**12)**

int main()

{

char s[256];

int i,n;

cin.getline(s,256);

n=strlen(s);

if (s[0]!=' ') s[0]-=32;

if (s[n-1]!=' ') s[n-1]-=32;

for(i=1;i<n-1;i++)

if(s[i]!=' ' && (s[i-1]==' ' || s[i+1]==' '))

s[i]-=32;

cout<<s;

return 0;

}

**1866)**

ifstream fin("prosir.in");

ofstream fout("prosir.out");

int main()

{

char c[201];

fin.getline(c, 201);

for(int i = 0; i < strlen(c)-1; i++)

if(c[i] != ' ' && (c[i + 1] == ' ' || c[i+1]=='.'))

c[i] = '5';

fout << c;

return 0;

}

**2828)**

char s[102], t[102];

int n,i,k;

int main()

{

cin.getline(s,102);

n=strlen(s);

if(s[0]>='A' && s[0]<='Z') t[**k++**]=s[i];

for(i=1;i<n;i++)

if(s[i]>='A' && s[i]<='Z' && s[i-1]==' ' )

t[**k++**]=s[i];

**t[k]=NULL;**

cout<<t;

return 0;

}

**13)**

**Sol 1 – character cu caracter**

char s[12];

int i,j,n;

int main()

{

cin.getline(s,12);

n=strlen(s);

for(i=n; i>=1; i--)

{

for(j=0; j<i; j++) cout<<s[j];

cout<<endl;

}

for(i=0; i<n; i++)

{

for(j=i; j<n; j++) cout<<s[j];

cout<<endl;

}

return 0;

}

**Sol 2 – cu functii sau pointeri**

**// prefixele**

for (i=strlen(s); i>=1; i--)

{ strncpy(t, s, i);

t[i]=0; // t[i]=NULL; terminator de sir

cout<<t<<endl;

}

**Sau**

for (i=strlen(s); i>=1; i--)

{ s[i]=0;

cout<<s<<endl;

}

**//sufixe**

**casa -> s+0**

**asa -> s+1**

**sa -> s+2**

**a -> s+3 // s+strlen(s)-1**

for(i=0; i<strlen(s); i++)

cout<<s+i<<endl; // strcpy(t, s+i); t[i]=0; cout<<t<<endl

**Obs. s+i =sufixul lui s care incepe cu s[i]**

**88) Palindrom**

**S=”capac”**

**Sol 1 – cu functia strrev(s)**

if(strcmp(s, strrev(s) )== 0) cout<<”DA”;

else cout<<”NU”;

**Sol 2 – cu verificarea perechilor de elemente simetrice fata de centru**

**n=strlen(s)**

**s[0], s[1], …. , s[n-2], s[n-1]**

**(s[0], s[n-1])**

**(s[1], s[n-2])**

**………………..**

**(s[i], s[n-1-i]), i=0,n/2-1**

ok=1;

for(i=0; i<n/2; i++)

if (s[i] != s[n-1-i])

{ ok=0;

break;

}

**sau**

ok=1;

i=0; j=n-1;

while (i<j && ok)

{ if (s[i] != s[j]) ok=0;

i++; j--;

}

**138)**

int mx=0, nr;

**char c, r;**

for(c=’a’; c<=’z’; c++)

{ // numaram in variabila nr frecventa literei c in sirul s

**nr=0;**

for(i=0; i<strlen(s); i++)

if (s[i]==c) nr++;

if(nr>mx)

{ mx=nr;

r=c;

}

}

cout<<r;

**155)**

int mx=0, nr;

**char c, r;**

for(c=’0’; c<=’9’; c++)

{ // numaram in variabila nr frecventa cifrei c in sirul s

**nr=0;**

for(i=0; i<strlen(s); i++)

if (s[i]==c) nr++;

if(nr>mx)

{ mx=nr;

r=c;

}

}

if (mx==0) cout<<”NU”;

else cout<<r;

**801)**

int main()

{

char s[256], char[256];

int i;

cin.getline(s,256);

while(s[0]==' ') // stergerea spatiilor de la inceputul sirului

**strcpy(s, s+1); // { strcpy(a,s+1); strcpy(s,a); } - PBINFO!**

while(s[strlen(s)-1]==' ') /// stergerea spatiilor de la sfarsitul sirului

s[strlen(s)-1]=0;

**i=0; // sterg spatiile din interior**

**while(i<strlen(s))**

**if(s[i]==' ' && s[i+1]==' ') strcpy(s+i,s+i+1); // { strcpy(a,s+i+1); strcpy(s+i,a);} - PBINFO!**

**else i++;**

cout<<s;

return 0;

}