

Лабораторная работа №3
Структуры. Объединения. Перечисления.

Выполнил:

Студент 1-го курса

Группы ИВТ-1.1

Шардт Максим Александрович

Оглавление

Комплект 1: Структуры	4
Задание 1.1	4
Задание 1.2	5
Задание 1.3	7
Задание 1.4	9
Задание 1.5	10
Комплект 2: Объединения и перечисления	12
Задание 2.1	12
Задание 2.2	13
Задание 2.3	14
Задание 2.4	15

1. Комплект 1: Структуры

1.1. Создать некоторую структуру с указателем на некоторую функцию в качестве поля. Вызвать эту функцию через имя переменной этой структуры и поле указателя на функцию.

Математическая модель

Создается структура foo с полем, в котором хранится указатель на некую функцию bar. Функция вызывается через имя переменной этой структуры.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
ptr	Структура foo	Экземпляр структуры foo
result	Integer	Результат вызова функции bar

Код программы

```
1  #include <stdio.h>
2
3  struct foo
4  {
5      int (*bar)();
6  };
7  int func(int x)
8  {
9      return x + 2;
10 }
11 int main(void)
12 {
13     struct foo ptr = {&func};
14     int result = ptr.bar(2);
15     printf("result: %d\n", result);
16
17     return 0;
18 }
```

Листинг 1: Lab3 - 1-1.c

```
result: 4
```

Выводится значение 4, так как к изначальному значению 2 функция прибавляет 2.

1.2. Создать структуру для вектора в 3-х мерном пространстве. Реализовать и использовать в своей программе следующие операции над векторами:

- скалярное умножение векторов;
- векторное произведение;
- модуль вектора;
- распечатка вектора в консоли.

В структуре вектора указать имя вектора в качестве отдельного поля этой структуры.

Математическая модель

Формула скалярного произведения векторов:

$$a \cdot b = x_1 x_2 + y_1 y_2 + z_1 z_2$$

Формула векторного произведения векторов:

$$a \times b = \begin{vmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$
$$a \times b = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x)$$

Нахождение модуля вектора:

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
a	Vector	Вектор A
b	Vector	Вектор B
a_length	Double	Модуль вектора A

dot_product	Integer	Скалярное умножение векторов А и В
cross_product	Vector	Векторное умножение векторов А и В

Код программы

```

1  #include <math.h>
2  #include <stdio.h>
3
4  struct vector
5  {
6      int x;
7      int y;
8      int z;
9      char name[20];
10 };
11
12 int main(void)
13 {
14     struct vector a = {-1, 2, -3, "first"};
15     struct vector b = {0, -4, 1, "second"};
16     // Модуль вектора
17     double a_length = sqrt(a.x * a.x + a.y * a.y + a.z * a.z);
18     // Скалярное умножение
19     int dot_product = a.x * b.x + a.y * b.y + a.z * b.z;
20     // Векторное умножение
21     struct vector cross_product;
22     cross_product.x = a.y * b.z - a.z * b.y;
23     cross_product.y = a.x * b.z - a.z * b.x;
24     cross_product.z = a.x * b.y - a.y * b.x;
25     // Вывод вектора
26     printf("%s {%d, %d, %d}\n", a.name, a.x, a.y, a.z);
27     printf("%s {%d, %d, %d}\n", b.name, b.x, b.y, b.z);
28
29     printf("Dot product: %d\n", dot_product);
30     printf("Cross product: {%d, %d, %d}\n", b.x, b.y, b.z);
31
32     return 0;
33 }

```

Листинг 2: Lab3 - 1-2.c

Результат выполненной работы

```
first {-1, 2, -3}
second {0, -4, 1}
Dot product: -11
Cross product: {0, -4, 1}
```

Выводятся два изначальных вектора и результаты их скалярное и векторного умножения.

- 1.3. Вычислить, используя структуру комплексного числа, комплексную экспоненту $\exp(z)$ некоторого $z \in \mathbb{C}$

Математическая модель

$$\exp(z) = 1 + z + \frac{1}{2!}z^2 + \frac{1}{3!}z^3 + \dots + \frac{1}{n!}z^n.$$

Для возведения в степень комплексного числа используется формула:

$$z^n = r^n \cdot \cos n \varphi + r^n \cdot i \sin n \varphi, \text{ где}$$

$$r = \sqrt{a^2 + b^2}, \varphi = \arctg \frac{b}{a}$$

Структура комплексного числа состоит из двух элементов: вещественной и мнимой.

