Liceul Teoretic „TRAIAN”

***JOC SCRABBLE***

LUCRARE PENTRU ATESTAREA

COMPETENŢELOR PROFESIONALE

Elev: LARION LOREDANA

Clasa: a XII-a B

Profesor coordonator: BARBU CRISTINA

2016 – 2017

***CUPRINS***

1. Motivaţia alegerii temei, utilitatea aplicaţiei

2. Structura aplicaţiei: organizarea conţinutului informaţional, structuri de date utilizate

3. Detalii tehnice de implementare

4. Resurse hard şi soft necesare

5. Modalităţi de utilizare

6. Posibilităţi de dezvoltare

***1.MOTIVAŢIA ALEGERII TEMEI, UTILITATEA APLICAŢIEI***

Motivul alegerii acestei teme a fost plăcerea de a juca SCRABBLE. Acest joc îmi amintește de copilarie și de momentele frumoase petrecute cu familia. Totodată acest joc te ajută la dezvoltarea vocabularului, fiind nevoit sa găsești cuvinte cât mai lungi și mai elevate pentru a câștiga în final.

Aplicația este utilă tuturor utilizatorilor de calculator pentru a se relaxa și în același timp pentru a se dezvolta pe planul comunicării și lexicului.

***Un scurt istoric al jocului și regulile acestuia***

SCRABBLE este un joc de litere între doi sau mai multi jucãtori (numãrul jucãtorilor este nelimitat la unele genuri de partidã). De regulã, scopul unui jucãtor este realizarea unui punctaj cât mai mare prin plasarea pe o tablã de joc a diverse cuvinte, cu ajutorul unor jetoane marcate cu litere.

Din anul 1931, când jocul a fost inventat de americanul Alfred Butts, scrabble a evoluat de la un joc distractiv spre un sport competitional. Aceste regulamente se referã la jocul competitional, practicat în întrecerile oficiale organizate sub egida Federatiei Române de Scrabble sau a unor organisme internationale la care F.R. Scrable este afiliatã sau pe care le acceptã ca autoritãti în domeniu. Regulamentele de mai jos sunt recomandate tuturor practicantilor jocului SCRABBLE.

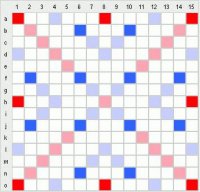
*Recuzita jocului*

Un joc de scrabble se compune din:

- O tablã de joc (din lemn, carton, plastic etc.) pe care este desenat un careu de 15 x 15 pãtrãtele, fiecare pãtrãtel având latura de cca 2 cm. Liniile si coloanele tablei sunt marcate pe margini astfel:

• Liniile sunt marcate pe marginea din stânga, cu litere de tipar, de la 'a' la 'o', începând de sus în jos, în ordine alfabeticã;

• Coloanele sunt marcate pe marginea de sus, cu numere de la 1 la 15, de la stânga la dreapta



Unele dintre pãtrãtelele tablei sunt marcate distinct (colorate). Acestea indicã multiplicarea valorii literelor sau a cuvintelor asezate pe ele, conform explicatiilor din figurã. Dacã un cuvânt acoperã douã sau mai multe astfel de pãtrãtele, proprietãtile lor se cumuleazã; de exemplu, o literã poate fi dublatã si întregul cuvânt triplat de douã ori (adicã multiplicat cu 9). Multiplicarea unei litere sau a unui cuvânt se face numai atunci când litera se plaseazã pe pãtrãtelul respectiv si ea opereazã asupra tuturor cuvintelor formate cu acea literã, la acea depunere.

Exemple de calcul al punctajului

- Un set de 100 jetoane (din lemn, carton, plastic etc.) care sã se suprapunã peste pãtrãtelele tablei. Aceste jetoane au înscrisã pe o parte o literã din alfabet (cu caractere mari, de tipar), fiecare literã având în partea din dreapta-jos un numãr care îi indicã valoarea. Douã dintre jetoane - numite jokeri - sunt nescrise (sau au un simbol caracteristic, ex.: J ), putând fi folosite ca reprezentând oricare literã din set sau literele K, Q, Y si W. Valoarea jokerului (jetonul alb sau marcat cu '?') este zero. Dupã ce jokerul a fost plasat pe tablã ca reprezentând o literã, semnificatia lui nu se mai schimbã pânã la sfârsitul jocului. Jokerul are aceleasi proprietãti ca si celelalte litere.

