

CUPRINS

<i>Calendarul examenului de Bacalaureat 2009</i>	3
<i>Programa pentru examenul de Bacalaureat – Informatică, intensiv</i>	4

	SUBIECTUL I	SUBIECTUL II	SUBIECTUL III
Varianta 1	7	111	313
Varianta 2	8	113	316
Varianta 3	9	115	319
Varianta 4	10	117	322
Varianta 5	11	119	325
Varianta 6	12	121	327
Varianta 7	13	123	329
Varianta 8	14	125	332
Varianta 9	15	127	335
Varianta 10	16	129	337
Varianta 11	17	131	339
Varianta 12	18	133	341
Varianta 13	19	135	343
Varianta 14	20	137	346
Varianta 15	21	139	348
Varianta 16	22	141	350
Varianta 17	23	143	352
Varianta 18	24	145	354
Varianta 19	25	147	356
Varianta 20	26	149	359
Varianta 21	27	151	361
Varianta 22	28	153	363
Varianta 23	29	155	365
Varianta 24	30	157	367
Varianta 25	31	159	369
Varianta 26	32	161	371
Varianta 27	33	163	373
Varianta 28	34	165	375
Varianta 29	35	167	377
Varianta 30	36	169	380
Varianta 31	37	171	382
Varianta 32	38	173	384
Varianta 33	39	175	386
Varianta 34	40	177	388

Varianta 35	41	179	390
Varianta 36	42	181	392
Varianta 37	43	183	394
Varianta 38	44	185	396
Varianta 39	45	187	398
Varianta 40	46	189	400
Varianta 41	47	191	402
Varianta 42	48	193	404
Varianta 43	49	195	406
Varianta 44	50	197	408
Varianta 45	51	199	411
Varianta 46	52	201	413
Varianta 47	53	203	415
Varianta 48	54	205	417
Varianta 49	56	207	419
Varianta 50	57	209	421
Varianta 51	58	211	424
Varianta 52	59	213	427
Varianta 53	60	215	430
Varianta 54	61	217	432
Varianta 55	62	219	435
Varianta 56	63	221	438
Varianta 57	64	223	440
Varianta 58	65	225	442
Varianta 59	66	227	444
Varianta 60	67	229	446
Varianta 61	68	231	448
Varianta 62	69	233	451
Varianta 63	71	235	453
Varianta 64	73	237	455
Varianta 65	74	239	457
Varianta 66	75	241	460
Varianta 67	76	243	462
Varianta 68	77	245	464
Varianta 69	78	247	466
Varianta 70	79	249	469
Varianta 71	80	251	471
Varianta 72	81	253	473
Varianta 73	82	255	475
Varianta 74	83	258	477

Varianta 75	84	260	480
Varianta 76	86	262	483
Varianta 77	87	264	485
Varianta 78	88	266	487
Varianta 79	89	268	489
Varianta 80	90	270	492
Varianta 81	91	272	495
Varianta 82	92	274	497
Varianta 83	93	276	500
Varianta 84	94	278	502
Varianta 85	95	280	504
Varianta 86	96	282	506
Varianta 87	97	284	508
Varianta 88	98	286	510
Varianta 89	99	288	512
Varianta 90	100	290	514
Varianta 91	101	292	517
Varianta 92	102	294	519
Varianta 93	103	296	521
Varianta 94	104	298	524
Varianta 95	105	300	526
Varianta 96	106	302	528
Varianta 97	107	304	531
Varianta 98	108	306	533
Varianta 99	109	308	535
Varianta 100	110	310	537

SUBIECTUL I

VARIANTA 003

1. Care dintre următoarele expresii Pascal|C/C++ are valoarea `true` | 1 dacă și numai dacă numărul real memorat în variabila `x` se află în intervalul $(-2, 2)$? (4p.)

(Pascal)

- a. `x*x-4<=0`
- b. `4-x*x>0`
- c. `(2<x) and (x<-2)`
- d. `(x-2) * (x+2) >0`

(C/C++)

- a. `x*x-4<=0`
- b. `4-x*x>0`
- c. `(2<x) &&(x<-2)`
- d. `(x-2) * (x+2) >0`

Răspuns corect b.

2. Se consideră algoritmul alăturat, descris în pseudocod:

S-a notat cu `x%y` restul împărțirii numărului întreg `x` la numărul întreg nenul `y`.

- a) Scrieți ce se va afișa dacă se citeșc, în această ordine, numerele: 2 5 16 9 12 13 5 0. (6p.)

Răspuns corect: 599350

```

citește z, x
  (numere naturale nenule)
cât timp x>0 execută
  citește y (număr natural)
  dacă z<y-x atunci
    scrie x%10
  altfel
    scrie y%10
  x←y

```

- b) Scrieți un șir de date de intrare, format doar din numere naturale cu o singură cifră fiecare, care să determine afișarea valorii 7310. (4p.)

