

1. (0.9p) Variabilele x și y pot memora valori întregi semipozitive. $x/2+x\%y-x/y==0$

Indicați numerele cu care x și y pot fi inițializate astfel încât expresia

C/C++ alăturată să aibă valoarea 1.

a. $x=9$ și $y=3$ b. $x=8$ și $y=4$ c. $x=4$ și $y=1$ d. $x=4$ și $y=2$

2. (0.9p) Utilizând metoda backtracking se generează, în ordine crescătoare, numere naturale de patru cifre din mulțimea $A = \{1,2,3,4,5\}$, numere care nu conțin două cifre impare alăturate Primele 8 numere generate sunt, în această ordine: 1212, 1214, 1221, 1222, 1223, 1224, 1225, 1232. Numărul de valori generate care au cifra miilor egală cu 3 și cifra unităților egală cu 2 este:

a. 20 b. 16 c. 10 d. 7

3. (0.9p) Se consideră fragmentul de cod alăturat, în care a este o matrice 4×4 având ca elemente numere întregi, iar $x\%y$ este restul împărțirii numărului natural x la numărul natural nenul y .

Valoarea afișată este:

a. 45 b. 52 c. 17 d. 10

```
int a[5][5], tr=0;
for(i=0;i<5;i++)
{
    for(j=0;j<5;j++)
        a[i][j]=(i*i+2*j+1)%5;
    tr += a[i][i];
}
printf("tr = %d\n\n",tr);
```

4. (0.9p) Variabila fig declarată alăturat reține lungimea laturii unui pătrat și coordonatele vârfului din stânga jos al pătratului. Secvența de instrucțiuni care definește un pătrat cu vârful în punctul (1,1) și latura 5.5 este:

a. $fig.vf.x = 1.;$
 $fig.vf.y = 1.;$
 $fig.latura = 5.5;$

b. $fig.punct.x = 1.;$
 $fig.punct.y = 1.;$
 $fig.latura = 5.5;$

c. $fig.punct->x = 1.;$
 $fig.punct->y = 1.;$
 $fig->latura = 5.5;$

d. $fig->vf->x = 1.;$
 $fig->vf->y = 1.;$
 $fig->latura = 5.5;$

```
struct punct {
    float x, y;
};
struct patrat {
    struct punct vf;
    float latura;
} fig;
```

5. (0.9p) Expresia C/C++

$!(-20 >= x) \ \&\& \ (x <= -5) \ || \ (x >= 1) \ \&\& \ !(x > 20)$

are valoarea 1 dacă și numai dacă valoarea memorată de variabila reală x aparține mulțimii:

a. $(-\infty, -20] \cup [20, \infty)$ b. $[-20, -5] \cup (-5, 20)$ c. $(-20, -5) \cup (1, 20)$ d. $(-20, -5] \cup [1, 20]$

6. (0.9p) Se consideră subprogramul f alăturat ($x\%y$ este restul împărțirii numărului natural x la numărul natural nenul y).

Rezultatul executării instrucțiunii:

`cout << f(0) << ' ' << f(20250708) << "\n\n";`

este:

a. 0 9 b. 0 2 c. 5 0 d. 25 450

```
int f (int n)
{
    if (n!=0)
        return n%5 + f(n/10);
    return 0;
}
```

7. (0.9p) Se consideră graful orientat cu 6 vârfuri, numerotate de la 1 la 6, reprezentat prin listele de adiacență indicate alăturat.

1: 4, 5
 2: 3, 6

Două drumuri sunt distincte dacă diferă prin cel puțin un arc. Numărul de drumuri distincte de la vârful 2 la vârful 4 este:

a. 0 b. 2 c. 1 **d. 3**

3: 1, 6
4: 5
5: listă vidă
6: 4, 5

8. (0.9p) Se consideră funcția `str_mod()` definită alăturat. Secvența de cod:

```
char txt[32];  
strcpy(txt, "polariton");  
str_mod(txt);
```

are ca efect afișarea textului:

a. plrtn **b. notiralop**
c. POLARITON d. nu este nimic afișat

```
void str_mod(char *s)  
{  
    int c, i, j;  
    for (i = 0, j = strlen(s)-1; i < j; i++, j--)  
    {  
        c = s[i]; s[i] = s[j]; s[j] = c;  
    }  
    cout << s << endl;  
}
```

9. (0.9p) Frunzele arborelui cu rădăcină, având 8 noduri, numerotate de la 1 la 8, reprezentat prin vectorul "de tați" (3,5,5,2,0,3,2,4) sunt:

a. 2,5,7,6 b. 1,8 c. 2,5,7 **d. 1,6,7,8**

10. (0.9p) Se consideră funcția `subst()` definită alăturat. Secvența de cod:

```
long nr = 235690;  
printf("subst(%ld) = %ld\n",nr,subst(nr));
```

are ca efect afișarea textului:

a. `subst(nr) = -1L`
b. `subst(235690) = 224680`
c. `subst(235690) = 124589`
d. `subst(235690) = 200600`

```
long subst(long n)  
{  
    if(n<0 || n>10000000)  
        return -1L;  
    unsigned int c[6],i;  
    long val = 0L;  
    for(i=0;i<6;i++)  
    {  
        c[5-i] = n%10;  
        if(c[5-i]%2)  
            c[5-i] = c[5-i]-1;  
        val += c[5-i]* (long)pow(10,i);  
        n = n/10;  
    }  
    return val;  
}
```

11. (4p) Fișierul text *input.txt* conține pe fiecare linie un singur număr întreg. Scrieți un program C/C++ care să afișeze numărul valorilor multipli de 4 din fișier și suma acestor valori.

12. (5p) Fișierul text *input.txt* conține pe fiecare linie trei stringuri, cu structura descrisă mai jos. Scrieți un program C/C++ care să identifice compușii supraconductori incluși în fișier, să afișeze pe linii distincte formula lor chimică și, la final, numărul lor total.

Exemplu de structură a fișierului de date de intrare:

Cu metal -

MgB2 supraconductor 39

Au metal -

...

```

printf("***** Subiectul 11 *****\n");
FILE * fin = fopen("input1.txt","r");
int num_values = 0, val, sum = 0;
while (fscanf(fin, "%d", &val) == 1)
{
    if(val % 4 == 0)
    {
        num_values++;
        sum += val;
    }
}
fclose(fin); fin = NULL;
printf("%d multipli de 4, suma lor = %d\n",num_values,sum);

printf("***** Subiectul 12 *****\n");
FILE *fp = fopen("input.txt","r");
char line[256];
int nr_occ = 0;

if (fp != NULL)
{
    while (fgets(line, sizeof(line), fp))
    {
        char *res;

        res = strtok(line, " ");
        if((strcmp(res,"supraconductor")==0))
        {
            nr_occ += 1;
            printf("%s\n",line);
        }

        while(res != NULL)
        {
            if((strcmp(res,"supraconductor")==0))
            {
                nr_occ += 1;
                printf("%s\n",line);
            }
            res = strtok(NULL, " ");
        }
    }

    fclose(fp);
    printf("Cuvântul supraconductor apare de %d ori.\n",nr_occ);
}

```