

## Лабораторная работа №3

**Тема Лабораторной работы:** Структуры. Объединения. Перечисления.

### Задание 1.1

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

**Список идентификаторов:**

Имя	Тип	Смысл
function	integer	функция
a	integer	Параметр функции
aa	integer	Определитель функции-указателя в структуре MGD
Struct MGD	struct	объединить несколько переменных разных типов в одну структуру
avc	Struct MGD	доступ к переменным и функциям MGD

**Код программы:**

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  void function(int a) {
6      printf("%d\n", a*a*a);
7  }
8
9  struct MGD {
10     int (*aa)(int);
11 };
12
13 int main(void) {
14     struct MGD avc = {&function};
15     avc.aa(15);
16
17     return 0;
18 }
```

## Результат выполненной работы:

```
PS C:\Users\gnevn\OneDrive\Рабочий стол\учеба\РГПУ\Программирование\ЛР3\Codes\output> & .\'1.1.exe'  
3375
```

### Задание 1.2

**Постановка задачи:** 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.

### Список идентификаторов:

Имя	Тип	Смысл
struct vector	struct	Определяет x,y,z
x	int	Промежуточная переменная
y	int	Промежуточная переменная
z	int	Промежуточная переменная
A	vector	Координаты A
B	vector	Координаты B
SY	int	Скалярное произведение A и B
i	int	значение первой координаты вектора
j	int	значение второго координаты вектора
k	int	значение третьего координаты вектора
d	double	Длина вектора A
c	double	Длина вектора B

### Код программы:

```

1  # include <stdio.h>
2  # include <stdlib.h>
3  # include <math.h>
4
5  struct vector{
6      int x;
7      int y;
8      int z;
9  };
10
11 int main(void){
12     struct vector A= {1,2,3};
13     struct vector B= {0,5,10};
14
15     int SY = (A.x*B.x)+(A.y*B.y)+(A.z*B.z);
16     printf("%d\n", SY);
17
18     int i = (A.y*B.z)-(A.z*B.y);
19     int j = (A.x*B.z)-(A.z*B.x);
20     int k = (A.x*B.y)-(A.y*B.x);
21     printf("%d\t%d\t%d\n", i, j, k);
22
23     double d = sqrt(A.x*A.x+A.y*A.y+A.z*A.z);
24     printf("%f\n", d);
25
26     double c = sqrt(B.x*B.x+B.y*B.y+B.z*B.z);
27     printf("%f\n", c);
28
29     printf("{%d,%d,%d}\n", A.x, A.y, A.z);
30     printf("{%d,%d,%d}\n", B.x, B.y, B.z);
31     |
32     return 0;
33 }

```

**Результат выполненной работы:**

```

40
5      10      5
3.741657
11.180340
{1,2,3}
{0,5,10}

```

### Задание 1.3

**Постановка задачи:** Вычислить, используя структуру комплексного числа, комплексную экспоненту  $\exp(z)$  некоторого  $z \in \mathbb{C}$ :

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

### Математическая модель:

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

### Список идентификаторов:

Имя	Тип	Смысл
complex: typedef	double	структуры для комплексного числа с действительной и мнимой частями
add	complex	функция сложения комплексных чисел
multiply	complex	функция умножения комплексных чисел
exp_c	complex и int	функция вычисления экспоненты комплексного числа
z	complex	комплексное число
n	int	количество членов в ряде Тейлора
result	complex	результат вычислений в функции exp_c
term	complex	промежуточный результат вычислений в функции exp_c
i	int	счетчик цикла в функции exp_c

### Код программы:

```

1  ✓ #include <stdio.h>
2    #include <math.h>
3
4  ✓ typedef struct {
5      double real;
6      double imag;
7  } complex;
8
9  ✓ complex add(complex a, complex b) {
10     complex result;
11     result.real = a.real + b.real;
12     result.imag = a.imag + b.imag;
13     return result;
14 }
15
16 ✓ complex multiply(complex a, complex b) {
17     complex result;
18     result.real = a.real * b.real - a.imag * b.imag;
19     result.imag = a.real * b.imag + a.imag * b.real;
20     return result;
21 }
22
23 ✓ complex exp_c(complex z, int n) {
24     complex result = {1, 0};
25     complex term = {1, 0};
26     int i;
27     ✓ for (i = 1; i <= n; i++) {
28         term = multiply(term, z);
29         term.real /= i;
30         term.imag /= i;
31         result = add(result, term);
32     }
33     return result;
34 }
35
36 int main() {
37     complex z = {1, 2};
38     int n = 10;
39     complex exp_z = exp_c(z, n);
40     printf("exp(z) = %f + %fi\n", exp_z.real, exp_z.imag);
41     return 0;
42 }

```

**Результат выполненной работы:**

```
exp(z) = -1.131389 + 2.471766i
```

## Задание 1.4

**Постановка задачи:** Используя так называемые "битовые" поля в структуре C, создать экономную структуру в оперативной памяти для заполнения даты некоторого события, например даты рождения человека

**Список идентификаторов:**

Имя	Тип	Смысл
DMY	struct	определение структуры с именем DMY
day	int	Входная переменная
month	int	Входная переменная
year	int	Входная переменная

**Код программы:**

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct DMY{
5      int day;
6      int month;
7      int year;
8  };
9
10 int main(void){
11     struct DMY BD = {25,12,1993};
12     printf("Day, Month, Year: %d\t%d\t%d\n", BD.day, BD.month, BD.year);
13     return 0;
14 }
```

**Результат выполненной работы:**

```
Day, Month, Year: 25    12    1993
```

## Задание 1.5

**Постановка задачи:** Реализовать в виде структур двунаправленный связный список и совершить отдельно его обход в прямом и обратном направлениях с распечаткой значений каждого элемента списка.