Numãrul de jetoane pentru fiecare literã si valoarea literelor la scrabble în limba românã

(exemplu de citire a tabelului de mai jos: 11 jetoane reprezentând litera A, fiecare cu valoarea 1... etc.)

11 • A1 2 • G9 6 • N1 6 • U1

2 • B9 1 • H10 5 • O1 2 • V8

5 • C1 10 • I1 4 • P2 1 • X10

4 • D2 1 • J10 7 • R1 1 • Z10

9 • E1 4 • L1 5 • S1 2 • Jo (jokeri)

2 • F8 3 • M4 7 • T1 Total: 100 de jetoane

- Un sãculet opac de pânzã pentru pãstrarea si extragerea literelor.

- Douã suporturi pe care se etaleazã literele. Pe un suport trebuie sã poatã fi manevrate usor 7 litere. Constructia suportului trebuie sã permitã ca partenerul sã poatã observa câte litere sunt plasate pe el.

- Ceas de control tip sah.

In competitiile oficiale organizate de federatie sau cu acordul federatiei, se admite Fondul de cuvinte stabilit de Comisia Centralã de Regulamente. In alte competitii, organizatorii pot stabili alt fond de cuvinte admise. La competitiile oficiale în limba românã si în limba englezã, jucãtorii trebuie sã se prezinte cu un joc complet de scrabble si cu un ceas de control.

*Reguli generale*

În timpul jocului cuvintele se pot forma si lega între ele astfel: Exemple de formare si legare a cuvintelor

• Prin lungire, adaugând una sau mai multe litere în fata sau / si în urma unui cuvânt aflat deja pe tablã;

• Prin intersectare, formând un cuvânt (orizontal sau vertical) care sã se încruciseze cu un altul aflat deja pe tablã;

• Prin alipire;

• Prin metode combinate ale celor de mai sus.

*Reguli fundamentale*

• Primul cuvânt, care se depune pe tabla goalã, trebuie sã fie format din cel putin 2 litere si cel mult 7 litere si trebuie sã treacã prin pãtrãtelul din centrul tablei (h8).

• La o depunere se pot plasa pe tablã între una si 7 litere, cu exceptia primei depuneri (vezi regula de mai sus). În nici un caz, nici mãcar la problemele de compunere, nu se admite plasarea pe tablã a mai mult de 7 litere la o depunere.

• Toate literele utilizate în cadrul unei depuneri se plaseazã numai orizontal (de la stânga la dreapta) sau numai vertical (de sus în jos), pe o singurã linie sau coloanã.

• Nu se pot depune cuvinte izolate, care sã nu aibã legaturã cu un alt cuvânt de pe tablã. Toate literele unei depuneri trebuie sã fie legate între ele (eventual, prin alte litere de pe tablã).

• Dacã se depune un joker, se va specifica ce literã înlocuieste, iar semnificatia acestui joker nu se mai schimbã pânã la sfârsitul partidei.

• O depunere este corectã dacã sunt respectate regulile fundamentale de mai sus, dacã toate cuvintele nou-formate pe tablã sunt acceptate de partener sau de arbitru ca fãcând parte din fondul de cuvinte admis si dacã jucãtorul s-a încadrat în timpul de gândire acordat.

*Alte reguli generale*

Punctajul unei depuneri se calculeazã însumând valorile literelor componente ale cuvântului format (tinând cont de multiplicarea cuvântului, dacã este cazul). Se adaugã si punctajul cuvintelor adiacente nou formate, care se calculeazã la fel ca si pentru cuvântul principal. Dacã în cadrul unei depuneri se plaseazã toate cele 7 litere, la calcularea punctajului se adaugã o primã de 50 de puncte (depunere - scrabble).

