1. **Program principal- Generarea permutarilor:**

#include <iostream>

using namespace std;

int n, s[100];

void init(int k)

{s[k]=0;}

int succesor(int k)

{if (s[k]<n)

{s[k]=s[k]+1;

return 1;}

else return 0; }

int valid(int k)

{for(int i=1; i<k; i++)

if(s[k]==s[i])

return 0;

return 1;}

int solutie(int k)

{ return k==n;}

void tipar()

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

cout<<s[i]<<" ";

cout<<endl; }

void bt(int k)

{

init(k);

while(succesor(k))

if(valid(k))

if(solutie(k))

tipar();

else bt(k+1);}

int main()

{

cout<<"n="; cin>>n;

bt(1);

}

**2. Sa se genereze toate permutarile numerelor de la 1 la n, in care 2 elememente aflate pe pozitiile consecutive sunt de paritati diferite:**

#include <iostream>

using namespace std;

int n, s[100];

void init(int k)

{s[k]=0;}

int succesor(int k)

{if (s[k]<n)

{s[k]=s[k]+1;

return 1;}

else return 0; }

int valid(int k)

{for(int i=1; i<k; i++)

if(k>1&&s[k]%2==s[k-1]%2)

return 0;

return 1;}

int solutie(int k)

{ return k==n;}

void tipar()

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

cout<<s[i]<<" ";

cout<<endl; }

void bt(int k)

{

init(k);

while(succesor(k))

if(valid(k))

if(solutie(k))

tipar();

else bt(k+1);}

int main()

{

cout<<"n="; cin>>n;

bt(1);

}

**3. Generare aranjamente de n elemente luate cate m:**

#include <iostream>

using namespace std;

int n, s[100], m;

void init(int k)

{s[k]=0;}

int succesor(int k)

{

if (s[k]<n)

{

s[k]=s[k]+1;

return 1;}

else return 0;}

int valid(int k)

{for(int i=1; i<k; i++)

if(s[k]==s[i])

return 0;

return 1;}

int solutie(int k)

{return k==m;}

void tipar()

{ for(int i=1; i<=m; i++)

cout<<s[i]<<" ";

cout<<endl;}

void bt(int k)

{ init(k);

while(succesor(k))

if(valid(k))

if(solutie(k))

tipar();

else bt(k+1);}

int main()

{ cout<<"m="; cin>>m;

cout<<"n="; cin>>n;

bt(1); }

**4. Generarea produsului cartezian**

#include <iostream>

using namespace std;

int n, s[100], w[100];

void init(int k)

{s[k]=0;}

int succesor(int k)

{

if (s[k]<w[k])

{s[k]=s[k]+1;

return 1;}

else return 0;}

int valid(int k)

{return 1;}

int solutie(int k)

{return k==n;}

void tipar()

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

cout<<s[i]<<" ";

cout<<endl;}

void bt(int k)

{

init(k);

while(succesor(k))

if(valid(k))

if(solutie(k))

tipar();

else bt(k+1);}

int main()

{

cout<<"n="; cin>>n;

for(int k=1; k<=n; k++)

{cin>>w[k];}

bt(1);

}

**6. Problema celor n dame:**

#include <iostream>

#include<cmath>

using namespace std;

int n,s[100];

void init (int k)

{s[k]=0;}

int succesor(int k)

{if(s[k]<n)

{s[k]=s[k]+1;return 1;}

else return 0;

}

int valid(int k)

{for(int i=1;i<k;i++)

if ((s[k]==s[i])||(abs(s[k]-s[i])==k-i))return 0;

return 1;

}

int solutie(int k)

{return k==n;}

void tipar()

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

cout<<s[i]<<" ";

cout<<endl;}

void bt(int k)

{init (k);

while(succesor(k))

if(valid(k))

if(solutie(k)) tipar();

else bt(k+1);

}

int main()

{cout<<"n=";cin>>n;

bt(1);}

**7. Se considera sirul cifrelor de la 0 la 9 sa se genereze toate nr de n cifre (n>=3) care nu contin 3 cifre pare sau 3 cifre impare alaturate.**

#include <iostream>

using namespace std;

int n, s[9];

void (init)(int k)