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
cNumber	complex	Изначальное комплексное число
n	Integer	Максимальная степень
exp	comlex	Экспоненциальная функция
fact	Integer	Факториал
arctgAB	Double	$\arctg(b / a)$
r	Double	Модуль комплексного числа

result	complex	Результат возвышения в степень комплексного числа
--------	---------	---

Код программы

```

1  #include <stdio.h>
2  #include <math.h>
3
4  struct complex
5  {
6      double real;
7      double imag;
8  };
9
10 int main(void)
11 {
12     int n = 12;
13     int fact = 1;
14     struct complex cNumber = {1.0, 2.0}; // 1 + 2i
15     struct complex exp = {cNumber.real + 1, cNumber.imag}; // 1 + z
16
17     for (int i = 2; i <= n; i++)
18     {
19         fact *= i;
20         double arctgAB = atan(cNumber.imag / cNumber.real);
21         double r = pow(sqrt(cNumber.real * cNumber.real + cNumber.imag *
cNumber.imag), i);
22         struct complex result = {r * cos(arctgAB * i), r * sin(arctgAB *
i)}};
23
24         exp.real += 1.0 / fact * result.real;
25         exp.imag += 1.0 / fact * result.imag;
26     }
27
28     printf("exp(%.2lf + %.2lfi) = %.5lf + %.5lfi\n", cNumber.real,
cNumber.imag, exp.real, exp.imag);
29
30
31
32     return 0;
33 }

```

Листинг 3: Lab3 - 1-3.c

Результат выполненной работы

$$\exp(1.00 + 2.00i) = -1.13120 + 2.47172i$$

Экспонента числа $1 + 2i$ равна $-1.13 + 2.47i$

- 1.4. Используя так называемые "битовые" поля в структуре C, создать экономную структуру в оперативной памяти для заполнения даты некоторого события, например даты рождения человека.

Математическая модель

Для записи дня требуется 5 бит (до 32), месяца 4 бита (до 16), а года 12 бит (до 4096).

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
date1	date	Первая дата
date2	date	Вторая дата
date3	date	Третья дата

Код программы

```
1 #include <stdio.h>
2
3 struct date
4 {
5     unsigned int day    : 5;
6     unsigned int month  : 4;
7     unsigned int year   : 12;
8 };
9
10 int main(void)
11 {
12     struct date date1 = {02, 02, 2002};
13     struct date date2 = {31, 11, 1999};
14     struct date date3 = {14, 02, 2011};
15     printf("date: %d.%d.%d\n", date1.day, date1.month, date1.year);
16     return 0;
17 }
```

Листинг 4: Lab3 - 1-4.c

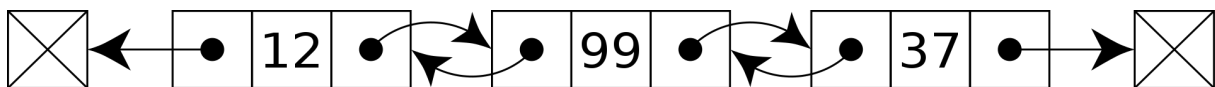
date: 2.2.2002

Выводится дата 02.02.2002.

- 1.5. Реализовать в виде структур двунаправленный связный список и совершить отдельно его обход в прямом и обратном направлениях с распечаткой значений каждого элемента списка.

Математическая модель

Графическое представление двунаправленного связного списка:



Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
head	ListNode	Указатель на первый элемент связного списка
tail	ListNode	Указатель на последний элемент связного списка
new_node	ListNode	Новый элемент
node	ListNode	Указатель на элементы списка для их последующего вывода

Код программы

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  struct ListNode
5  {
6      struct ListNode *pre;
7      struct ListNode *next;
8      int value;
9  };
10
11 int main(void)
12 {
```



```

13 // Начало списка
14 struct ListNode *head = (struct ListNode *)malloc(sizeof(struct
ListNode));
15 // Конец списка
16 struct ListNode *tail = head;
17
18 // Заполняем список значениями от 1 до 10
19 for (int i = 0; i < 10; i++)
20 {
21     struct ListNode *new_node = (struct ListNode
*)malloc(sizeof(struct ListNode));
22     // Новое значение
23     new_node->value = i;
24     // Указываем предыдущий элемент
25     new_node->pre = tail;
26     // Т.к. этот элемент последний, сл. элемент равен нулю
27     new_node->next = NULL;
28     // Обновляем указатель предыдущего элемента
29     tail->next = new_node;
30     // Обновляем конец списка
31     tail = new_node;
32 }
33 // Выводим значения списка
34 struct ListNode *node = head;
35 while (node != NULL)
36 {
37     printf("%d -> ", node->value);
38     node = node->next;
39 }
40 printf("NULL\n");
41 // Выводим значения списка в обратном порядке
42 node = tail;
43 while (node != NULL)
44 {
45     printf("%d -> ", node->value);
46     node = node->pre;
47 }
48 printf("NULL\n");
49
50 return 0;
51 }

```

Листинг 5: Lab3 - 1-5.c

Результат выполненной работы

```
0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> NULL
9 -> 8 -> 7 -> 6 -> 5 -> 4 -> 3 -> 2 -> 1 -> 0 -> NULL
```

Выводятся элементы списка в прямом и обратном порядке

2. Комплект 2: Объединения и перечисления

- 2.1. Напишите программу, которая использует указатель на некоторое объединение union.

Математическая модель



Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
decimalNumber	Integer	Целое число в myUnion
floatNumber	Float	Вещественно число в myUnion
a	myUnion	
b	myUnion	Указатель на a

Код программы