Exemple de calcul al punctajului

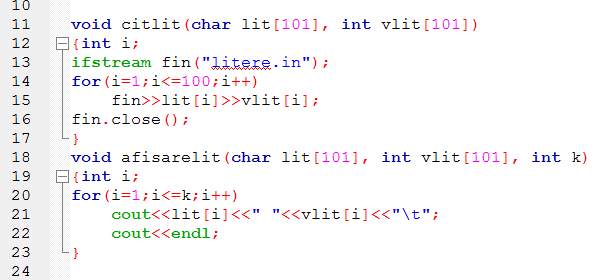
Pentru notatie pe buletinul de joc, pozitia de plasare a unui cuvânt se precizeazã indicând coordonatele pãtrãtelului în care se aflã prima literã a cuvântului, astfel:

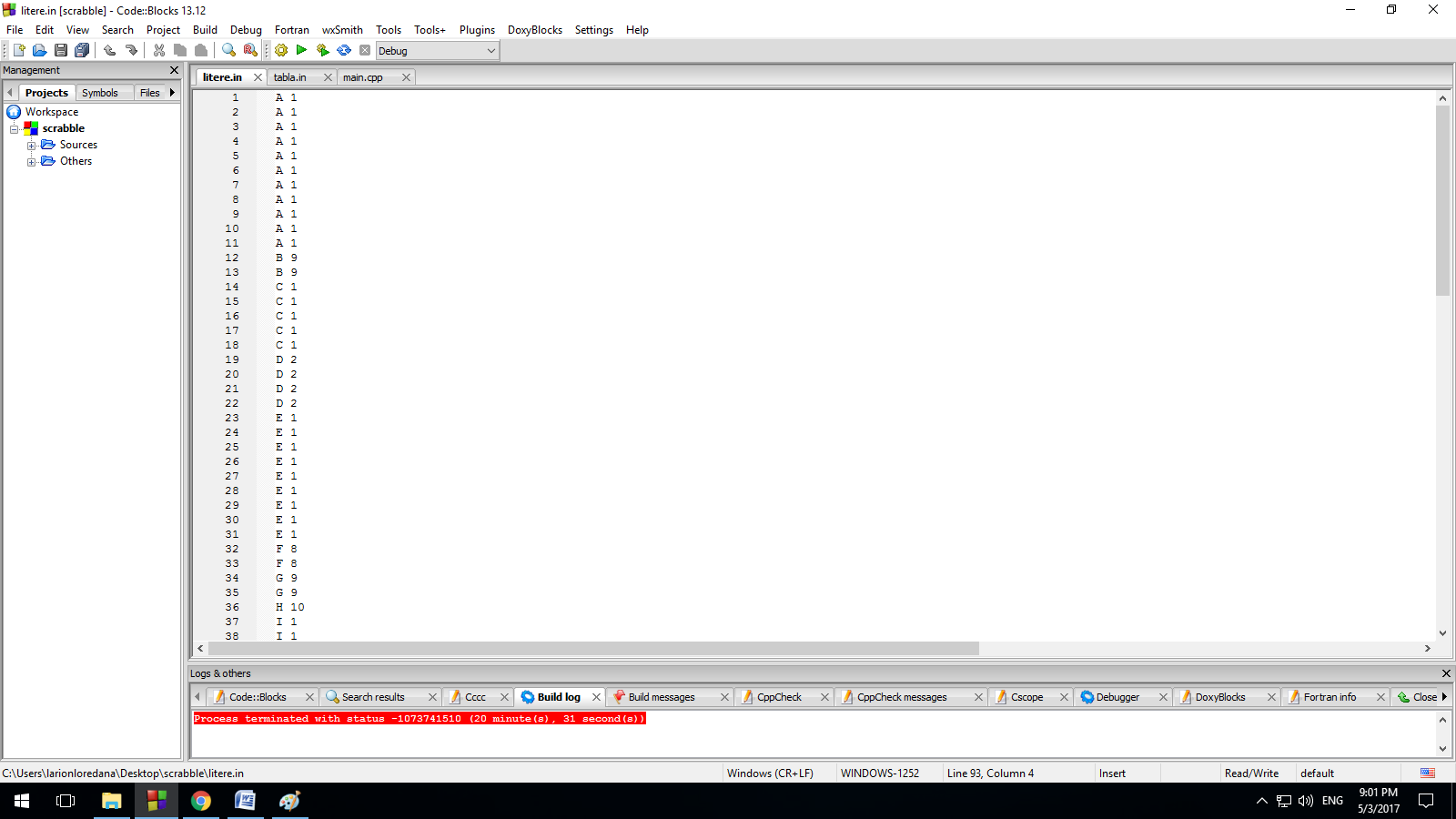
* Dacã un cuvânt este plasat orizontal, se indicã întâi litera, apoi numãrul (ex.: h8).
* Dacã un cuvânt este plasat vertical, se indicã întâi numãrul, apoi litera (ex.: 8h).