**Список идентификаторов:**

Имя	Тип	Смысл
Node	struct	определение структуры
head	указатель на структуру Node	указатель на начало списка
tail	указатель на структуру Node	указатель на конец списка
i	int	Переменная цикла for
newNode	Указатель на struct	указатель на новый узел списка
current	Указатель на struct	указатель на текущий узел списка

## Код программы:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Node {
5      int data;
6      struct Node* next;
7      struct Node* prev;
8  };
9
10 int main() {
11     struct Node* head = NULL;
12     struct Node* tail = NULL;
13
14     for (int i = 1; i <= 5; i++) {
15         struct Node* newNode = malloc(sizeof(struct Node));
16         newNode->data = i;
17         newNode->next = NULL;
18         newNode->prev = tail;
19         if (tail) {
20             tail->next = newNode;
21         } else {
22             head = newNode;
23         }
24         tail = newNode;
25     }
26
27     for (struct Node* current = head; current; current = current->next) {
28         printf("%d ", current->data);
29     }
30     printf("\n");
31
32     for (struct Node* current = tail; current; current = current->prev) {
33         printf("%d ", current->data);
34     }
35     printf("\n");
36
37     return 0;
38 }

```

**Результат выполненной работы:**

```
1 2 3 4 5
5 4 3 2 1
```

### Задание 2.1

**Постановка задачи:** Напишите программу, которая использует указатель на некоторое объединение union.

**Список идентификаторов:**

Имя	Тип	Смысл
IF	union	Объединение
i	int	Входная переменная
f	int	Входная переменная
u	IF	Входная переменная
p	IF	Указатель на u

**Код программы:**

```
1  # include <stdio.h>
2  # include <stdlib.h>
3
4  union IF
5  {
6  int i;
7  float f;
8  };
9
10 int main(void)
11 {
12 union IF u;
13 u.i = 15;
14 printf("integer i:%d\n", u.i);
15
16 union IF *p = &u;
17 p->f = 3.14;
18 printf("float f:%f\n", p->f);
19
20 return 0;
21 }
```

**Результат выполненной работы:**



```
integer i:15
float f:3.140000
```

## Задание 2.2

**Постановка задачи:** Напишите программу, которая использует union для побайтовой распечатки типа unsigned long.

**Список идентификаторов:**

Имя	Тип	Смысл
ByteConverter	union	Объединение
num	unsigned long	хранение числа, которое будет преобразовано в массив байтов.
bytes	unsigned char	хранение представления числа в виде массива байтов.
i	int	переменная цикла
converter	ByteConverter	хранение числа и его представления в виде массива байтов

**Код программы:**

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  union ByteConverter {
5      unsigned long num;
6      unsigned char bytes[sizeof(unsigned long)];
7  };
8
9  int main() {
10     union ByteConverter converter;
11     converter.num = 0x123456789;
12
13     printf("Number: %lu\n", converter.num);
14     printf("Bytes: ");
15     for (int i = 0; i < sizeof(unsigned long); i++) {
16         printf("%02X ", converter.bytes[i]);
17     }
18     printf("\n");
19
20     return 0;
21 }

```

**Результат выполненной работы:**

```

Number: 591751049
Bytes: 89 67 45 23

```

### Задание 2.3

**Постановка задачи:** Создайте перечислимый тип данных (enum) для семи дней недели и распечатайте на экране его значения, как целые числа

**Список идентификаторов:**

Имя	Тип	Смысл
DOW	enum	перечисление
Mo	Int	элемент перечисления DOW
Tu	Int	элемент перечисления DOW
We	Int	элемент перечисления DOW
Th	Int	элемент перечисления DOW
Fr	int	элемент

		intперечисления DOW
Sa	Int	элемент перечисления DOW
Su	int	элемент перечисления DOW

### Код программы:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  enum DOW
5  {
6  Mo=1,
7  Tu=2,
8  We=3,
9  Th=4,
10 Fr=5,
11 Sa=6,
12 Su=7
13 };
14
15 int main(void)
16 {
17 printf("Monday %d\n", Mo);
18 printf("Tuesday %d\n", Tu);
19 printf("Wednesday %d\n", We);
20 printf("Thursday %d\n", Th);
21 printf("Friday %d\n", Fr);
22 printf("Saturday %d\n", Sa);
23 printf("Sunday %d\n", Su);
24
25 return 0;
26 }

```

### Результат выполненной работы:

```

Monday 1
Tuesday 2
Wednesday 3
Thursday 4
Friday 5
Saturday 6
Sunday 7

```

## Задание 2.4

**Постановка задачи:** 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.

**Список идентификаторов:**

Имя	Тип	Смысл

**Код программы:**

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  enum DataType {
5      INT,
6      FLOAT,
7      CHAR
8  };
9
10 union Data {
11     int i;
12     float f;
13     char c;
14 };
15
16 struct Log {
17     enum DataType type;
18     union Data data;
19 };
20 |
```

**Результат выполненной работы:**

Не удалось написать код