

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных
систем

Лабораторная работа №1

по дисциплине: Алгоритмы и структуры данных
тема: «Встроенные структуры данных»

Выполнил: ст. группы ПВ-202
Буйвало Анастасия Андреевна

Проверил:
Кабалянц Петр Степанович
Маньшин Илья Михайлович

Белгород 2021 г.

Лабораторная работа №1

«Встроенные структуры данных»

Цель работы:

Изучение базовых типов данных языка C как структур данных (СД).

Вариант №7:

Тип 1	Тип 2	Тип 3
long int	double	{ cat, dog,mouse,tiger } animal

Задание:

1. Для типов данных (см. Варианты заданий в таблицах 1,2) определить:
 - 1.1. Абстрактный уровень представления СД:
 - 1.1.1. Характер организованности и изменчивости.
 - 1.1.2. Набор допустимых операций.
 - 1.2. Физический уровень представления СД:
 - 1.2.1. Схему хранения.
 - 1.2.2. Объем памяти, занимаемый экземпляром СД.
 - 1.2.3. Формат внутреннего представления СД и способ его интерпретации.
 - 1.2.4. Характеристику допустимых значений.
 - 1.2.5. Тип доступа к элементам.
 - 1.3. Логический уровень представления СД.
Способ описания СД и экземпляра СД на языке программирования.
2. Для заданных типов данных определить набор значений, необходимый для изучения физического уровня представления СД.
3. Преобразовать значения в двоичный код.
4. Преобразовать двоичный код в значение.
5. Разработать и отладить программу, выдающую двоичное представление значений, заданных СД.
В программе использовать процедуры PrintByte и PrintVar.

Спецификация процедуры PrintByte:

1. Заголовок: procedure PrintByte(a:byte)/void PrintByte(unsigned char a).

2. Назначение: выводит на экран монитора двоичное представление переменной `a` типа `byte/unsigned char`.

3. Входные параметры: `a`.

4. Выходные параметры: нет.

Рекомендации: использовать побитовые операции сдвига и логического умножения.

Спецификация процедуры `PrintVar`:

1. Заголовок: `procedure PrintVar(var a; size:word)/ void PrintVar(void a, unsigned int size)`.

2. Назначение: выводит на экран монитора двоичное представление переменной `a` произвольного типа размером `size` байт.

3. Входные параметры: `a` — переменная произвольного типа, значение которой выводится на экран в двоичном представлении (нетипизованный параметр); `size` — объем памяти (в байтах) занимаемый переменной `a`.

4. Выходные параметры: нет.

Рекомендации: нетипизованную переменную `a` привести к типу «массив байт», значение каждого элемента которого выводить на экран в двоичном представлении процедурой `PrintByte`.

6. Обработать программой значения, полученные в результате выполнения пункта 3 задания. Сделать выводы.

7. Разработать и отладить программу, определяющую значение переменной по ее двоичному представлению по следующему алгоритму:

1. Ввести двоичный код в переменную `S` строкового типа.

2. Преобразовать `S` в вектор `V` типа «массив байт».

3. Привести `V` к заданному типу. Вывести значение.

4. Конец.

8. Обработать программой значения, полученные в результате выполнения пункта 4 задания. Сделать выводы.

Выполнение работы:

Задание 1:

Тип данных `long int`:

Абстрактный уровень представления СД:

1. Характер организованности – простейшие;
2. Изменчивость – статическая СД;
3. Набор допустимых операций: доступ, присваивание, инициализация, арифметические, логические, отношения, побитовые.

Физический уровень представления СД:

1. Схема хранения: последовательная.
2. Объем памяти, занимаемый экземпляром СД: 4 байта.
3. Формат внутреннего представления СД и способ его интерпретации:

Знаковый бит (ЗБ)	Биты значения
	4 байта

$N = 4$ байта

Способ интерпретации:

Переведем число, записанное в битах значения из ДК в ПК, если ЗБ = 1.

$$A = (-1)^{\text{ЗБ}} * 2^{N-1} - (1 * \overline{\text{ЗБ}})$$

4. Характеристика допустимых значений: -2147483648...2147483647
5. Тип доступа к элементам: прямой

Логический уровень представления СД:

1. Способ описания СД и экземпляра СД на языке С:
Способ описания СД: `long int/ long`;
Способ описания экземпляра: `long int a`;

Тип данных `double`:

Абстрактный уровень представления СД:

1. Характер организованности – простейшие;
2. Изменчивость – статическая СД;
3. Набор допустимых операций: доступ, присваивание, инициализации, отношения, арифметические, логические.