***2.STRUCTURA APLICAŢIEI: ORGANIZAREA CONŢINUTULUI INFORMAŢIONAL, STRUCTURI DE DATE UTILIZATE***

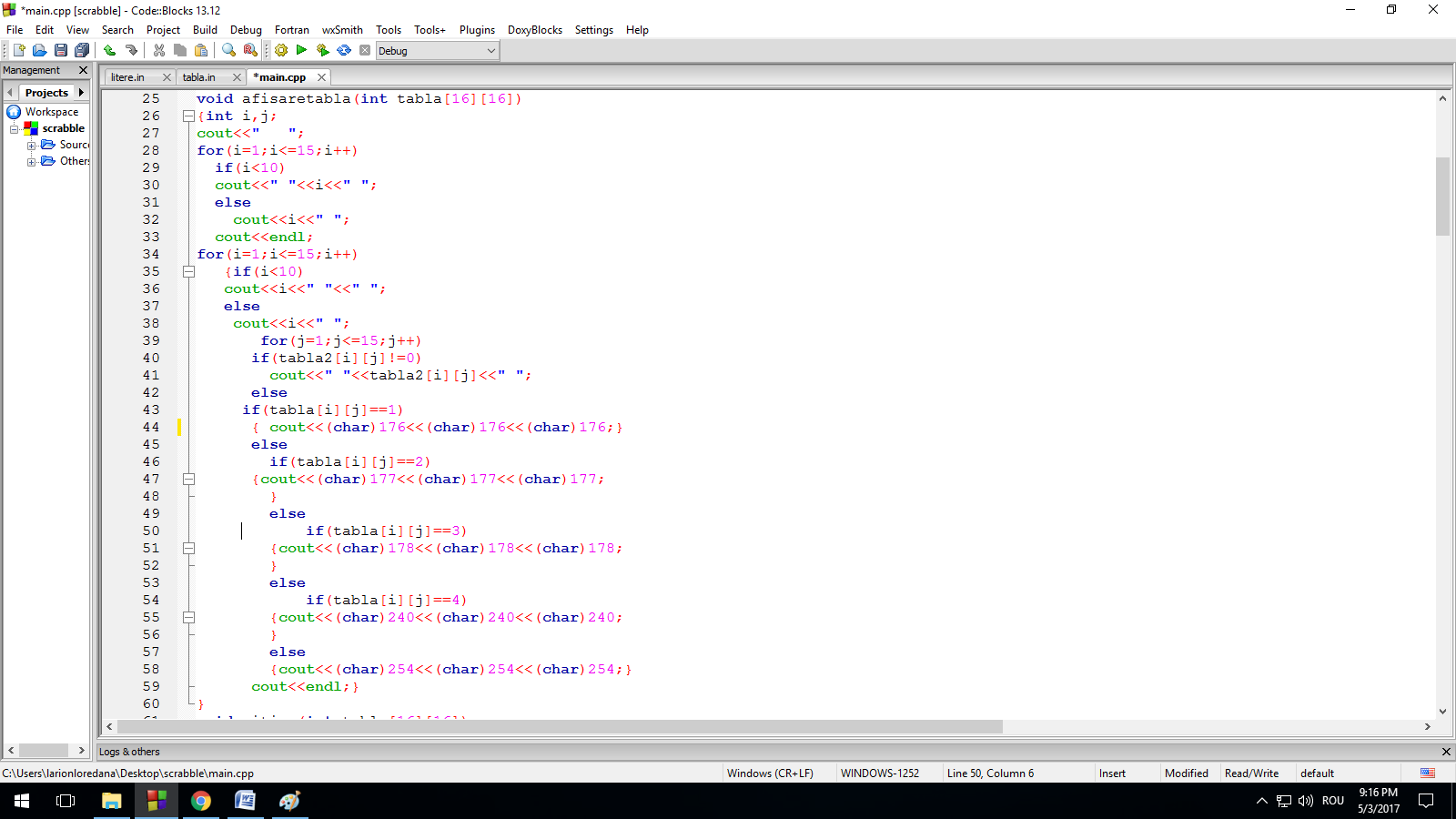
Programul jocului este alcătuit din mai multe subprogame ce sunt apelate în final, în funcția main

