1. $\boxed{b}$ 2. $\boxed{c}$. 3. $\boxed{a}$. 4. $\boxed{c.6}$ 5. $\boxed{b.2}$

II 1. a. 2.

b). 28, 29.

c)

```
#include <iostream>.
int main()
{ int n, nr=0; std::cin>>n;
  for(int i=n; i<=1; i--)
  { int x=0, y=1; int r;
    for(int j=1; j<=i; j++)
    { r=2*x-y; x=y; y=r; }.
    If(y>0)
      nr++;
  }
  std::cout<<nr;
  return 0
}.
```

d)

```
citeste n.
nr ← 0
i ← n;
cât timp (i ≥ 1) executa
    x ← 0 y ← 1
    pentru j ← 1, i executa
        r ← 2*x-y; x ← y; y ← R;
    daca y>0 atunci
        nr ← nr+1.
    i ← i-1
scrie nr
```

2. $x_1 = 4$, $x_2 = 5$.

3.

```
for(int i=0; i<4; i++)
{ for(int j=0; j<5; j++)
  { a[i][j]=(j·4)+(i+1); }.
}.
```

S III. 1.

```
int divPrimMax(int n, int& p)
{ ~~p=2~~; p=0; int d=2;
  while(n) (n>1) //altfel inf loop.
  ~~{ If(n%d==0)~~         If(n%d==0)
  ~~  { p=d; n/=d }.~~      { p=d; n/=d; }.
  ~~else.~~                else
  ~~  d++;~~               d++.
}.
```

```
2.  # include <iostream>
    # include <cstring>.
    int main()  char p[105];                  p
    { char ~~str~~ [105]; std::cin.getline(str, 100); int cons=0, vow=0;
           ~~string~~                                       int nr=0; char *str=p;
      while(*str)
      { ~~If(strchr(*str, "aeiou"))~~
        {
        If(*str=='\0')
        { ~~vow~~ If(vow==cons)
                      nr++;
           vow=0, cons=0;
        }
        else If(strchr(*str, "aeiou"))
                 vow++;
             else
                 cons++;
        }                  && vows!=0)
        If( vows == cons )     // Nu mi-am dat seama până nu l-am
        {    nr++; }           // rulat dar dacă nu am pus-o
        std::cout << nr;       // la final nu numără ultima val.
        return 0;
    }.
```

```
# include < iostream >
# include < fstream >.
 int main()
{ std :: ofstream fout ("leac.out");
  int p1, p2; std::cin>>p1 >>p2;
   for(int i = 1; i <= 9; i++)
    {    if((p1%i)&& p1/i <= 9)
         { int i2 = p1/i;
            { for(int j = 1; j <= 9 && p2/j != 0 ; j++)
               {.    if( p2%j == 0 && p2/j <= 9)    j2 = p2/j;
                   {    for( e = 0; e <= 9; e++)
                         { ~~std :: cout<<i<<i2<<e<<e<<e<<j<<j2~~
                        std:: cout<<i<<i2<<e<<e<<e<<j<<j2<<'\n';
                   }.                                          <<'\n';
               }.
            }.
         }.
    }
    return 0;
}.
```

// Este eficient dpdvd. al timpului deoarece nu generează de mai multe ori cifrele, oprindu-se atunci când p1/i sau p2/j sunt < 1.

// Dpdvd al memoriei nu utilizează structuri unidimensionale.

// Alg iterează prin valorile posibile ale lui i (prima cifră) și continuă doar dacă verifică condiția, apoi generează ultimele 2 cif prin aceeași metodă.

// Enunțul cere pregenerarea prime + ultime cifre și num lor.

//