Физический уровень представления СД:

1. Схема хранения: последовательная.
2. Объем памяти, занимаемый экземпляром СД: 8 байта
3. Формат внутреннего представления СД и способ его интерпретации:

s = 1 бит	e = 11 бит	m = 54 бита
8 байт		

Способ интерпретации:

$V = (-1)^s * 2^{e-1023} * 1.m$, если $0 < e < 2047$

$V = (-1)^s * 2^{1022} * 0.m$, если $e = 0$ и $m \neq 0$

$V = (-1)^s$, если $e = 0$ и $m = 0$

$V = (-1)^s * Inf$, если $e = 2047$ и $m = 0$

$V = NaN$, если $e = 2047$ и $m \neq 0$

- Характеристика допустимых значений: $[1.797693 * 10^{-308}; 1.797693 * 10^{308}]$

Точность 15-16 знаков.

- Тип доступа к элементам: прямой

Логический уровень представления СД:

- Способ описания СД и экземпляра СД на языке С:

Способ описания СД: `double;`

Способ описания экземпляра: `double a;`

Тип данных {cat, dog, mouse, tiger}animal :

Абстрактный уровень представления СД:

- Характер организованности – последовательность;
- Изменчивость – статическая СД;
- Набор допустимых операций: доступ, присваивание, инициализация, арифметические, логические, отношения, побитовые

Физический уровень представления СД:

- Схема хранения: последовательная.
- Объем памяти, занимаемый экземпляром СД: 4 байта
- Формат внутреннего представления СД и способ его интерпретации:

0	1	...	N – 1
4 байта			

Способ интерпретации:

0 – первый элемент перечисляемого типа;

1] – второй элемент перечисляемого типа;

...

N – 1 – N-ый элемент перечисляемого типа

- Характеристика допустимых значений: именованные целочисленные константы

$1,0101001_2 * 2^3 = 1010,1001_2 = 10,5625_{10}$
исходное число: $10,5625_{10}$

3. enum {cat, dog, mouse, tiger} animal:

00000000 00000000 00000000 00000000₂ = 0 (cat)

00000000 00000000 00000000 00000001₂ = 1 (dog)

00000000 00000000 00000000 00000010₂ = 2 (mouse)

00000000 00000000 00000000 00000011₂ = 3 (tiger)

Задание 5:

Разработать и отладить программу, выдающую двоичное представление значений, заданных СД.

В программе использовать процедуры *PrintByte* и *PrintVar*.

```
#include <stdio.h>

// Выводит на экран монитора двоичное представление переменной "a" типа unsigned char
void PrintByte(unsigned char a)
{
    unsigned int size = sizeof(unsigned char) * 8;
    for(int i = 1<<(size-1); i > 0; i >= 1)
        putc(i & a ? '1' : '0', stdout);
}

//Выводит на экран двоичное представление переменной по указателю a произвольного
типа размера size байт
void PrintVar(void *a, unsigned int size)
{
    unsigned char *p = (unsigned char*)a+(size-1);
    for(int i = 0; i < size; i++, p--) {
        PrintByte(*p);
        printf(" ");
    }
}

int main()
{
    long int a = 1;

    printf("1 = ");
    PrintVar(&a, sizeof(long int));
    printf("\n");

    a = -1;
    printf("-1 = ");
    PrintVar(&a, sizeof(long int));
}
```



```

printf("\n");

double b = 1;

printf("1 = ");
PrintVar(&b, sizeof(double));
printf("\n");

b = -1;
printf("-1 = ");
PrintVar(&b, sizeof(double));
printf("\n");

b = -10.5625;
printf("-10.5625 = ");
PrintVar(&b, sizeof(double));
printf("\n");

enum animal {cat, dog, mouse, tiger};
enum animal one = cat;
printf("enum 0(cat) = ");
PrintVar(&(one), sizeof(int));
printf("\n");

one = dog;
printf("enum 1(dog) = ");
PrintVar(&(one), sizeof(int));
printf("\n");

one = mouse;
printf("enum 2(mouse) = ");
PrintVar(&(one), sizeof(int));
printf("\n");

one = tiger;
printf("enum 3(tiger)= ");
PrintVar(&(one), sizeof(int));
printf("\n");

return 0;
}

```

Задание 6:

Обработать программой значения, полученные в результате выполнения пункта 3 задания.

```

1 = 00000000 00000000 00000000 00000001
-1 = 11111111 11111111 11111111 11111111
1 = 00111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
-1 = 10111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
-10.5625 = 11000000 00100101 00100000 00000000 00000000 00000000 00000000 00000000
enum 0(cat) = 00000000 00000000 00000000 00000000
enum 1(dog) = 00000000 00000000 00000000 00000001
enum 2(mouse) = 00000000 00000000 00000000 00000010
enum 3(tiger)= 00000000 00000000 00000000 00000011

```

Вычисленные нами значения полностью совпали с значениями, выведенными на экран программой.

Задание 7:

Разработать и отладить программу, определяющую значение переменной по ее двоичному представлению по следующему алгоритму:

1. Ввести двоичный код в переменную S строкового типа.
2. Преобразовать S в вектор B типа «массив байт».
3. Привести B к заданному типу. Вывести значение.
4. Конец.

[illegible]

}

Вывод: был выполнен перевод чисел в двоичную систему счисления, результаты, полученные программно совпали с результатами полученными в ручную.