Структуры, объединения, перечисления

Комплект 1

Задача 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 переменной	Тип данных	Описание
sq()	int	Фукнция квадрата числа
f	struct Func	Структура, содержащая указатель на функцию
x	int	Аргумент функции

Код программы

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


int sq(int x){
    return x * x;
}

struct Func {
    int (*func)(int);
};

int main(void) {
    struct Func f = {sq};
    int x = 10;

    printf("\n%d^2 = %d\n\n", x, f.func(x));
```

```
    return 0;
}
```

Результат работы программы

```
10^2 = 100
```

Задача 1.2

Постановка задачи

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

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

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

Список идентификаторов

Имя переменной	Тип данных	Описание
a, b, c	struct Vector	Структуры векторов

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

struct Vector {
    double x, y, z;
};

double scalar_pr(struct Vector *a, struct Vector *b){
    return a->x * b->x + a->y * b->y + a->z * b->z;
}

void vector_pr(struct Vector *a, struct Vector *b, struct Vector *c){
    c->x = a->y * b->z - a->z * b->y;
    c->y = a->z * b->x - a->x * b->z;
    c->z = a->x * b->y - a->y * b->x;
}

double vector_mod(struct Vector *a){
    return sqrt(a->x * a->x + a->y * a->y + a->z * a->z);
}
```

```c
int main(void) {
    struct Vector a = {2, 3 ,6};
    struct Vector b = {1, 1.5 ,3};
    struct Vector c;

    printf("\na = (%.2f; %.2f; %.2f)", a.x, a.y, a.z);
    printf("\nb = (%.2f; %.2f; %.2f)\n", b.x, b.y, b.z);
    printf("\na · b = %.2f", scalar_pr(&a, &b));
    vector_pr(&a, &b, &c);
    printf("\na × b = c (%.2f; %.2f; %.2f)", c.x, c.y, c.z);
    printf("\n|a| = %.2f", vector_mod(&a));
    printf("\n|b| = %.2f\n\n", vector_mod(&b));

    return 0;
}
```

Результат работы программы

```
a = (2.00; 3.00; 6.00)
b = (1.00; 1.50; 3.00)

a · b = 24.50
a × b = c (0.00; 0.00; 0.00)
|a| = 7.00
|b| = 3.50
```

Задача 1.3

Постановка задачи

Вычислить, используя структуру комплексного числа, комплексную экспоненту $exp(z)$ некоторого $z \in C$:

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

Математическая модель

$$exp(z) = \sum_{k=0}^{\infty} \frac{z^k}{k!}$$

$$U_k = U_{k-1} \cdot M$$

$$M = \frac{z}{k}$$

$$S_0 = 1, \quad U_0 = 1$$

Список идентификаторов

Имя переменной	Тип данных	Описание
x, y	double	Целая и мнимая части числа
s	struct Complex	Сумма членов ряда
u	struct Complex	Текущий член ряда
k	int	Индекс члена ряда
z	struct Complex	Аргумент функции
e	struct Complex	Значение функции

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

struct Complex {
    double x, y;
};

struct Complex complex_exp(struct Complex *z) {
    struct Complex s = {1.0, 0.0};
    struct Complex u = {1.0, 0.0};
```

```
    int k = 1;

    do {
        struct Complex t = u;
        t.x = (u.x * z->x - u.y * z->y) / k;
        t.y = (u.x * z->y + u.y * z->x) / k;
        u = t;
        s.x += u.x;
        s.y += u.y;

        k++;
    } while (sqrt(u.x * u.x + u.y * u.y) > 1e-4);

    return s;
}

int main(void) {
    struct Complex z;

    printf("\nPlease enter complex number z\n");
    printf("real part: ");
    scanf("%lf", &z.x);
    printf("imaginary part: ");
    scanf("%lf", &z.y);

    struct Complex e = complex_exp(&z);
    printf("\nexp(%.4f + %.4fi) = %.4f + %.4fi\n\n", z.x, z.y, e.x, e.y);

    return 0;
}
```

Результат работы программы

```
Please enter complex number z
real part: 3
imaginary part: 4

exp(3.0000 + 4.0000i) = -13.1288 + -15.2008i
```

Задача 1.4

Постановка задачи

Используя так называемые "битовые" поля в структуре С, создать экономную структуру в оперативной памяти для заполнения данных некоего события, например дату рождения человека.

Список идентификаторов

Имя переменной	Тип данных	Описание
x, y	struct BirthDate	Даты рождения
day, month, year	unsigned int	День, месяц, гож
century	unsigned int	True — 21 век False — 20 век

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

struct BirthDate {
    unsigned int day: 5;
    unsigned int month: 4;
    unsigned int year: 7;
    unsigned int century: 1;
};

int main(void) {
    struct BirthDate x = {15, 2, 6, 1};
    struct BirthDate y = {3, 12, 99, 0};

    printf("\nBirth date X: %02d.%02d.%02d%02d", x.day, x.month,
x.century ? 20 : 19, x.year);
    printf("\nBirth date Y: %02d.%02d.%02d%02d\n\n", y.day, y.month,
y.century ? 20 : 19, y.year);

    return 0;
}
```

Результат работы программы

```
Birth date X: 15.02.2006
Birth date Y: 03.12.1999
```

Задача 1.5

Постановка задачи

Реализовать в виде структуры двунаправленный связный список и осуществить отдельно его обход в прямом и обратном направлениях с распечаткой значений каждого элемента списка.

