# Задачи от минали устни и писмени изпити

* **Геометрия**

Да се провери дали точка принадлежи на следните фигури :

a)

(2, 0)

b)

(2, 0)

* **Формули**

1. Запишете на езика C++ следните математически формули:

а) ...........................................................................................................................

б) ...............................................................................................................................

в)sin(2x+4)+cos(x-1)-…………...........................................................

г)

1. Дадено е естествено число n. Да се напише програма, която намира сумата :
2. Нека n дадено цяло число n (n > 1). Да се напише програмен фрагмент, който намира стойността на израза:

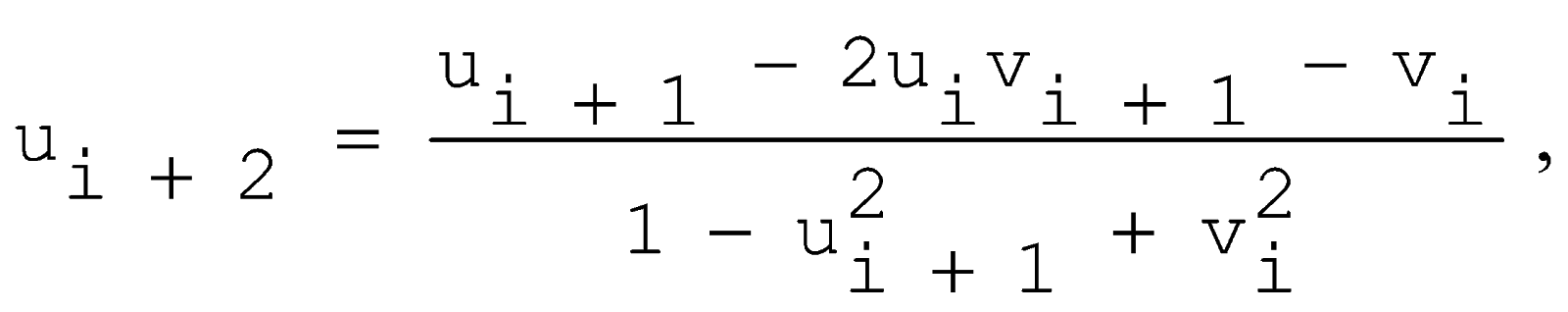
1.2 + 2.3.4 + 3.4.5.6 + … + n.(n + 1). … .(n + n).

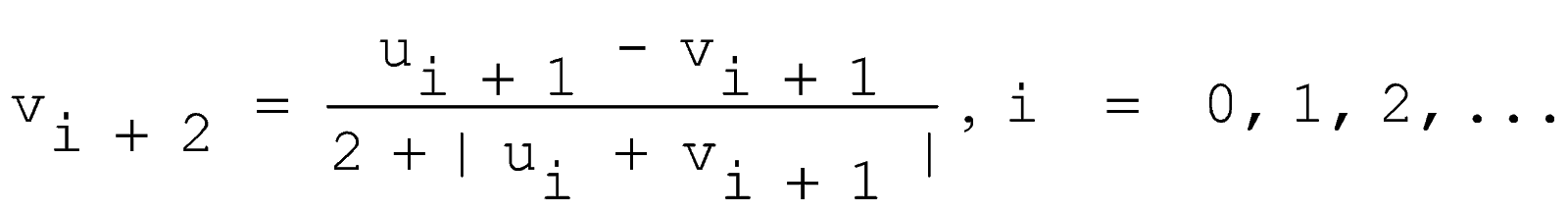
* **Редици**

1. Даден е едномерният масив от символни низове *a*0, *a*1, ..., *a*2*n*-1 (1 ≤ *n* ≤ 50). Елементите *a*0, *a*1, ..., *an*-1 са сортирани в низходящ ред, а елементите на *an*, an+1, ..., *a*2*n*-1 са сортирани във възходящ ред. Да се напише функция, която слива подредиците *a*0, *a*1, ..., *an*-1  и *an*, an+1, ..., *a*2*n*-1 в масива *b*0, *b*1, ..., *b*2*n*-1, който е сортиран в низходящ ред.
2. Да се напише булева функция, която проверява дали за редица от k.n числа е в сила свойството: първите k числа образуват строго монотонно растяща редица, вторите k числа образуват строго монотонно намаляваща редица, следващите k числа образуват строго монотонно растяща редица и т.н. За целта да се дефинират и използват помощни функции.
3. Редиците u0, u1, u2, … и v0, v1, v2, … са дефинирани по следния начин:

