В домашно № 4 по "Дискретни структури" беше дадена следната задача:

По колко начина може Дядо Коледа да раздаде 19 различни подаръка на 6 деца така, че всяко дете да получи поне два подаръка?

Задачата може да се реши чрез принципа за включване и изключване. Обаче има и втори начин — с помощта на рекурентно уравнение. А именно:

Нека $f(n; m)$ е броят на начините, по които Дядо Коледа може да раздаде m подаръка на n деца, където m и n са цели числа, $n \geq 1$, $m \geq 0$.

Нека Дядо Коледа е дал общо k подаръка на първите $n-1$ деца, а другите $m-k$ подаръка е дал на n-тото дете. Множеството от тези k подаръка може да се избере по C_m^k начина, а разпределянето им между първите $n-1$ деца може да стане по $f(n-1; k)$ начина съгласно с дефиницята на функцията f. Затова при всяка възможна стойност на k има $C_m^k . f(n-1; k)$ начина за раздаване на подаръците. Остава да сумираме по всички възможни стойности на k, а те са следните: 0, 1, 2, 3, ... , $m-2$ (това, че k е цяло неотрицателно число, е ясно от смисъла на k: брой раздадени подаръци на първите $n-1$ деца; а това, че $k \leq m-2$, следва от изискването, че на всяко дете, вкл. на n-тото, трябва да бъдат дадени поне два подаръка, значи $m-k \geq 2$, откъдето $k \leq m-2$).

Така получаваме следното рекурентно уравнение за функцията f:

$$\boxed{f(n; m) = \sum_{k=0}^{m-2} C_m^k . f(n-1; k) \quad \text{при всяко } n \geq 2 \text{ и всяко } m \geq 2.}$$

Ограниченията за допустимите стойности на аргументите се получават от следните съображения. Тъй като допустимите стойности на първия аргумент са $n \geq 1$, то това важи и за дясната страна, където имаме $n-1$ вместо n, затова $n-1 \geq 1$, откъдето $n \geq 2$. Колкото до втория аргумент, тъй като сумационният индекс k приема стойности от 0 до $m-2$, то трябва $0 \leq k \leq m-2$, тоест $0 \leq m-2$, откъдето следва, че $m \geq 2$.

Функцията f удовлетворява и следните *начални условия* (те покриват случаите, необхванати от рекурентното уравнение):

$\boxed{f(n; 0) = 0, \ f(n; 1) = 0 \text{ при всяко } n \geq 1}$, защото по-малко от два подаръка не могат да бъдат раздадени по никакъв начин, без да се наруши изискването, че всяко дете трябва да получи поне два подаръка;

$\boxed{f(1; m) = 1 \text{ при всяко } m \geq 2}$, защото при едно дете (и поне два подаръка) има само един начин за раздаване на подаръците: всички подаръци се дават на единственото дете.

Началните условия и рекурентното уравнение определят функцията f еднозначно. От това не следва, че функцията f може да се намери в явен вид, т.е. чрез нерекурентна формула. Не всички уравнения са решими в този смисъл. Дори когато са решими, това може да е невъзможно с известните ни методи (има и други, неизучени от нас, методи за решаване на рекурентни уравнения). Най-сетне, дори да можем да решим уравнението, това усилие може да се окаже ненужно.

Именно така стоят нещата в тази задача. Рекурентното уравнение е линейно, но е с променлив брой членове и коефициентите му не са постоянни. Затова то не може да се реши чрез характеристично уравнение. Това не е проблем, тъй като в задачата не се изисква явна формула за f, а само числен отговор: трябва да пресметнем $f(6; 19)$. Това може да стане чрез рекурентната зависимост, като започнем от началните условия и изчислим f за всички по-малки стойности на аргументите. Тъй като f е функция на два аргумента, удобно е да представим стойностите ѝ във вид на таблица от числа:

m \ n	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	1					
3	1					
4	1					
5	1					
6	1					
7	1					
8	1					
9	1					
10	1					
11	1					
12	1					
13	1					
14	1					
15	1					
16	1					
17	1					
18	1					
19	1					

Стойностите на f, попълнени в първата колонка и в първите два реда, се получават от трите начални условия.

