Структуры

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    struct Date{
        int day;
        int month;
        int year;
    } dt;

    dt.day = 19;
    dt.month = 1;
    dt.year = 2023;

    Date dt2;
    dt2 = { 20, 1, 2023 };

    cout << dt.day + dt2.day << endl;

}
```

```
#include <iostream>
#include <string>
using namespace std;

struct Student
{
    int id;
    string name;
    int age;
    double grade;
};

int main()
{
    Student st[5];

    for (int i = 0; i < 5; i++)
    {
        cin >> st[i].id>>st[i].name>>st[i].age>>st[i].grade;
    }
```

```
    for (int i = 0; i < 5; i++)
    {
        cout << st[i].name << " -------------- " << st[i].grade << endl;
    }

}
```

```
#include <iostream>
#include <string>
using namespace std;

struct Student
{
    int id;
    string name;
    int age;
    double grade;
};

int main()
{
    Student st[5], max;

    for (int i = 0; i < 5; i++)
    {
        cin >> st[i].id>>st[i].name>>st[i].age>>st[i].grade;
    }

    max = st[0];

    for (int i = 0; i < 5; i++)
    {
        if (st[i].grade > max.grade)
        {
            max = st[i];
        }
    }
    cout << max.id << ' ' << max.name << "----------" << max.grade;
}
```

```
#include <iostream>
#include <string>
using namespace std;
struct Student
    {
        int id;
        string name;
```

```
        int age;
        double grade;
    };

void print_student(Student s){
    cout << s.id << ' ' << s.name << "----------" << s.grade;
}

int main()
{
    const int n=2;

    Student st[n], max;
    cout << "id\tname\tage\tgrade\n";
    for (int i = 0; i < n; i++)
    {
        cin >> st[i].id>>st[i].name>>st[i].age>>st[i].grade;
    }

    max = st[0];

    for (int i = 0; i < n; i++)
    {
        if (st[i].grade > max.grade)
        {
            max = st[i];
        }
    }
    print_student(max);
}
```

Какой йогурт покупать ?

```
#include <iostream>
#include <string>
using namespace std;

struct Yogurt
{
    string name;
    int mass;
    double price;
};

int main()
{
    Yogurt y[4]{
        {"Danone", 300, 16500},
        {"Actimel", 450, 19000 },
        {"Rastishka", 250 , 10000 },
        {"Essi", 300, 15000}
```

```
    }, max = y[0];
    for (int i = 1; i < 4; i++)
    {
        if (max.mass/max.price < y[i].mass/y[i].price)
        {
            max = y[i];
        }
    }
    cout << max.name << '\t' << max.mass << '\t' << max.price << endl;
}
```

Укозатели и структуры

```
#include <iostream>
#include <string>
using namespace std;

struct Yogurt
{
    string name;
    int mass;
    double price;
};

int main()
{
    int n;
    cout << "n="; cin >> n;
    Yogurt* y = new Yogurt[n];

    for (int i = 0; i < n; i++)
    {
        cout << i + 1 << " yogurt:\n";
        cin >> y[i].name >> y[i].mass >> y[i].price;
    }


}
```

Создадим совой язык программирование

```
#include <iostream>
#include <string>
using namespace std;

#define строка string     // переопределяет любую команду или символы
#define вывод cout
#define __ <<
```

```
#define конец endl
#define если if
#define иначе else

typedef long unsigned int LUI; // переопределяет тип

typedef struct
{
    строка name;
    int mass;
    double price;
}  Yogurt;

int main()
{
    setlocale(LC_ALL, "ru");
    LUI a = 3888323434;
    вывод __ a __ конец;
    Yogurt y{ "Actimel", 450 , 16999.99 };
    вывод __ y.name __ " " __ y.mass __ " " __ y.price __ конец;
    если(a % 2 == 0) {
        вывод __ "четная";
    }
    иначе{
        вывод __ "не четная";
    }
}
```

Сложные структуры

```
#include <iostream>
#include <string>
using namespace std;

struct Customer
{
    string name;
    string org;
    struct Date
    {
        int day;
        int moth;
        int year;
    } db;
    void get_balance() {
        cout << "Your balance: " << balance <<"$";
    }
    void set_balance(double money) {
        if (money > 0)
        {
            balance += money;
        }
```

```
        else
        {
            cout << "Incorrect\n";
        }
    }
    void spent_money(double money) {
        if (balance < money)
        {
            cout << "Not enough money";
        }
        else
        {
            balance -= money;
        }
    }
private:
    double balance = 0;
};

int main() {
    Customer c1;
    c1.name = "Anvar";
    c1.db={ 12,9,1998 };
    c1.org = "Isystem";
    c1.set_balance(100);
    c1.spent_money(34);
    c1.get_balance();
}
```

Перегрузка операторов

```
#include <iostream>
#include <string>
using namespace std;

struct Vector
{
    int x;
    int y;
    int z;

    inline Vector operator+(Vector a) {
        return { a.x + x, a.y + y , a.z + z };
    }
    inline Vector operator -(Vector a) {
        return { x - a.x, y - a.y , z - a.z };
    }
};

int main() {
    Vector V1, V2, sum;
    cout << "1 - Vector:\n\t";
```

```
    cin >> V1.x >> V1.y >> V1.z;
    cout << "2 - Vector:\n\t";
    cin >> V2.x >> V2.y >> V2.z;
    sum = V1 + V2;
    cout << "\nResul:\n\t";
    cout << sum.x << '\t' << sum.y << '\t' << sum.z << endl;
    //cout << V1.x + V2.x <<'\t'<< V1.y + V2.y<< '\t' << V1.z + V2.z;
}
```

Йогур + Йогурт = ?

```
#include <iostream>
#include <string>
using namespace std;

struct Yogurt
{
    string name;
    int mass;
    double price;

    inline Yogurt operator + (Yogurt y) {
        return { name[0] + y.name , mass + y.mass, price + y.price};
    }

    void print_y() {
        cout << name << " " << mass << " " << price << endl;
    }

};

int main() {
    Yogurt sum, y1{ "Actimel", 450, 1.7 }, y2{"Rastishka", 500, 2.1};
    sum = y1 + y2;
    sum.print_y();
}
```

Прегрузишиее операторов

```
#include <iostream>
#include <string>
using namespace std;

struct Vector
{
    int x;
    int y;
    int z;
    inline int operator*(Vector a) {
```

```
        return (x * a.x + y * a.y + z * a.z);
    }
    inline Vector operator*(int b) {
        return { x * b, y * b, z * b };
    }
};

int main() {
    Vector V1, V2, p;
    int sum;
    cout << "1 - Vector:\n\t";
    cin >> V1.x >> V1.y >> V1.z;
    cout << "2 - Vector:\n\t";
    cin >> V2.x >> V2.y >> V2.z;
    sum = V1 * V2;
    cout << "\nResul:\n\t";
    cout << sum << endl;
    //cout << V1.x + V2.x <<'\t'<< V1.y + V2.y<< '\t' << V1.z + V2.z;
    p = V1 * 5;
    cout << p.x << '\t' << p.y << '\t' << p.z;
}
```