

Examenul național de bacalaureat 2025
Proba E. d)
Informatică

BAREM DE EVALUARE ȘI DE NOTARE
Limbajul C/C++

Simulare

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

- Se punctează orice modalitate de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit în barem. Nu se acordă fracțiuni de punct. Se acordă zece puncte din oficiu. Nota finală se calculează prin împărțirea la zece a punctajului total acordat pentru lucrare.
- Rezolvările de mai jos sunt recomandări ale cerințelor propuse, nicidecum singurele metode de rezolvări posibile.

SUBIECTUL I (20 de puncte)

1c 2a 3d 4b 5c	5x4p.
----------------	-------

SUBIECTUL al II-lea (40 de puncte)

1	Algoritmul verifică dacă ultimele cifre din numărul x coincid cu ultimele cifre din numărul nr, unde se verifică inițial ultima cifră, apoi ultimele două și așa mai departe în funcție de valoarea lui p.		
	a) Răspuns corect: 3 (6p.)	b) Se acordă câte 3p. pentru fiecare set. Exemplu: 2 2 32 2 (6p.)	
	<pre>#include <iostream> using namespace std; int main() { int x, n, nr, p, i, k; p = 1; k = 0; cin >> x >> n; for(i = 1; i <= n; i++){ cin >> nr; if(nr % p == x % p) k++; p = p * 10; } cout << k; return 0; }</pre>	<pre>p←1; k←0 citește x,n (numere naturale nenule) i←1 execută citește nr (număr natural) dacă nr%p=x%p atunci k←k+1 p←p*10 i←i+1 cât timp i≤n scrie k</pre>	
2		3	
Numărul minim de arce: 3 Arcele: (E,F), (F,D), (D,A)		<pre>k=0; for(int i=1;i<strlen(s)-1;i++) if(strchr("!@#\$",s[i])!=NULL) k++; cout<<k;</pre>	
-afișarea numărului minim de arce		3p.	-inițializarea variabilei k și declararea altor variabile 1p.
-afișarea oricăror arce corespunzătoare		1x3p.	-determinarea valorii cerute 4p. -corectitudinea globală a secvenței ¹⁾ 1p.

SUBIECTUL al III-lea

(30 de puncte)

1		
<pre> void prime(int x, int y){ int d, k, OK, vecinStanga, vecinDreapta; for(int i = x; i <= y; i++){ OK = 1; for(d = 2; d <= i / 2; d++){ if(i % d == 0) OK = 0; } if(OK == 1){ for(k = i - 1; k >= 1; k--){ OK = 1; for(d = 2; d <= k / 2; d++){ if(k % d == 0) OK = 0; } if(OK == 1){ vecinStanga = k; k = 1; } } for(k = i + 1; k <= 2 * i; k++){ OK = 1; for(d = 2; d <= k / 2; d++){ if(k % d == 0) OK = 0; } if(OK == 1){ vecinDreapta = k; k = 2 * i; } } cout << "(" << vecinStanga << " " << i << " " << vecinDreapta << ") "; } } } </pre>	-antet al subprogramului (*) -determinare a valorii cerute (**) -afișare a datelor -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului¹⁾ (*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect specific (algoritm principal corect de determinare al unui număr prim, algoritm principal corect de determinare al unui vecin din stânga, algoritm principal corect de determinare al unui vecin din dreapta) conform cerinței.	2p. 6p. 1p. 1p.
2		
<pre> #include <iostream> using namespace std; int main() { int M[20][20], n, k, i, j, aux; cin >> n; for(i = 0; i < n; i++) for(j = 0; j < n; j++) cin >> M[i][j]; cin >> k; while(k != 0){ for(i = 0; i < n; i++){ if(M[i][i] % 2 == 0){ aux = M[i][i]; for(int j = i + 1; j < n; j++){ if(M[j][j] % 2 == 0){ M[i][i] = M[j][j]; i = j; } } M[i][i] = aux; } } k--; } for(i = 0; i < n; i++){ for(j = 0; j < n; j++){ cout << M[i][j] << " "; } cout << endl; } return 0; } </pre>	-declarare a unei variabile care să memoreze un tablou bidimensional -citire a datelor -determinare a datelor cerute (*) -afișare a datelor -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului¹⁾ (*) Se acordă câte 1p. pentru fiecare aspect specific (identificare a unui element de pe diagonala principală, identificare a unui element par, permutare a unui element par, permutare de k ori, mutare element dintr-un capăt în celălalt, valori suport) conform cerinței.	1p. 1p. 6p. 1p. 1p.

3		
a	b	
Algoritmul proiectat afișeze pe ecran de câte ori apare primul șir în cel de-al 2-lea sau mesajul nu exista în caz contrar cu ajutorul unui vector care memorează cele maxim 10 elemente din fișier. Ulterior se compară fiecare număr care apare în fișier cu elementele vectorului. Dacă găsim un element comun, se trece la următorul, dacă nu, resetăm totul. Repetăm acești pași până găsim toate elementele din vector în fișier sau nu găsim nimic și afișăm mesajul conform. Algoritmul proiectat este eficient din punct de vedere al timpului de executare, deoarece este unul liniar, având complexitatea $O("n")$, unde 'n' reprezintă numărul de elemente din fișier.	<pre>#include <iostream> #include <fstream> using namespace std; int main() { ifstream f("bac.txt"); int V[11], n, m, i, nr, k = 0, j = 0, cnt = 0, nrFinal = 0; f >> n >> m; for(i = 0; i < n; i++) f >> V[k++]; for(i = 0; i < m; i++){ f >> nr; if(nr == V[j]){ cnt++; j++; if(cnt == n){ nrFinal++; j = cnt = 0; } } else{ j = 0; cnt = 0; if(nr == V[j]){ cnt++; j++; } } } if(nrFinal != 0) cout << nrFinal; else cout << "nu exista"; f.close(); return 0; }</pre>	
a) Pentru răspuns corect: -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principal corect dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă rezolvării corecte a acestei cerințe o reprezintă problema de mai sus.
b) Pentru răspuns corect: -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorilor cerute (**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului¹⁾	8p. 1p. 5p. 1p. 1p.	

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

Rezolvările prezente în barem nu reprezintă singurele metode de rezolvare a cerințelor propuse