По-нататък използваме рекурентното уравнение. Тъй като в лявата страна на уравнението първият аргумент е n, а в дясната страна е $n-1$, то следва, че стойностите във всяка колонка се получават с помощта на стойностите в съседната колонка отляво. Затова изчисляваме колонките отляво надясно, като стойностите във всяка отделна колонка изчисляваме отгоре надолу.

Например за $f(2;\ 2)$ заместваме $n=2,\ \ m=2$ в рекурентното уравнение:

$$f(2;\ 2)=\sum_{k=0}^{0} C_2^k\,.\,f(2-1;\ k)=C_2^0\,.\,f(1;\ 0)=1\,.\,f(1;\ 0)=1\,.\,0=0\,.$$

Аналогично, ако заместим $n=2,\ \ m=3$, ще получим

$$f(2;\ 3)=\sum_{k=0}^{1} C_3^k\,.\,f(2-1;\ k)=C_3^0\,.\,f(1;\ 0)+C_3^1\,.\,f(1;\ 1)=1\,.\,0+3\,.\,0=0+0=0\,.$$

Но ако заместим $n=2,\ \ m=4$, ще получим ненулева стойност: $f(2;\ 4)=$

$$\sum_{k=0}^{2} C_4^k\,.\,f(2-1;\ k)=C_4^0\,.\,f(1;\ 0)+C_4^1\,.\,f(1;\ 1)+C_4^2\,.\,f(1;\ 2)=1\,.\,0+4\,.\,0+6\,.\,1=6.$$

Попълваме пресметнатите стойности в таблицата (вж. жълтите клетки):

m \ n	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	1	0				
3	1	0				
4	1	6				
5	1					
6	1					
7	1					
8	1					
9	1					
10	1					
11	1					
12	1					
13	1					
14	1					
15	1					
16	1					
17	1					
18	1					
19	1					

Ясно е, че можем да продължим този процес неограничено: при $n=2$ заместваме m с 5, 6, 7, 8 и т.н. до 19 (вкл.), след което цялата втора колонка ще се окаже попълнена. След това пресмятаме стойностите в третата колонка, като при $n=3$ заместваме m с числата от 2 до 19 (вкл.). После попълваме четвъртата, петата и шестата колонка и таблицата е готова.

Описаната процедура очевидно не изисква никакви нови идеи, а значи и никакви интелектуални усилия, но за сметка на това е свързана с дълги и досадни сметки, които е по-добре да бъдат извършени от машина. За целта съставяме компютърна програма на някой от известните езици, например C.

```
#include <stdio.h>

double Combin(double n, double k) {
  // number of combinations,  0 <= k <= n
  for (double Result = 1 ; k > 0 ; k-- , n--)
    Result *= n/k;
  return Result;
}

void main() {
  double f[7][20]; // f[n][m] = f(n,m)

  // initial conditions:
  for (int n = 1; n <=  6; n++)
    f[n][0] = f[n][1] = 0;
  for (int m = 2; m <= 19; m++)
    f[1][m] = 1;

  // recurrence relation:
  for (n = 2; n <= 6; n++)
    for (m = 2; m <= 19; m++) {
      f[n][m] = 0;
      for (int k = 0; k <= m-2; k++)
        f[n][m] += Combin(m,k) * f[n-1][k];
    }

  // output:
  for (m = 0; m <= 19; m++) {
    for (n = 1; n <= 5; n++)
      printf("%.0f\t", f[n][m]);
    printf("%.0f\n", f[n][m]);
  }
}
```

Записваме първичния код във файл, напр. `GIFTS.CPP` и го компилираме до изпълнима програма `GIFTS.EXE`. След стартиране програмата ще отпечата цялата таблица. Разбира се, тя би могла да бъде съставена така, че да отпечатва само числото, което търсим. Кодът по-горе показва как може да се отпечата цяла таблица във вид, удобен за четене от други програми. Пресметнатите стойности се отпечатват в обикновен текстов формат (*plain text*); в рамките на един ред числата се разделят с табулации (*tab-delimited*). Този формат е удобен за четене от повечето известни софтуерни продукти, в т.ч. Microsoft Excel.

