Pregătire pentru examenul de Bacalaureat şi examenul de admitere la Facultatea de Matematică şi Informatică

Probleme propuse pentru sesiunea 1

16 decembrie 2017

INFORMATICA

Tema 1: Algoritmi de prelucrare a datelor simple:

- Prelucrări asupra cifrelor (extragerea şi analiza cifrelor unui număr natural)
- Proprietăţi de divizibilitate (verificare divizibilitate, determinarea celui mai mare divizor comun, verificare şi generare numere prime, descompunere în factori primi)
- Calcule de sume şi produse
- Generare şiruri date prin relaţii de recurenţă

1. Fie *n* un număr natural nenul. Descrieţi algoritmi pentru:
    a. Determinarea sumei tuturor cifrelor lui *n*. De exemplu, pentru *n=26326* se obţine valoarea *19*.
    b. Determinarea valorii obţinute prin inversarea cifrelor numărului n. De exemplu, pentru valoarea *26326* se obţine valoarea *62362*.
    c. Determinarea tuturor cifrelor binare ale lui *n*.
    d. Determinarea tuturor divizorilor proprii ai lui *n*.
    e. Afișarea mesajului „da” dacă cea mai semnificativă cifră a unui număr n este strict mai mare decât cifra unităţilor sau mesajul „nu” în caz contrar. De exemplu, pentru *n=5832* se va afişa „da” pentru că *5>2*, iar pentru *n=4539* se va afişa „nu”.

2. *Cifra destinului*. Cifra destinului este cifra obţinută prin adunarea cifrelor ce intervin în data naşterii; adunarea cifrelor rezultatului obţinut se repetă până se ajunge la o singură cifră. De exemplu, pentru 01.02.1999 se obţine: 1+2+1+9+9+9=31 → 3+1=4. Descrieţi un algoritm care calculează cifra destinului pornind de la data naşterii specificată prin cele trei valori (zi, luna, an).

3. *Numere asemenea*. Două numere naturale sunt asemenea dacă scrierile celor două numere în baza *10* au aceleași cifre. De exemplu, numerele *23326* și *623* sunt asemenea, deoarece mulțimea cifrelor este aceiași (*{2,3,6}*). Descrieţi un algoritm care preia o pereche de numere naturale şi determină dacă sunt sau nu asemenea.

4. *Conversie între baze de numeraţie*. Se consideră că numărul natural nenul *n* este reprezentat în baza de numeraţie *b1* ($2\leq b1 \leq 10$) şi se doreşte construirea numărului natural *m* care reprezintă aceeaşi valoare însă în baza *b2* ($2\leq b2 \leq 10$). Atât numărul iniţial cât şi cel convertit sunt specificate prin variabile întregi (nu este posibilă utilizarea unui tablou pentru stocarea cifrelor). De exemplu pentru *n=2210* şi *b1=3*, *b2=5* se obţine *m=300*.

5. *Reguli de codificare*. Se consideră un număr natural *n* constituit din $k\geq 2$ cifre ($n=c_k c_{k-1}\ldots c_1$) şi se pune problema codificării lui *n* prin schimbarea poziţiei unor cifre fără a stoca cifrele lui n într-un tablou. Descrieţi algoritmi pentru următoarele două variante de codificare:

    a. *Permutare circulară a cifrelor*. Fiind dată o valoare *p* ($1\leq p\leq k-1$), se să construiască numărul $m=c_p c_{p-1}.. c_1\ c_k c_{k-1}\ldots\ c_{p+1}$. De exemplu pentru *n=45612* şi *p=2* se obţine *m=12456*.

Blvd. V. Pârvan 4, 300223, Timişoara, România
Tel: +40-(0)256-592.155
Email: mateinfo@e-uvt.ro
www.math.uvt.ro

b. *Interschimbare a două cifre*. Fiind date două valori *p1* şi *p2* ($1 \le p2 \le p1 \le k-1$) să se construiască numărul obţinut prin interschimbarea cifrelor de pe poziţiile indicate de *p1* şi *p2* (din $n=c_k c_{k-1} \ldots c_{p1} \ldots c_{p2} .. c_1$ se obţine $m= c_k c_{k-1} \ldots c_{p2} \ldots c_{p1} .. c_1$), toate celelalte cifre rămânând pe poziţia lor iniţială. De exemplu pentru *n=45612*, *p1=4*, *p2=2* se obţine *m=41652*.

6. Pentru un număr natural *n*, să se scrie un algoritm care determină numărul maxim de divizori pe care îi are un număr din mulţimea *{2,3,…,n}*. De exemplu, dacă *n=15*, numărul maxim de divizori este 5 (numărul 12 are 5 divizori: 2,3,4,6,12).

7. Descrieţi un algoritm care pentru două numere naturale nenule a şi b determină numitorul şi numărătorul fracţiei ireductibile egale cu a/b.

8. Să se descrie un algoritm care verifică dacă un număr *n* este prim sau nu folosind teorema lui Wilson. (Condiția necesară și suficientă ca un număr natural *p*, *p>1*, să fie prim este ca *(p-1)!+1* să fie divizibil cu *p*.)

9. Să se descrie un algoritm care afişează toate numerele prime dintr-un interval dat *[a,b]*, *a*, *b* numere naturale, *a<b*.

10. Se consideră un natural n>2. Descrieţi un algoritm care să afişeze toate numerele *x* mai mici decât n, care au proprietatea că *x-1* şi *x+1* sunt numere prime. Exemplu: pentru n=43, se vor afişa numerele: 4 6 12 18 30 42.

11. Scrieți un algoritm care afişează descompunerea unui număr natural *n* în factori primi. De exemplu pentru *n=18*, se va afişa *2^1x3^2*.

12. Să se scrie un algoritm care pentru numărul natural *n* dat, determină sumele:

    a. S=1*2+2*3+3*4...+n*(n+1)
    b. S=1*n+2*(n-1)+3*(n-2)+...+n*1

13. Estimaţi cu precizia ε>0 limitele șirurilor următoare (pentru punctele a. şi b. se consideră că *x* este o valoare dată din intervalul (0,1)):

    a. $s_n = \sum_{i=0}^{n} \frac{(-1)^i x^{2i+1}}{(2i+1)!}$
    b. $s_n = \sum_{i=0}^{n} \frac{(-1)^i x^{2i}}{(2i)!}$
    c. $s_n = \sum_{i=1}^{n} \frac{1}{(2i+1)^2}$

    Indicaţie. Calculul se opreşte atunci când diferenţa dintre doi termeni consecutivi ai șirului este mai mică decât constanta ε ($|s_n - s_{n-1}| < \varepsilon, \varepsilon = 0.001$).

14. Să se scrie un algoritm care determină dacă un număr natural dat *m* face parte din șirul lui Fibonacci sau nu. Șirul lui Fibonacci este definit prin relațiile: *f(0)=1*, *f(1)=1*, *f(n)=f(n-1)+f(n-2)*, pentru *n>=2*. Nu se vor folosi tablouri.

Blvd. V. Pârvan 4, 300223, Timişoara, România
Tel: +40-(0)256-592.155
Email: mateinfo@e-uvt.ro
www.math.uvt.ro