Список идентификаторов

Имя переменной	Тип данных	Описание
f0, f1	struct Node *	Указатели на первые 2 ноды списка
pt	struct Node *	Указатель на текущую ноду
f	struct Node *	Указатель на создаваемую ноду

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define n    12


struct Node {
    int num;
    struct Node *prev;
    struct Node *next;
};

int main(void) {
    struct Node *f0 = (struct Node *)malloc(sizeof(struct Node));
    struct Node *f1 = (struct Node *)malloc(sizeof(struct Node));
    f0->next = f1;
    f1->prev = f0;
    f1->num = 1;

    struct Node *pt = f1;
    for (int i = 2; i < n; i++) {
        struct Node *f = (struct Node *)malloc(sizeof(struct Node));
        f->prev = pt;
        f->next = NULL;
        f->num = f->prev->num + f->prev->prev->num;
```

```
        pt->next = f;

        pt = f;
    }

    // Direct
    printf("\n");
    pt = f0;
    while (pt != NULL) {
        printf(pt->next == NULL ? "%d\n" : "%d, ", pt->num);
        pt = pt->next;
    }

    // Reverse
    pt = f0;
    while(pt->next != NULL) {
        pt = pt->next;
    }

    while (pt != NULL) {
        printf(pt->prev == NULL ? "%d\n\n" : "%d, ", pt->num);
        pt = pt->prev;
    }

    // Let's free memory
    pt = f0;
    while (pt != NULL) {
        free(pt);
        pt = pt->next;
    }

    return 0;
}
```

Результат работы программы

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
89, 55, 34, 21, 13, 8, 5, 3, 2, 1, 1, 0
```

Комплект 2

Задача 2.1

Постановка задачи

Написать программу, которая использует указатель на некоторое объединение union.

Список идентификаторов

Имя переменной	Тип данных	Описание
id	unit intdouble *	Указатель на структуру

Код программы

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


union intdouble {
    int i;
    double d;
};

int main(void) {
    union intdouble *id = (union intdouble *)malloc(sizeof(union
intdouble));

    id->i = 5;
    printf("\nint: %d", id->i);
    id->d = 5;
    printf("\nint: %.2f\n", id->d);
    printf("\nint: %d\n\n", id->i); // Не работает, т. к. заполнили поле
double d

    free(id);

    return 0;
}
```

Результат работы программы

```
int: 5
int: 5.00

int: 0
```

Задача 2.2

Постановка задачи

Написать программу, которая использует union для побайтовой распечатки типа unsigned long.

Список идентификаторов

Имя переменной	Тип данных	Описание
ln	union lnum	Структура
num	unsigned long	Число
b	unsigned char	Массив байт

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


union lnum {
    unsigned long num;
    unsigned char b[sizeof(unsigned long)];
};

int main(void) {
    union lnum ln;
    ln.num = 1234567890;

    printf("\n");
    for (int i =  0; i < sizeof(unsigned long); i++)
        printf("%d\n", ln.b[i]);
    printf("\n");

    return 0;
}
```

Результат работы программы

```
210
2
```

```
150
73
0
0
0
0
```

Задача 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 переменной	Тип данных	Описание
Week	enum	Дни недели

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


enum Week {mon, tue, wed, thu, fri, sat, sun};

int main(void) {

    printf("\nWeek:");
    for (int i = mon; i < 7; i++)
        printf("\n%d", i+1);
    printf("\n\n");

    return 0;
}
```

Результат работы программы

```
Week:
1
2
3
4
5
6
7
```

Задача 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 объединениями внутри, заполняя вспомогательно 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 переменной	Тип данных	Описание
Week	enum	Дни недели

Код программы

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


enum Tag {INT, DOUBLE};

union Num {
    int i;
    double d;
};

struct UnTag {
    union Num num;
    enum Tag tag;
};

int main(void) {
    struct UnTag *pa = NULL;
    int size = 10;

    // Выделим память
    pa = (struct UnTag *)malloc(sizeof(struct UnTag) * size);
    if (pa == NULL) {
```

```
        printf("Error :(\n\n");
        return EXIT_FAILURE;
    }

    // Заполним массив
    for(int i = 0; i < size; i++) {
        if (i % 2 == 0) {
            pa[i].num.i = i / 2;
            pa[i].tag = INT;
        } else {
            pa[i].num.d = (double)i / 2;
            pa[i].tag = DOUBLE;
        }
    }

    // Вывод массива
    printf("\nHere is your array :)\n[");
    for (int i = 0; i < size; i++)
        if (pa[i].tag == INT)
            printf(i == size - 1 ? "%d]\n\n" : "%d, ", pa[i].num.i);
        else
            printf(i == size - 1 ? "%.2f]\n\n" : "%.2f, ", pa[i].num.d);


    // Освободим память
    free(pa);

    return 0;
}
```

Результат работы программы

```
Here is your array :)
[0, 0.50, 1, 1.50, 2, 2.50, 3, 3.50, 4, 4.50]
```