Ако стартираме програмата `GIFTS.EXE`, то тя ще отпечата таблицата върху стандартното изходно устройство (по подразбиране това е мониторът). Стандартният изход може да бъде пренасочен към произволно избран файл посредством инструкцията

```
GIFTS.EXE >RESULTS.TXT
```

на командния ред. В случая е избрано име на файла `RESULTS.TXT`. То може да бъде произволно. Ако файлът не съществува, командата ще създаде файл с това име. Ако файлът вече съществува, старото му съдържание ще бъде изтрито!

И така, след изпълнението на горната инструкция се появява файл с име `RESULTS.TXT` в текущата директория, който съдържа цялата таблица с числа в обикновен текстов формат; числата от всеки ред са разделени с табулации.

n / *m*	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	1	0	0	0	0	0
4	1	6	0	0	0	0
5	1	20	0	0	0	0
6	1	50	90	0	0	0
7	1	112	630	0	0	0
8	1	238	2940	2520	0	0
9	1	492	11508	30240	0	0
10	1	1002	40950	226800	113400	0
11	1	2024	137610	1367520	2079000	0
12	1	4070	445896	7271880	22869000	7484400
13	1	8164	1410552	35692800	196396200	194594400
14	1	16354	4390386	165957792	1454653200	2951348400
15	1	32736	13514046	742822080	9771762000	34205371200
16	1	65502	41278068	3234711480	61305644400	336151015200
17	1	131036	125405532	13803744864	365953988400	2954995243200
18	1	262106	379557198	58021888080	2104720417800	23983147188000
19	1	524248	1145747538	241116750624	11765939126760	183421913875200

Файлът `RESULTS.TXT` може да бъде отворен с всяка програма за работа с електронни таблици, например с Microsoft Excel. Резултатът ще изглежда по начин, подобен на показания в таблицата по-горе.

Числото, което ни интересува, се намира в долния десен ъгъл на таблицата: $\boxed{f(6;\,19) = 183\ 421\ 913\ 875\ 200}$. Това число е отговорът на задачата, тоест съществуват общо 183 421 913 875 200 начина, по които Дядо Коледа може да раздаде 19 различни подаръка на 6 деца така, че всяко дете да получи поне два подаръка.

Забележка: Чрез таблицата можем да открием и други закономерности (те обаче не са нужни за решението на задачата и не е задължително да бъдат описани в домашните работи). Например:

1) $\boxed{f(n;m) = 0 \text{ при } m < 2n}$. Това е естествено следствие от изискването всяко дете да получи поне два подаръка. За да бъде изпълнено това изискване, трябва броят на подаръците да бъде поне два пъти по-голям от броя на децата.
2) В граничния случай $m = 2n$ може да се изведе явна формула за f. Всяко дете ще получи точно два подаръка, остава само да уточним кои два подаръка. Това става с помощта на комбинации без повторение. Първото дете избира два подаръка от общо $2n$, второто дете избира два подаръка от останалите $2n-2$, третото дете избира два от останалите $2n-4$ подаръка и т.н. до последното дете, което "избира" два от последните два подаръка. От правилото за умножение следва, че
$f(n;2n) = C_{2n}^2 \,.\, C_{2n-2}^2 \,.\, C_{2n-4}^2 \,.\, C_{2n-6}^2 \ldots C_4^2 \,.\, C_2^2 =$
$= \dfrac{(2n)!}{2!\,(2n-2)!} \cdot \dfrac{(2n-2)!}{2!\,(2n-4)!} \cdot \dfrac{(2n-4)!}{2!\,(2n-6)!} \cdot \dfrac{(2n-6)!}{2!\,(2n-8)!} \cdot \ldots \cdot \dfrac{4!}{2!\,2!} \cdot \dfrac{2!}{2!0!}$;
множителите в числителя и знаменателя се съкращават с изключение на множителя 0! (но той тъй или иначе е равен на 1), $(2n)!$ и множителите $2! = 2$ (общо n на брой). Следователно $\boxed{f(n;2n) = \dfrac{(2n)!}{2^n}}$. Например при $n = 4$ имаме $f(4;\,8) = \dfrac{8!}{2^4} = \dfrac{1\,.\,2\,.\,3\,.\,4\,.\,5\,.\,6\,.\,7\,.\,8}{2\,.\,2\,.\,2\,.\,2} = \dfrac{7!}{2} = \dfrac{5040}{2} = 2520$.