u0 = u1 = 0,

v0 = v1 = 1,

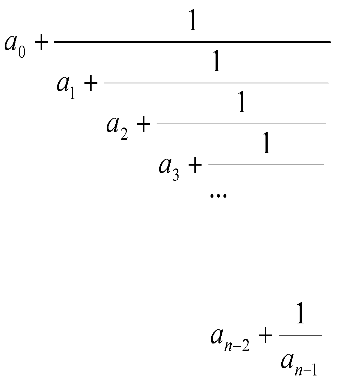




Да се напише програма, която намира u200 и v200.

* **Рекурсивни задачи**

1. Дасе дефинира **рекурсивна** функция, която намира позицията, в която трябва да се включи елемент в сортирана във възходящ ред редица, за да се запази сортировката. Редицата е представена чрез едномерен масив.
2. Да се дефинира **рекурсивна функция**, която проверява дали частта между i-я и j-я символ на символен низ е симетрична (дали е палиндром).
3. Дадена е матрица *A*[*n* x *n*] от цели числа.Да се дефинира **рекурсивна** функция, която прилага функцията *f* над елементите под главния диагонал и включително него  и функцията *g* – над елементите над главния диагонал. Функциите *f* и *g* са целочислени и се задават чрез указатели към функции.
4. Дасе дефинира **рекурсивна** функция, която намира стойността на израза:



1. Дасе дефинира **рекурсивна функция**, която въвежда от клавиатурата правилно записан израз от вида, зададен по-долу и намира стойността на израза:

<израз> ::= <цифра>| **f**(<израз>, <израз>)|**g**(<израз>, <израз>)

<цифра> ::= 0|1|…|9

а **f** и **g** са дефинирани по следния начин:

**f**(x, y) = (x2+y2)%10

**g**(x, y) = (3x+4y)%10

* **Backtracking**

1. Лабиринт е представен с булева квадратна матрица An x n (n > 1). Клетката (i, j) е или проходима (ai,j е истина), или е непроходима (ai,j е лъжа). В непроходима клетка може да се влезе, но от нея не може да се излезе. Да се напише програма, която проверява дали съществува път от съседни в диагонално направление проходими клетки на лабиринта, който започва от даденa клетка на лабиринта и завършва в долния му десен ъгъл. В случай, че път съществува програмата да намира и извежда координатите на клетките, през които пътят преминава, както и дължината му (броя на клетките от началната до крайната).

* **Матрици**

1. Да се напише програма, която проверява дали в частта над вторичния главен диагонал на квадратна матрица от естествени числа съществува число на Фибоначи.
2. Да се напише програма, която в дадена правоъгълна целочислена матрица *a* проверява дали съществува стълб, в който всеки елемент е равен на сумата от индексите си (*aij*=*i*+*j*).
3. Дадена е квадратна мрежа от клетки, всяка от които е празна или запълнена. Запълнените клетки, които са свързани, т.е.  имат съседни в хоризонтално, вертикално или диагонално направление, образуват област. Да се напише програма, която намира броя на областите и размера (в брой клетки) на всяка област.

* **Низове**

1. Дадена е правоъгълна таблица от думи, представени чрез символни низове. Да се намери изречението, което се образува след последователното конкатениране на думите, обхождайки таблицата по редове, започвайки от долния ред и отдясно, наляво.
2. Да се напише функция, която замества всички срещания на даден низ *what* в даден друг низ *where* с толкова на брой символи „\*“, колкото е дължината на *what*. Функцията да връща указател към последния заместен символ в *where* или NULL, ако няма такъв, както и броя на срещанията на *what* в *where*, които са заместени със звездички.