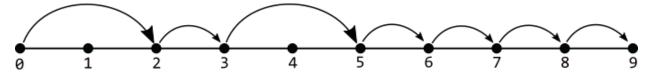
Семинар #6: Динамическое программирование. Классные задачи.

Задача о кузнечике:

Кузнечик находится в начале координата (точке x=0) и может прыгать вперёд на 1 шаг или на 2 шага. Найти количество разных путей которыми кузнечик может добраться из точки 0 до точки x=n. Пример одного из путей на рисунке:



Решение:

Пусть f(k) – это количество всевозможных путей от точки x=0 до точки x=k. Предположим, что мы знаем все f(k) до k=n-1. Как найти f(n)?

В точку n кузнечик может попасть из точки n-1 или из точки n-2. Число возможных путей до точки n равно сумме числа путей до точки n-1 и числа путей до точки n-2:

$$f(n) = f(n-1) + f(n-2)$$

Решение на языке С:

```
#include <stdio.h>
#define MAX 100

int main()
{
    int f[MAX] = {};
    int n;
    scanf("%d", &n);
    f[0] = 1;
    f[1] = 1;

    for (int i = 2; i <= n; ++i)
    {
        f[i] = f[i-1] + f[i-2];
    }
    printf("%d\n", f[n]);
}</pre>
```

Задачи:

- СуперКузнечик: Предположим, что кузнечик может прыгать на любое число шагов. Найдите количество разных путей которыми кузнечик может добраться из точки x=0 до точки x=n.
- **Кузнечик на плоскости:** Кузнечик находится на плоскости в точке (0,0) и может прыгать на 1 шаг вправо или на 1 шаг вверх. Найдите *количество разных путей* которыми кузнечик может добраться из точки (0,0) до точки (n,m).

• **Кузнечик на платной дороге:** Пусть кузнечик может прыгать на 1, 2 или 3 шага на прямой. Но в каждой точке с кузнечика взимается плата — некоторое целое число рублей. Плата в каждой точке задаётся в виде массива длины n + 1 (от 0 до n включительно). Кузнечику нужно добраться из точки x = 0 до точки x = n, заплатив как можно меньше. Найдите оптимальный путь кузнечика и количество рублей, которые должен он заплатить.

Например, для входа:

8 0 1 9 9 2 2 6 5 1

Оптимальная плата и оптимальный путь будут равны:

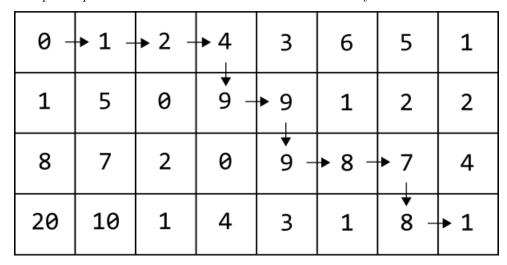
6 0 1 4 5 8

Протестируйте вашу программу на следующих тестах:

вход	выход
файл grasshopper_test_0.txt	6
	0 1 4 5 8
Φ айл grasshopper_test_1.txt	12
	0 3 5 8 10
Φ айл grasshopper_test_2.txt	266
	0 2 5 6 9 11 13 15 18 20 22 24 27 30 33 36 39 41 44 47 50
	53 56 59 62 64 66 69 72 75 78 80 82 85 87 90 92 95 98 100

Задача о черепашке:

Черепашка находится в вехнем левом углу прямоугольной таблицы. В каждой ячейке таблицы лежит определённое число кочанов капусты. Черепашка может двигаться либо вниз либо вправо по таблице на 1 шаг. Найдите такой путь, при котором черепашка съест наибольшее число кочанов капусты.



На вход задаче поступает файл с размером таблицы и самой таблицей. Нужно напечатать путь в виде строки из символов R (Right) и D (Down) и количество кочанов капусты на этом пути. Например, для таблицы на изображении программа должна выдать:

RRRDRDRRDR 58

Протестируйте вашу программу на следующих тестах:

вход	выход
файл turtle_test_0.txt	RRRDRDRRDR 58
Φ айл $turtle_test_1.txt$	DRRRRRRRDDDDDDD 244
ϕ айл turtle_test_2.txt	RRRRRRRRRRRDDRRDRRRRDDDRDRDDDRRDRRRRDDDRRRDDDD
	${\tt RRDRDDDDDRRDDRRRDRRRRRRRRRRRRRRRRRRRDDRRDDDD$
	DDDDDRDRRRRDDDDDDDDDDDRRRRDRDDDRRRRDRDDDRRRR

Подмассив максимальной суммы:

Напишите программу, которая принимает на вход массив и находит непрерывный подмассив, имеющий наибольшую сумму. Например, для массива:

 $\{5, -4, 6, -9, 7, 4, -9, 2, 4, 8, -7, 4, 3, 9, 5, -4, -3, -3, 7, -5\}$

подмассив максимальной суммы будет такой:

 $\{7, 4, -9, 2, 4, 8, -7, 4, 3, 9, 5\}$

Подсказка:

f(n) — в этой задаче — это сумма лучшего подмассива, который заканчивается в точке n. Также потребуется хранить start(n) — индекс первого элемента подмассива, который заканчивается в точке n.