

Clasa a V-a

Problema 1 (100 puncte)

Săritura cangurului

A fost odată ca niciodată, a fost un cangur care creștea ca Făt Frumos din poveste, într-un an precum alții în zece. Într-o zi a început să facă sărituri. Și a sărit pentru început 7 metri. A doua zi a sărit, în plus față de ziua precedentă, de zece ori mai mult. În a treia zi a reușit să sară, în plus față de prima zi, de zece ori mai mult decât în ziua a doua. În a patra zi a sărit, în plus față de prima zi, de zece ori mai mult decât în ziua a treia. Și tot așa mai departe.

Cerință:

Scrieți un program care calculează câți metri a sărit cangurul, în total, în **n** zile.

Date de intrare:

Se citește de la tastatura valoarea lui **n**.

Date de ieșire:

Se va afișa pe ecran câți metri a sărit cangurul, în total, după cele **n** zile.

Restricții:

n este număr natural, strict mai mic ca 12.

Exemplu:

Pentru $n=3$ se va afișa: 861 metri.

Clasa a V-a

Problema 2 (100 puncte)

Numere prime

Se numește număr prim, un număr care este divizibil doar cu 1 și cu el însuși. Astfel în intervalul $[1, 30]$ numerele prime vor fi: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, în total 10 numere prime.

Notă: Numărul 1 nu este considerat număr prim!

Cerință:

Dându-se două numere n, k să se determine $2 \cdot k$ numere prime situate în centrul listei numerelor prime din intervalul $[1, n]$, în cazul în care în interval este un număr par de numere prime și $2 \cdot k - 1$ numere din centrul listei de numere prime, în cazul în care numărul de numere prime este impar.

Dacă numărul $2 \cdot k$ (respectiv $2 \cdot k - 1$) este mai mare decât numărul de numere prime din intervalul considerat, atunci se vor afișa toate numerele prime din interval.

Date de intrare:

De la tastatură se vor citi două numere n și k , cu semnificația:

n = marginea superioară a intervalului din care se determină numerele prime ;

k = are semnificația din enunț.

Date de ieșire:

Se vor afișa pe ecran cele $2 \cdot k$ sau $2 \cdot k - 1$ numere cerute, separate prin spațiu

Restricții:

$$1 \leq n \leq 10000$$

$$1 \leq k \leq 30$$

$$k \leq n$$

Exemple:

Date introduse de la tastatură:	Date afișate pe ecran:
21 2	5 7 11 13
18 2	5 7 11
18 18	2 3 5 7 11 13 17
100 7	17 19 23 29 31 37 41 43 47 53 59 61 67

Timp maxim de executare 1 sec/test

Clasa a V-a

Problema 3 (100 puncte)

Poarta Orintiei

Copa bate la poarta Orintiei, dar poarta e programată să nu se deschidă decât după ce se introduc, într-o casetă cu s spații ($3 \leq s \leq 10$), s cifre strigate de portar. Portarul a strigat: „1”, Copa a butonat 1, în primul spațiu de la stânga la dreapta. Portarul a strigat: „0”, și, în timp ce Copa butona 0 în spațiul al doilea, 1 a devenit 2 în spațiul anterior. Portarul a strigat: „7”. Copa scria 7 în spațiul al treilea, iar în primul spațiu, 2 devine 3, iar în al doilea spațiu, 0 devine 1. Și tot așa, până la al s -lea spațiu, când Copa reușește să scrie toate cifrele și apare tot codul. Și poarta se deschise, dar... surpriză, mai era o poartă, iar codul acesteia, N , era cel mai mic număr format din cât mai multe dintre cifrele codului anterior, astfel încât nici o cifră să nu se repete.

Notă: după 9 urmează 0.

Cerință:

Disperat de atâta informatizare, Copa, umil cetățean al Orintiei vă cere sprijinul să *calculați cel de-al doilea cod N* .

Date de intrare:

De la tastatură se introduc:

s , pe prima linie

$a(1)$ → cele s cifre, pe următoarele s linii

$a(2)$

$a(3)$

...

$a(s)$

Date de ieșire:

Pe ecran, se va afișa N

Exemplu:

Pentru datele de intrare

10

1

0

7

9

7

3

6

9

4

6

se va afișa:

102456789

Observatie:

Dacă se vor afișa cifrele lui N cu un spațiu între ele, atunci se va pierde 1/5 din punctajul fiecărui test.

Timp de executare: 1 s/test