Răspuns corect: de exemplu, 9 1 7 3 1 0. Șirul de date de intrare poate fi de forma: `z, x, 7, 3, 1, 0`, unde $0 < z \leq 9$ și $\text{maximum}\{7-z, 1\} \leq x \leq 9$

- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura `cât timp ... execută` cu o structură repetitivă de alt tip. (6p.)

Rezolvare. O soluție poate fi algoritmul alăturat:

```

citește z, x
  (numere naturale nenule)
repetă
  citește y (număr natural)
  dacă z<y-x
    atunci scrie x%10
  altfel scrie y%10
  x←y
cât timp x>0

```

- d) Scrieți programul Pascal | C/C++ corespunzător algoritmului dat. (10p.)

SUBIECTUL II

VARIANTA 010

1. Considerând declarația alăturată, care dintre următoarele secvențe de instrucțiuni afișează valorile memorate în cele două câmpuri ale variabilei **x**, separate printr-un spațiu? (4p.)

(Pascal)	(C/C++)
<pre>type p=record a,b:integer end; var x:p;</pre>	<pre>struct { int a; int b; } x;</pre>
<p>a. write(x.a, ' ', x.b);</p> <p>b. write(a.x, ' ', b.x);</p> <p>c. write(x);</p> <p>d. write(x->a, ' ', x->b);</p>	<p>a. cout<<x.a<<" "<<x.b; printf("%d %d", x.a, x.b);</p> <p>b. cout<<a.x<<" "<<b.x; printf("%d %d", a.x, b.x);</p> <p>c. cout<<x; printf("%d", x);</p> <p>d. cout<<a->x<<" "<<b->x; printf("%d %d", a->x, b->x);</p>

Răspuns corect a

2. Într-o listă liniară simplu înlănțuită, alocată dinamic, fiecare element reține în câmpul **info** o valoare întreagă, iar în câmpul **urm** adresa elementului următor din listă sau **nil/NULL** dacă nu există un element următor. Variabila **p** reține adresa primului element din listă. Lista conține, începând de la primul element, în această ordine, valorile: 2, 3, 4, 5, 6, 7, 8. Ce se va afișa în urma executării secvenței de instrucțiuni alăturată? (4p.)

(Pascal)	(C/C++)
<pre>while (p^.urm<>nil) and (p<>nil) do begin p^.urm:=p^.urm^.urm; p:=p^.urm; write(p^.info, ' '); end;</pre>	<pre>while (p->urm!=NULL&&p!=NULL) { p->urm=p->urm->urm; p=p->urm; cout<<p->info<<' '; }</pre>
<p>a. 2 5 8 b. 2 4 8</p>	<p>c. 2 4 6 8 d. 4 6 8</p>

Răspuns corect d

3. Se consideră un graf neorientat cu 80 de noduri și 3160 muchii. Care este numărul de muchii ce pot fi eliminate astfel încât graful parțial obținut să fie arbore? (6p.)

Răspuns corect: 3081

4. Ce se va afișa în urma executării secvenței de instrucțiuni alăturate dacă variabila **s** memorează șirul de caractere **abbacdde**, iar variabila **i** este de tip întreg? (6p.)

SUBIECTUL II

(Pascal)

```
i:=1;  
while i<length(s) do  
  if s[i]=s[i+1] then  
    delete(s,i,1)  
  else  
    i:=i+1;  
write(s);
```

Răspuns corect: abacde

(C/C++)

```
i=0;  
while (i<strlen(s)-1)  
  if (s[i]==s[i+1])  
    strcpy(s+i,s+i+1);  
  else i=i+1;  
cout<<s; | printf("%s",s);
```

5. Scrieți un program Pascal|C/C++ care citește de la tastatură două numere naturale n și p ($2 \leq n \leq 20$, $1 \leq p \leq 20$) și construiește în memorie un tablou bidimensional cu n linii și p coloane. Tabloul va fi construit astfel încât, parcurgând matricea linie cu linie de sus în jos și fiecare linie de la stânga la dreapta, să se obțină șirul primelor $n \cdot p$ pătrate perfecte pare, ordonat strict crescător, ca în exemplu. Tabloul astfel construit va fi afișat pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, cu câte un spațiu între elementele fiecărei linii.

Exemplu: pentru $n=2$, $p=3$ programul va afișa

0 4 16	
36 64 100	(10p.)

Rezolvare: O posibilitate de construire a matricei este prezentată de secvența pseudocod:

```
k ← 0  
┌ pentru i ← 1, n execută  
│ ┌ pentru j ← 1, p execută  
│ │ A[i, j] ← k*k  
│ │ k ← k+2  
│ └─┘  
└─┘
```