{ if (k==1)

s[k]=0;

else

s[k]=-1;}

int succesor(int k)

{ if (s[k]<9)

{s[k]++; return 1;}

else return 0;}

int valid (int k)

{ if(k>=3)

if((s[k]%2==s[k-1]%2)&&(s[k]%2==s[k-2]))

return 0;}

int solutie(int k)

{ return k==n;}

void tipar()

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

cout<<s[i]<<" ";

cout<<endl; }

void bt(int k)

{ init (k);

while (succesor(k))

if (valid(k) )

if(solutie (k))

tipar ();

else bt(k+1);}

int main() {cout<<"n= "; cin>>n; bt(1);}

**8. Se citeste de la tastatura un cuvant de max 10 litere mici distincte:**

1. **Sa se afiseze toate anagramele cuvantului citit:**

#include <iostream>

#include <string.h>

using namespace std;

int n , s[11];

char sir[11];

void init(int k)

{s[k]=-1;}

int succesor(int k)

{if(s[k]<n)

{s[k]=s[k]+1;return 1;}

else return 0;}

int valid(int k)

{for(int i=1;i<k;i++)

if(s[k]==s[i] )return 0;

return 1;

}

int solutie(int k)

{return k==n;}

void tipar()

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

cout<<sir[s[i]]<<" ";

cout<<endl;

}

void bt(int k)

{init (k);

while(succesor(k))

if(valid(k))

if(solutie (k)) tipar();

else bt(k+1);}

int main()

{cin>>sir;

n=strlen(sir)-1;

bt(1);

}

1. **Se citesc de la tastatura toate anagramele cuvantului citit care au proprietatea ca nu contin 2 vocale alaturate. Daca nu se pot forma anagrame se va afisa “imposibil”:**

#include <iostream>

#include <string.h>

using namespace std;

int n , s[11];

char sir[11];

void init(int k)

{s[k]=-1;}

int succesor(int k)

{if(s[k]<n)

{s[k]=s[k]+1;return 1;}

else return 0;}

int valid(int k)

{for(int i=1;i<k;i++)

if(s[k]==s[i] )return 0;

return 1;

}

int solutie(int k)

{return k==n;}

void tipar()

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

cout<<sir[s[i]]<<" ";

cout<<endl;

}

void bt(int k)

{init (k);

while(succesor(k))

if(valid(k))

if(solutie (k)) tipar();

else bt(k+1);}

int main()

{cin>>sir;

n=strlen(sir)-1;

bt(1);

}

**9. Se citesc din fisierul fazanin.txt numerele naturale n, si m (n<=15, m<=n) si apoi n cuvinte distincte cu cel mult 10 litere fiecare. Sa se afiseze in fisierul fazanout.txt toate secvente de cate m cuvinte dintre**

**cele citite care sa respecte conditiile jocului "fazan".**

**Exemplu:**

**fazanin.txt**

**8 3**

**paul alina asfalt nas ultim imagine nasture real**

**fazanout.txtultim imagine**

**alina nas asfalt**

**alina nasture real**

**nasture real alina**

**real alina nas**

**real alina nasture**

**paul**

#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

int n, m, s[11];

char cuvinte[15][11];

void init(int k) {

s[k] = -1;

}

int succesor(int k) {

if (s[k] < n - 1) {

s[k] = s[k] + 1;

return 1;

} else {

return 0;

}

}

int valid(int k) {

for (int i = 1; i < k; i++) {

if (s[k] == s[i]) {

return 0;

}

}

if (k > 1 && cuvinte[s[k - 1]][strlen(cuvinte[s[k - 1]]) - 1] != cuvinte[s[k]][0]) {

return 0;

}

return 1;

}

int solutie(int k) {

return k == m;

}

void tipar() {

for (int i = 1; i <= m; i++) {

cout << cuvinte[s[i]] << " ";

}

cout << endl;

}

void bt(int k) {

init(k);

while (succesor(k)) {

if (valid(k)) {

if (solutie(k)) {

tipar();

} else {

bt(k + 1);

}

}

}

}

int main() {

ifstream fin("fazanin.txt");

fin >> n >> m;

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

fin >> cuvinte[i];

}

fin.close();

bt(1);

return 0;

}