①

**ENUNȚ**

Să se determine valoarea minimă care are exact 2 cifre impare dintr-un șir a cu n elemente.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5; | * 11
 |
| a=[11, 13, 131, 19, 9] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>#include <algorithm>#include <climits>using namespace std;int nrcfimp(int n){ if(n<10) return (n%2); else { return (n%2) + nrcfimp(n/10); }}void minim(int a[], int n){ int mini=INT\_MAX; for(int i=0; i<n; i++) { if(nrcfimp(a[i])==2) { if(a[i]<mini) mini=a[i]; } } cout<<mini;} | int main(){ int n, i; cin>>n; int a[n+1]; for(i=0; i<n; i++) cin>>a[i]; minim(a, n); return 0;} |

②

**ENUNȚ**

Se citește un șir a cu n elemente. Să se determine cmmdc-ul celor n elemente.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5; | * 5
 |
| a=[15, 25, 35, 45, 55] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>using namespace std;int cmmdc(int a, int b){ if(b==0) return a; else { return cmmdc(b, a%b); }}void f (int a[], int n){ int cm=a[0], i; for(i=1; i<n; i++) { cm=cmmdc(cm, a[i]); } cout<<cm;} | int main(){ int n, i; cin>>n; int a[n+1]; for(i=0; i<n; i++) cin>>a[i]; f(a, n); return 0;} |

③

**ENUNȚ**

Se citește un șir a cu n elemente. Să se determine cmmmc-ul celor n elemente.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5; | * 17325
 |
| a=[15, 25, 35, 45, 55] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>using namespace std;int cmmdc(int a, int b){ if(b==0) return a; else { return cmmdc(b, a%b); }}void f (int a[], int n){ int cm=a[0], i; for(i=1; i<n; i++) { cm=cm\*a[i]/cmmdc(cm, a[i]); } cout<<cm;} | int main(){ int n, i; cin>>n; int a[n+1]; for(i=0; i<n; i++) cin>>a[i]; f(a, n); return 0;} |

④

**ENUNȚ**

Se citește un șir a cu n elemente. Să se calculeze suma elementelor prime.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5; | * 15
 |
| a=[1, 3, 5, 7, 9] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>using namespace std;int prim(int n, int i=2){ if(n<2) return 0; if(n==2) return 1; if(i\*i>n) return 1; if(n%i==0) return 0; return(prim(n, i+1));}void f (int a[], int n){ int s=0, i; for(i=0; i<n; i++) { if(prim(a[i])) s+=a[i]; } cout<<s;} | int main(){ int n, i, a[100]; cin>>n; for(i=0; i<n; i++) cin>>a[i]; f(a, n); return 0;} |

⑤

**ENUNȚ**

Se citește un șir a cu n elemente. Să se determine numărul elementelor egale cu o valoare dată V.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5;V=3; | * 3
 |
| a=[1, 3, 3, 2, 3] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>using namespace std;int f (int a[], int n, int V){ if(n==1) { if(a[n-1]==V) return 1; else return 0; } else { return (a[n-1]==V)+f(a, n-1, V); }}int main(){ int n, i, a[100], V; cin>>n>>V; for(i=0; i<n; i++) cin>>a[i]; cout<<f(a, n, V); return 0;} |  |

⑥

**ENUNȚ**

Se citește un șir a cu n elemente. Să se determine suma elementelor cu proprietatea că cifrele sunt egale.

