1. Scrieţi  programul care citeste  un numar si afiseaza suma cifrelor de pe pozitiile

impare.  N=23456 , 6 pozitia 0, 5, pozitia 1, cifra 4 pozitia …

|  |  |
| --- | --- |
| s=0; i=0;/// i este pozitia numerotata de la dr la st(0 pozitia unitatilor)  while (n!=0) ///  { c=n%10;  if(i%2==1) s=s+c;  i++;  n=n/10;  } | int v[10];  i=1;  while(n!=0)  { c= n%10;  v[i++]=c;  n=n/10;  }  nrc=i-1;  N=2345 … v=(5,4,3,2) |

Exemplu : - aflam cifra de pe pozitia k

N=2343 si k=5

|  |
| --- |
| cin>>n>>k;  s=0; i=0;/// I este pozitia numerotata de la dr la st(o pozitia unitatilor)  cif=-1;  while (n!=0) ///  { c=n%10;  if(i==k)  cif =c;  i++;  n=n/10;  }  If(cif>=0)  cout <<cif;  else  cout <<”Nu exista pozitia ”; |

2. Scrieţi programul care citeste un   numar si afiseaza  numarul fara prima si ultima cifra.

 N=23467 nc= 346

|  |
| --- |
| n=n/10; /// elimina ultima cifra Ex =34507  nc=0; p=1;  while (n>9) /// nu va lua prima cifra /// while (n>99)  { c=n%10;  nc= nc + c\*p;  p=p\*10;  n=n/10;  }  Cout <<nc; |

X= ab, a= x/pow(10,k) si b = x%pow(10,k)

Ex : x= 234567 k=3 …. X =ab unde a=234 iar b=567

P=1; while(n>9) { p=p\*10; n=n/10;} …. n=n%p;

3.  Scrieţi programul care citeste  un numar si verifica daca cifrele sunt in ordine crescatoare. n= 12378 ….

 n%10 (n/10)%10

|  |
| --- |
| ok=1;  while(n!=0)  { c=n%10; c1=(n/10)%10; /// cifre vecine  If ( c <c1)  Ok=0;  N=n/10;  }  If(ok==1)  cout <<”Cifre crescatoare”;  else  Cout <<<”Nu sunt ”; |

4. Scrieti programul care citeste  un numar si o cifra c. Programul afiseaza numarul obţinut din **n** prin eliminarea tuturor apariţiilor cifrei **c**.

Dacă, după eliminare, numărul nu mai conţine nicio cifră sau conţine doar cifre **0**, rezultatul returnat va fi **0**.

N=234256 se cere sa se stearga cifra c=2; rezultatul este n= 3456

 /// Stergerea cifrelor se face cu algoritmul de clonare

|  |
| --- |
| cin>>n>>c;  nc=0;p=1;  while(n!=0)  { cif=n%10;  If(c!=cif)  {nc=nc +cif\*p; p=p\*10; }  n=n/10;  } |

Exemplu : sa se stearga cifrele impare

|  |
| --- |
| cin>>n;  nc=0;p=1;  while(n!=0)  { c=n%10;  If(c %2==0 )  {nc=nc +c\*p; p=p\*10; }  n=n/10;  }  Cout <<nc; |

Exemplu : sa se dubleze cifrele pare : n=2340 rezulta nc=2234400

|  |
| --- |
| cin>>n;  nc=0;p=1;  while(n!=0)  { c=n%10;  If(c %2==0 )  {nc=nc +c\*p; p=p\*10;  nc=nc +c\*p; p=p\*10;  }  else  {nc=nc +c\*p; p=p\*10; }  n=n/10;  }  Cout <<nc; |

**5.** Scrieti programul care citeste  un numar **a** un număr natural afiseaza ultima cifră pară a sa. Dacă numărul nu conţine cifre pare, programul afiseaza valoarea **-1**.

De exemplu, dacă **a=8345**, subprogramul va returna **4**.

N=3175

|  |  |
| --- | --- |
| cin>>n;  cp=-1;  while(n!=0)  { c=n%10;  If(c%2==0)  {cp=c;break;}  n=n/10;  }  Cout <<cp; | cin>>n;  ok=0;  while(n!=0 && ok==0)  { c=n%10;  If(c%2==0)  {cp=c;ok=1; }  n=n/10;  }  If(ok==1)  Cout <<cp;  Else  Cout <<-1; |

6. Se citeşte de la tastatură un număr întreg x. Să se verifice dacă x este  
a) dublic - are doua cifre alăturate egale  
b) echilibrat - numărul de cifre pare este egal cu numărul de cifre impare  
c) ciuruit - are minim două zerouri  
d) corect - suma şi produsul cifrelor sunt pătrate perfecte  
e) legat - diferenţa dintre oricare două cifre este 1  
f) pitic - toate cifrele sunt mai mici decât 4

7. Se citeste un numar natural n. Sa se determine daca numarul este format doar din aceeasi cifra.  
Exemplu pentru numarul 1111 se afiseaza DA iar pentru 1211 se afiseaza NU.