```
1 #include <stdio.h>
2
3 union myUnion
4 {
5     int decimalNumber;
6     float floatNumber;
7 };
8
```

```

9  int main(void)
10 {
11     union myUnion a;
12     union myUnion *b = &a;
13
14     a.decimalNumber = 10;
15     printf("before: %d, %f\n", a.decimalNumber, a.floatNumber);
16
17     b->floatNumber = 11.1;
18     printf("after: %d, %f\n", b->decimalNumber, b->floatNumber);
19
20     return 0;
21 }

```

Листинг 6: Lab3 - 2-1.c

Результат выполненной работы

```

before: 10, 0.000000
after: 1093769626, 11.100000

```

- 2.2. Напишите программу, которая использует union для побайтовой распечатки типа unsigned long.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
value	char	Значение внутри Byte
number	unsigned long	Изначальное число
byte	Byte	Юнион байт

Код программы

```

1  #include <stdio.h>
2
3  union Byte
4  {
5      char value;
6  };
7
8  int main(void)

```

```

9 {
10
11     unsigned long number = 1234567890123456789;
12     union Byte byte;
13     for (char *p = (char *)&number; p < (char *)&number + sizeof(unsigned
long); p++)
14     {
15         byte.value = *p;
16         printf("%hhx ", byte.value);
17     }
18     printf("\n");
19     return 0;
20 }

```

Листинг 7: Lab3 - 2-2.c

Результат выполненной работы

```
15 81 e9 7d f4 10 22 11
```

- 2.3. Создайте перечислимый тип данных (enum) для семи дней недели и распечатайте на экране его значения, как целые числа

Математическая модель

Семь дней недели будут отображаться числами от 0 до 6.

Список идентификаторов

<i>Имя переменной</i>	<i>Тип данных</i>	<i>Смысловое значение</i>
day	enum	Дни недели

Код программы

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     enum weekDays
6     {
7         MONDAY,
8         TUESDAY,
9         WEDNESDAY,

```

```

10     THURSDAY,
11     FRIDAY,
12     SATURDAY,
13     SUNDAY
14 } day = MONDAY;
15
16 for (int i = 0; i <= 6; i++)
17     printf("%d ", day + i);
18
19 printf("\n");
20 return 0;
21 }

```

Листинг 8: Lab3 - 2-3.c

Результат выполненной работы

```
0 1 2 3 4 5 6
```

- 2.4. Создайте так называемое размеченное объединение `union`, которое заключено в виде поля структуры `struct` вместе с ещё одним полем, которое является перечислением `enum` и служит индикатором того, что именно на текущий момент хранится в таком вложенном объединении. Создать и заполнить динамический массив таких структур с объединениями внутри, заполняя вспомогательное поле перечисления `enum` для сохранения информации о хранимом в каждом размеченном объединении типе данных. Реализовать распечатку данных массива таких структур в консоль.

Математическая модель

В структуре содержится два элемента: `union` и `enum`. В первом содержится либо целое число, либо вещественное, во втором тип первого. Элементы массива заполняются случайными значениями. И, в зависимости от случайного числа, имеют либо целый, либо вещественный тип.

Список идентификаторов

Имя переменной	Тип данных	Смысловое значение
size	Integer	Размер массива
array	myStruct	Массив структур

```
1 #include <stdlib.h>
2 #include <time.h>
3 #include <stdio.h>
4
5 struct myStruct
6 {
7     union myUnion
8     {
9         int decimal;
10        float floating;
11    } value;
12    enum
13    {
14        DECIMAL,
15        FLOATING
16    } type;
17 };
18
19 int main(void)
20 {
21     srandom(time(NULL));
22     const int size = 5;
23     struct myStruct *array = (struct myStruct *)malloc(sizeof(struct
myStruct) * size);
24
25     for (int i = 0; i < size; i++)
26     {
27         if (random() > RAND_MAX / 2)
28         {
29             array[i].type = DECIMAL;
30             array[i].value.decimal = random();
31         }
32         else
33         {
34             array[i].type = FLOATING;
35             array[i].value.floating = (float)RAND_MAX / (float)random();
36         }
37     }
38     for (int i = 0; i < size; i++)
39     {
40         if (array[i].type == DECIMAL)
```

```

41         printf("[%d]: %d\n", i, array[i].value.decimal);
42     else
43         printf("[%d]: %f\n", i, array[i].value.floating);
44 }
45
46 return 0;
47 }

```

Листинг 9: Lab3 - 2-4.c

Результат выполненной работы

[0]: 2.993201	[0]: 127887490
[1]: 3.766881	[1]: 1.456174
[2]: 8123010	[2]: 1.827014
[3]: 1.277884	[3]: 2.270920
[4]: 2.503270	[4]: 1.252753