**EXEMPLU**

|  |  |
| --- | --- |
| n=5; | * 61
 |
| a=[1, 2, 3, 12, 55] |

**REZOLVARE**

|  |  |
| --- | --- |
| #include <iostream>using namespace std;int ceg(int n, int nn){ if(n<=9) { if(n==nn) return 1; else return 0; } else { return (n%10==nn)\*ceg(n/10, nn); }} | int f (int a[], int n){ if(n==1) { if(ceg(a[n-1], a[n-1]%10)) return a[n-1]; else return 0; } else { if(ceg(a[n-1], a[n-1]%10)) return a[n-1]+f(a, n-1); else return f(a, n-1); }}int main(){ int n, i, a[100], V; cin>>n; for(i=0; i<n; i++) cin>>a[i]; cout<<f(a, n); return 0;} |

⑦

**ENUNȚ**

Pentru definiţia de mai jos a subprogramului f, ce se afişează ca urmare a apelului f(121,1);?

void f(long n, int i)

{

 if(n!=0)

 if(n%3>0)

 {

 cout<<i;

 f(n/3,i+1);

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| n= 121 --> n%3= 1 >0 --> i= 1 --> n/3= 40 --> i+1 =2n= 40 --> n%3= 1 >0 --> i= 2 --> n/3= 13 --> i+1 =3n= 13 --> n%3= 1 >0 --> i= 3 --> n/3= 4 --> i+1 =4n= 4 --> n%3= 1 >0 --> i= 4 --> n/3= 1 --> i+1 =5n= 1 --> n%3= 1 >0 --> i= 5 --> n/3= 0 --> STOP | 12345 |

**RĂSPUNS**

12345

⑧

**ENUNȚ**

Pentru definiţia de mai jos a subprogramului f, ce se afişează ca urmare a apelului f(12345);?

void f(long n)

{

 cout<<n%10;

 if(n!=0)

 {

 f(n/100);

 cout<<n%10;

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| f(12345) --> 5f(123) --> 3f(1) --> 1f(0) --> 0STOPrest(1) --> 1rest(123) --> 3rest(12345) --> 5 | 5310135 |

**RĂSPUNS**

5310135

⑨

**ENUNȚ**

Pentru definiţia alăturată a subprogramului f, ce se afişează ca urmare a apelului f(26);?

void f (int x)

{

 if(x>0)

 if(x%4==0)

 {

 cout<<’x’;

 f(x-1);

 }

 else

 {

 f(x/3);

 cout<<’y’;

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
|  x= 26 >0 --> x%4= 2 --> ramura 2--> x/3= 8 --> x= 8 >0 --> x%4= 0 --> ramura 1 --> 'x'--> f(7) --> x= 7 >0 --> x%4= 3 --> ramura 2--> x/3= 2 --> x= 2 >0 --> x%4= 2 --> ramura 2--> x/3= 0 --> x= 0 >0 STOP--> rest(8) --> 'y'--> rest(7) --> 'y'--> rest(2) --> 'y' | xyyy |

**RĂSPUNS**

xyyy

⑩

**ENUNȚ**

Pentru definiţia alăturată a subprogramului f, ce se afişează ca urmare a apelului f(15,2);?

void f (int n, int x)

{

 if(x>n)

 cout<<0;

 else if(x%4<=1)

 f(n,x+1);

 else

 {

 f(n,x+3);

 cout<<1;

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| ( 15 , 2 ) x%4= 2 >1 --> f(15, 5) --> 1 rest( 15 , 5 ) x%4= 1 <=1 --> f(15, 6)( 15 , 6 ) x%4= 2 >1 --> f(15, 9) --> 1 rest( 15 , 9 ) x%4= 1 <=1 --> f(15, 10)( 15 , 10 ) x%4= 2 >1 --> f(15, 13) --> 1 rest( 15 , 13 ) x%4= 1 <=1 --> f(15, 14)( 15 , 14 ) x%4= 2 >1 --> f(15, 17) --> 1 rest( 15 , 17 ) --> x>n --> 0 STOP | 01111 |

**RĂSPUNS**

01111

⑪

**ENUNȚ**

Pentru subprogramul f definit mai jos, ce se afişează ca urmare a apelului f(3,17)?

void f ( int a, int b)

{

 if(a<=b)

 {

 f(a+1,b-2);

 cout<<’\*’;

 }

 else

 cout<<b;

