

Subiectul I	
1.a	d)
2. a) -3 b) a=5, b=10, k=3 sau a=10, b=5, k=3 c) citeste a,b,k (numere naturale) nr←1 s←0 ┌daca a>b atunci │ i←a │ a←b │ b←i │ ── i←a ┌cattimp i<=b executa │ s←s+nr*i │ nr←-1*nr │ i←i+1 │ ── scriek+s	<pre>#include <iostream> using namespace std; int main() { int i,k,a,b,nr,s; cin>>a>>b>>k; nr=1; s=0; if(a>b) {i=a; a=b; b=i;} for(i=a;i<=b;i++) { s=s+nr*i; nr=-1*nr; } cout<<k+s; return 0; }</pre>
Subiectul II	
1.c	5.
2. d	#include <iostream>
3.if(a[k][k]<a[k][n+1-k]) cout<<a[k][k]; else cout<<a[k][n+1-k];	#include <cstring> using namespace std;
4. 2,3	int main() { char s[71]; int k,p,i; cin.getline(s,71); k=0;p=1; for(i=0;i<strlen(s);i++) if(s[i]=='a' && p==1) { k++; p=0; } else if(s[i]==' ') p=1; cout<<k; return 0; }
Subiectul III	
1.a	4.b)
2.1342, 136	#include <fstream> #include <iostream> using namespace std; ifstream fin("numere.in"); int main() { int a,k=0; while(fin>>a) if(a%3==1) { cout<<a<<" "; k++; } if(k==0) cout<<"Nu exista"; fin.close(); return 0; }
3. void num(int n,int v[50]) { for(int i=1;i<n;i++) if(v[i]<=v[0]) v[i]=0; v[0]=0; }	
4.a) Observam ca sirul descris este o progresie aritmetica de ratie 3 si termen initial 1, deci toate numerele din sir vor fi de forma 3k+1. In timpul citirii elementelor din fisier le afisam pe cele care au restul 1 la împărțirea la 3. Algoritmul este eficient ca spatiu de memorie deoarece se folosesc doar 2 variabile simple (nu se folosesc vectori pentru memorarea numerelor citite). Algoritmul este eficient ca timp de executie deoarece rezolva problema de la prima parcurgere a elementelor din fisier, adica la citirea acestora.	