X\n");

printf("64-bit long long\n");
printArray(Mq, N, "%d, %016lX\n");

printf("32-bit float\n");
printArray(Mfs, N, "%.2f, %e\n");

printf("64-bit double\n");
printArray(Mfl, N, "%.2lf, %le\n");

```

Вывод:

16-bit short

-1314, FADE

-1314, FADE

-1314, FADE

-1314, FADE

-1314, FADE

32-bit int

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

64-bit long long

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

32-bit float

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

64-bit double

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

1.67, 1.666667e+00

Задание Л1.35. Вариант 5 (Mfl).

Для одного из массивов M (по варианту согласно таблице Л1.3) выведите на экран адреса:

– начала массива — M ;

– начального (нулевого) элемента массива — $\&(M[0])$;

– следующего (с индексом 1) элемента массива — $\&(M[1])$; при помощи функции `libc printf()` как указатели (формат p). Сравните полученные значения между собой и с размером элемента массива M .

Программный код:

```
printf("Adress massiva: %p\n", &Mfl);  
printf("Adress nulevogo elementa massiva: %p\n", &Mfl[0]);  
printf("Adress pervogo elementa massiva: %p\n", &Mfl[1]);
```

Вывод:

```
Adress massiva: 0x7ffffcb20  
Adress nulevogo elementa massiva: 0x7ffffcb20  
Adress pervogo elementa massiva: 0x7ffffcb28
```

Задание Л1.36.

Для каждого массива M из пяти созданных введите с клавиатуры новое значение элемента $M[i]$, $i = 2$ при помощи функции `libc scanf()`. Проанализировав возвращённое `scanf()` значение, определите корректность ввода; при необходимости отобразите сообщение об ошибке при помощи функции `libc puts()`. Очистка буфера после некорректного ввода во всех заданиях данной лабораторной работы не обязательна. Выведите массивы на экран до и после ввода, каждый раз — в обеих формах, описанных в Л1.34; убедитесь, что элемент $M[i]$ приобрёл ожидаемое значение, а другие элементы массива не изменились (если изменились — проверьте, верно ли вы указали модификатор размера).

Программный код:

```
printf("Enter new Ms[2]: ");  
  
if (scanf("%hd", &Ms[2]) != 1) {  
    puts("Error");  
}  
  
printf("16-bit short\n");  
  
for (int i = 0; i < N; i++) {  
    printf("%d, %04hX\n", Ms[i], Ms[i]);  
}
```

```
printf("Enter new Ml[2]: ");  
if (scanf("%d", &Ml[2]) != 1) {  
    puts("Error");  
}
```

```
printf("32-bit int\n");  
for (int i = 0; i < N; i++) {  
    printf("%d, %08X\n", Ml[i], Ml[i]);  
}
```

```
printf("Enter new Mq[2]: ");  
if (scanf("%lld", &Mq[2]) != 1) {  
    puts("Error");  
}
```

```
printf("64-bit long long\n");  
for (int i = 0; i < N; i++) {  
    printf("%d, %016lX\n", Mq[i], Mq[i]);  
}
```

```
printf("Enter new Mfs[2]: ");  
if (scanf("%f", &Mfs[2]) != 1) {  
    puts("Error");  
}
```

```
printf("32-bit float\n");  
for (int i = 0; i < N; i++) {  
    printf("%.2f, %e\n", Mfs[i], Mfs[i]);  
}
```

```
printf("Enter new Mfl[2]: ");  
if (scanf("%lf", &Mfl[2]) != 1) {
```



```

    puts("Error");
}

printf("64-bit double\n");
for (int i = 0; i < N; i++) {
    printf("%.2lf, %le\n", Mf[i], Mf[i]);
}

```

Вывод:

Enter new Ms[2]:256

16-bit short

-1314, FADE

-1314, FADE

256, 0100

-1314, FADE

-1314, FADE

Enter new Ml[2]:4526

32-bit int

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

4526, 000011AE

-1377721894, ADEIAIDA

-1377721894, ADEIAIDA

Enter new Mq[2]:465214

64-bit long long

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

465214, 0000000000007193E

1374792478, 0000CIA55IFIABIE

1374792478, 0000CIA55IFIABIE

Enter new Mfs[2]:2541.54

32-bit float

1.67, 1.666667e+00

1.67, 1.666667e+00

2541.54, 2.541540e+03

1.67, 1.666667e+00

1.67, 1.666667e+00

Enter new Mf[2]:45654.54

64-bit double

1.67, 1.666667e+00

1.67, 1.666667e+00

45654.54, 4.565454e+04

1.67, 1.666667e+00

1.67, 1.666667e+00