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| ( 3 , 17 ) --> f ( 4 , 15 ) --> \* rest( 4 , 15 ) --> f ( 5 , 13 ) --> \* rest( 5 , 13 ) --> f ( 6 , 11 ) --> \* rest( 6 , 11 ) --> f ( 7 , 9 ) --> \* rest( 7 , 9 ) --> f ( 8 , 7 ) --> \* rest( 8 , 7 ) --> 7 STOP | 7\*\*\*\*\* |

**RĂSPUNS**

7\*\*\*\*\*

⑫

**ENUNȚ**

Se consideră subprogramul alăturat:

Ce valoare are f(128,2)?

int f(int a, int b)

{

 if (b<1)

 return -1;

 else if (a%b==0)

 return 1+f(a/b,b);

 else

 return 0;

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| ( 128 , 2 ) --> 128%2=0 --> 1 + f ( 64 , 2)( 64 , 2 ) --> 64%2=0 --> 1 + f ( 32 , 2)( 32 , 2 ) --> 32%2=0 --> 1 + f ( 16 , 2)( 16 , 2 ) --> 16%2=0 --> 1 + f ( 8 , 2)( 8 , 2 ) --> 32%2=0 --> 1 + f ( 16 , 2)( 4 , 2 ) --> 4%2=0 --> 1 + f ( 2 , 2)( 2 , 2 ) --> 2%2=0 --> 1 + f ( 1 , 2)( 1 , 2 ) --> 1%2=1 --> 0 STOP--> 1 + ( 1 + ( 1 + ( 1 + ( 1 + ( 1 + 1 ) ) ) ) ) = 7 | /\* cout<<f(128, 2) \*/7 |

**RĂSPUNS**

7

⑬

**ENUNȚ**

Se consideră subprogramul f definit alăturat. Ce se va afişa în urma apelului f(12345);?

void f(long int n)

{

 if (n!=0)

 {

 if (n%2 == 0)

 cout<<n%10;

 f(n/10);

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| n= 12345 --> n%2= 1 --> f(1234)n= 1234 --> n%2= 0 --> n%10 = 4 --> **4** --> f(123)n= 123 --> n%2= 1 --> f(12)n= 12 --> n%2= 0 --> n%10 = 2 --> **2** --> f(1)n= 1 --> n%2= 1 --> n%10 = 1 --> f(0)n= 0 STOP | 42 |

**RĂSPUNS**

42

⑭

**ENUNȚ**

Pentru definiţia de mai jos a subprogramului f, ce se afişează ca urmare a apelului f(121,1);?

void f(long n, int i)

{

 if(n!=0)

 if(n%3>0)

 {

 cout<<i;

 f(n/3,i+1);

 }

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
|  n= 3 --> f(2)--> n= 2 --> n%2=0 --> 2 space --> f(1)--> n= 1 --> f(0)--> endl --> STOPrest --> 1 spacerest --> 2 spacerest --> 3 space | 2 1 2 3 |

**RĂSPUNS**

2\_

1\_2\_3\_ (\*

|  |  |
| --- | --- |
| (\* \_ = spațiu |  |

⑮

**ENUNȚ**

Se consideră subprogramul f cu definiţia alăturată. Ce valoare are f(1213111,1)?

int f (long int n, int k)

{

 if (n!=0)

 if(n%10==k)

 return 1+f(n/10,k);

 else

 return 0;

 else

 return 0;

}

**EXEPLICAȚIE**

|  |  |
| --- | --- |
| **RULEAZĂ** | **COUT** |
| n=1213111 k=1 --> 1213111%10 = 1 = k --> 1 + f(121311, 1)n= 121311 k=1 --> 121311%10 = 1 = k --> 1 + f(12131, 1)n= 12131 k=1 --> 12131%10 = 1 = k --> 1 + f(1213, 1)n= 1213 k=1 --> 1213%10 = 3 != k --> 0 STOP--> 1 + ( 1 + 1 ) = 3 | 3 |

**RĂSPUNS**

3