

```

//_____file my.h
#include <stdio.h> //простой класс -- вектор с целыми координатами
#include<stdlib.h>
#define N 5 //размер вектора
class my
{
    int x[N],n;//n -- текущий размер вектора n<=N, он изменяется в декременте
    //память выделена нединамически, деструктор, конструктор копии в перегрузке не нужны
public:
    my( ); // конструктор без параметров
    my &operator --(void); // декремент справа
    my operator --(int); // декремент слева
    my operator +(const my &); // сложение покоординатное, слагаемые не изменяются
    my operator *(int); // умножение покоординатное слева
    friend my operator * (int,const my &); // умножение на число справа
    friend void pr(const my &); // печать вектора на экран
};

//_____file my.cpp
#include"my.h"
my::my()
{
    n=N; //память выделять не нужно, размер вектора =N
    for (int i=0;i<N;i++)
        x[i]=rand()%11; //заполняем координаты случайными числами
}
my my::operator +(const my &v)
{
    my tmp; tmp.n=(n<v.n?v.n:n); // размер суммы -- минимум из размеров слагаемых
    for (int i=0;i<n;i++)
        tmp.x[i]=x[i]+v.x[i];
    return tmp;
}
my operator * (int r, const my &v) // (int)*v

```

```

{
    return my(v)*r;
}

```

```

my my::operator * (int r) //v*(int)

```

```

{
    my tmp; tmp.n=n;
    for (int i=0;i<n;i++)
        tmp.x[i]=x[i]*r;

    return tmp;
}

```

```

my &my::operator --() // --v;

```

```

{
    n--; //сначала уменьшаем размер
    if(n<0)n=0;
    return *this;//возвращаем измененный объект
}

```

```

my my::operator --(int ) // v--;

```

```

{
    my tmp(*this);// сохраняем объект в tmp
    n--;if(n<0)n=0;//меняем размер у объекта *this
    return tmp;//возвращаем неизменный объект (который сохранили во временном tmp)
}

```

```

void pr(const my &v)

```

```

{
    for (int i=0;i<v.n;i++)
        printf("%d ",v.x[i]);

    printf("\n");
}

```

```
}  
  
//_____file main.cpp  
#include"my.h"  
  
int main()  
{  
  
    my v1,v2; // конструктор без параметров  
  
    pr(v1); // печать  
  
    v2= v1-- * (-1)+(1)* --v1+my()*0+0*my(); //присваивание +конструктор копии  
    //(неперегруженные)  
  
    pr(v1);  
  
    pr(v2);  
  
    return 0;  
  
}
```