

Informatică
Barem de evaluare și de notare
Sesiunea iulie 2017

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem.
- Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea la 10 a punctajului total acordat pentru lucrare.

SUBIECTUL I (30 puncte)

1.		c)	5p	
2.	a)	3	5p	Observație: Algoritmul afișează <i>cmmdc</i> al numerelor citite.
	b)	9 9	5p	- orice soluție care verifică condiția de <i>cmmdc</i> – 3p - soluția corectă – 5p
	c)	Pentru algoritm pseudocod corect: -echivalența prelucrării realizate -corectitudinea instrucțiunii repetitive -corectitudinea globală a algoritmului (1)	5p 3p 1p 1p	
	d)	Pentru program corect: -declarare variabile -citire date -afișare date -instrucțiuni de decizie corecte -instrucțiuni repetitive corecte -corectitudine globală a programului (1)	10p 1p 1p 1p 4p 2p 1p	

SUBIECTUL II (30 puncte)

1.		b)	5p	
2.		Pentru rezolvare corectă: -citirea numărului și inițializarea corectă a variabilelor -algoritmul de verificare că numărul este palindrom -afișarea rezultatului -corectitudinea globală a programului (1)	10p 1p 7p 1p 1p	Se poate testa că numărul este palindrom prin două metode: 1) prin crearea numărului invers (cât timp n este diferit de zero, $invers = invers * 10 + \text{ultima cifră a numărului } n$, apoi se elimină ultima cifră a lui n) și compararea acestuia cu numărul inițial. În caz de egalitate n este palindrom altfel n nu este palindrom.

			2) prin repetarea comparării dintre cea mai semnificativă și cea mai puțin semnificativă cifră a numărului n , până când cele două cifre diferă, sau până când s-a efectuat numărul suficient de comparații. În primul caz n nu este palindrom altfel n este palindrom. Cele două cifre simetrice care sunt egale se vor elimina din număr.	
3.	a)	Pentru rezolvare corectă: -citire n și citirea elementelor matricei -determinarea numărului de parcurgeri de pătrate -parcursere linie de la stânga la dreapta -parcursere coloană de sus în jos -parcursere linie de la dreapta la stânga -parcursere coloană de jos în sus -corectitudinea globală a algoritmului (1)	11p 1p 1p 2p 2p 2p 2p 1p	Se observă că avem de parcurs în sens orar $n/2$ (pentru n par), respectiv $(n - 1)/2 + 1$ (pentru n impar) pătrate de forma: $ \begin{array}{ccc} a_{k,k} & \dots & a_{k,n-k+1} \\ \vdots & & \vdots \\ a_{n-k+1,k} & \dots & a_{n-k+1,n-k+1} \end{array} $ Operația de parcursere a pătratului k implică: - parcurserea laturii de sus a pătratului de la stânga la dreapta - parcurserea laturii din dreapta a pătratului de sus în jos, - parcurserea laturii de jos a pătratului de la dreapta la stânga, - parcurserea laturii din stânga a pătratului de jos în sus
	b)	Pentru răspuns corect: -menționarea rolului variabilelor utilizate (*) -date de intrare identificate corect -date de ieșire identificate corect	4p 2p 1p 1p	(*) Se acordă numai 1p dacă se omite explicarea unor variabile.

SUBIECTUL III (30 puncte)

1.	$2 * m + v [i+1-j]$ -pentru termenul $2 * m$ -pentru termenul $v [i+1-j]$ -pentru suma termenilor	5p 2p 2p 1p	Observație: Bucla „while” a secvenței de program realizează conversia în baza 2 a numărului n . Cifrele binare se memorează în tabloul v , cifra de rang minim la indicele 1, cifra de rang următor la indicele 2, ..., respectiv cifra de rang maxim la indicele i . Variabila i memorează numărul de cifre binare ale lui n . Primul „for” realizează o permutare circulară la stânga cu o poziție a elementelor $v[1], v[2], \dots, v[i]$.
----	--	-----------------------------	---

			Al doilea „for” parcurge elementele vectorului v începând cu rangul $i - 1$ în vederea determinării valorii zecimale a numărului binar reprezentat de cifrele binare ale lui v în variabila m . Se afișează m ori de câte ori cifrele de rang curent și rang curent plus 1 nu sunt ambele 0.
2.	18 -determinarea numerelor Fibonacci mai mici ca 2017 -determinarea indicilor numerelor Fibonacci mai mici ca 2017 -determinarea soluției f_{18}	5p 2p 2p 1p	Observație: Numerele lui Fibonacci până la întâlnirea primului număr mai mare ca 2017 sunt: $f_1 = 1$ $f_2 = 1$ $f_3 = 2$ $f_4 = 3$ $f_5 = 5$ $f_6 = 8$ $f_7 = 13$ $f_8 = 21$ $f_9 = 34$ $f_{10} = 55$ $f_{11} = 89$ $f_{12} = 144$ $f_{13} = 233$ $f_{14} = 377$ $f_{15} = 610$ $f_{16} = 987$ $f_{17} = 1597$ $f_{18} = 2584$
3.	Pentru program corect -citirea datelor de intrare -determinarea corectă a valorii maxime a x_i -determinarea corectă a valorii minime a y_i -realizarea corectă a testului de existență a unei soluții -declararea tuturor variabilelor, -corectitudinea globală a programului (1)	10p 1p 3p 3p 1p 1p 1p	Rezolvarea constă în găsirea celei mai mari valori mx dintre capetele stânga ale intervalelor x_i , găsirea celei mai mici valori my dintre capetele drepte ale intervalelor y_i și compararea mx cu my . Dacă $mx \leq my$ atunci există soluții și răspunsul este „DA”, altfel răspunsul este „NU”.
4.	Pentru program corect -citirea fișierului de intrare -scrierea buclei pentru procesarea numerelor din șirul de intrare -localizarea subșirului maxim de elemente egale -localizarea subșirului maxim de elemente distincte -scrierea corectă a subșirului maxim de elemente egale	10p 2p 3p 1p 1p 1p	Se citesc toate numerele din fișierul de intrare într-un tablou v . Pentru determinarea celui mai lung subșir de elemente succesive egale se parcurge vectorul începând cu prima poziție $i = 1$. Apoi se verifică subșirul curent începând cu $j = i$ până când se găsește un element diferit de precedentele. Se compară

	<p>-scrierea corectă a subșirului maxim de elemente distincte</p> <p>-declararea tuturor variabilelor, corectitudinea globală a programului</p>	<p>1p</p> <p>1p</p>	<p>lungimea noului subșir găsit și se memorează poziția sa de început și lungimea dacă lungimea este strict mai mare decât maximul curent. Se trece apoi la subșirul următor începând cu poziția j. Pentru determinarea celui mai lung subșir de elemente succesive astfel încât oricare două sunt diferite, se parcurge vectorul începând cu prima poziție $i = 1$. Apoi se verifică subșirul curent începând cu $j = i$ până când se găsește un element egal cu unul dintre precedentele pe poziția k. Se compară lungimea noului subșir găsit și se memorează poziția sa de început și lungimea dacă lungimea este strict mai mare decât maximul curent. Se trece apoi la subșirul următor începând cu poziția $k + 1$ (mai eficient) sau cu poziția $i + 1$ (mai ineficient, dar corect). În toate situațiile se testează ca subșirul curent analizat cu indici de la i la j să nu depășească tabloul v cu indici de la 1 la n.</p>
--	---	---------------------	---

(1) Corectitudinea globală vizează structura, sintaxa, alte aspecte de corectitudine neprecizate în barem.