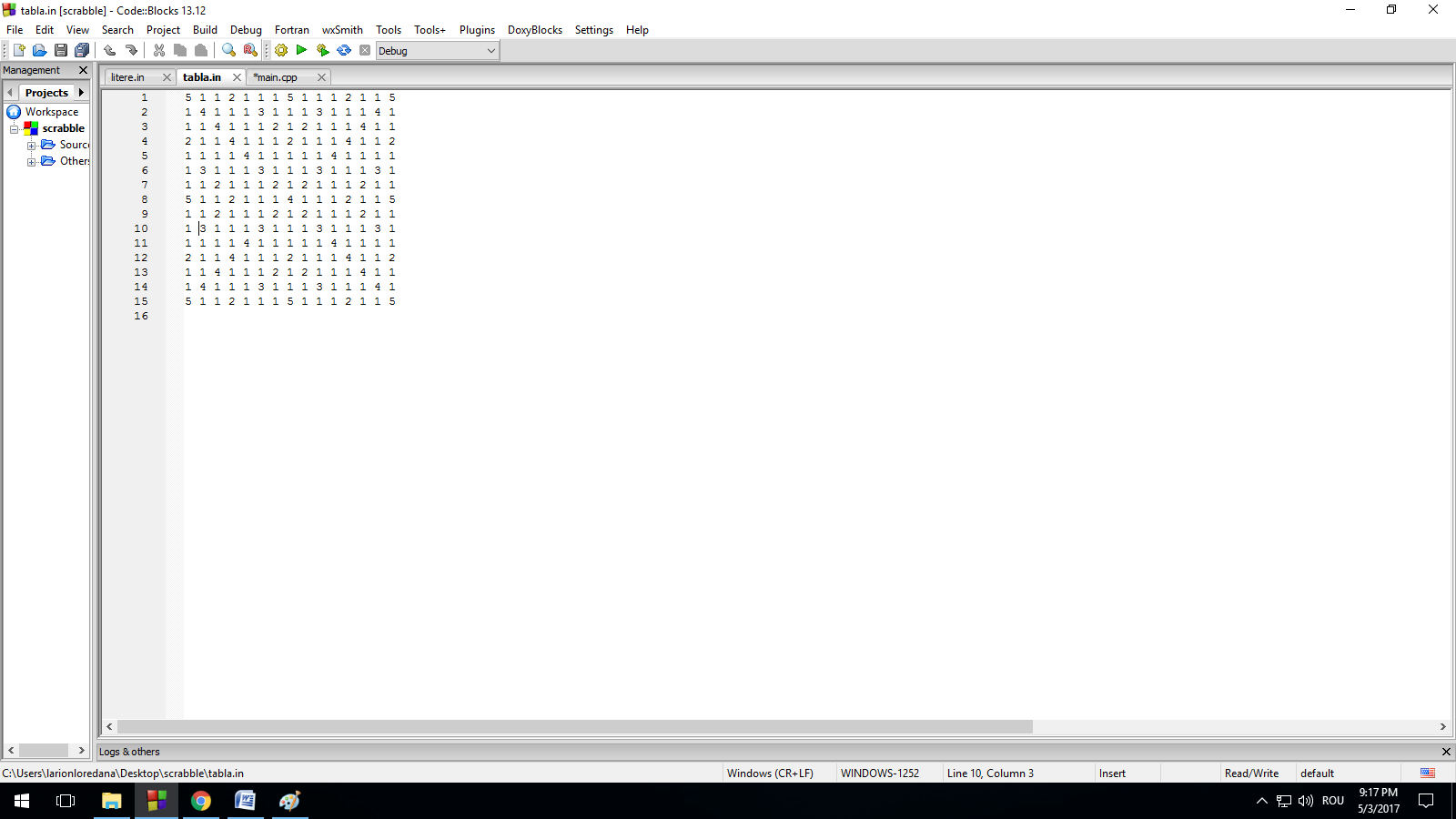
* Citirea literelor din fișierul “litere.in” prin subprogramul *citlit* și afișarea acestora pe ecran prin subprogramul *afisarelit*:



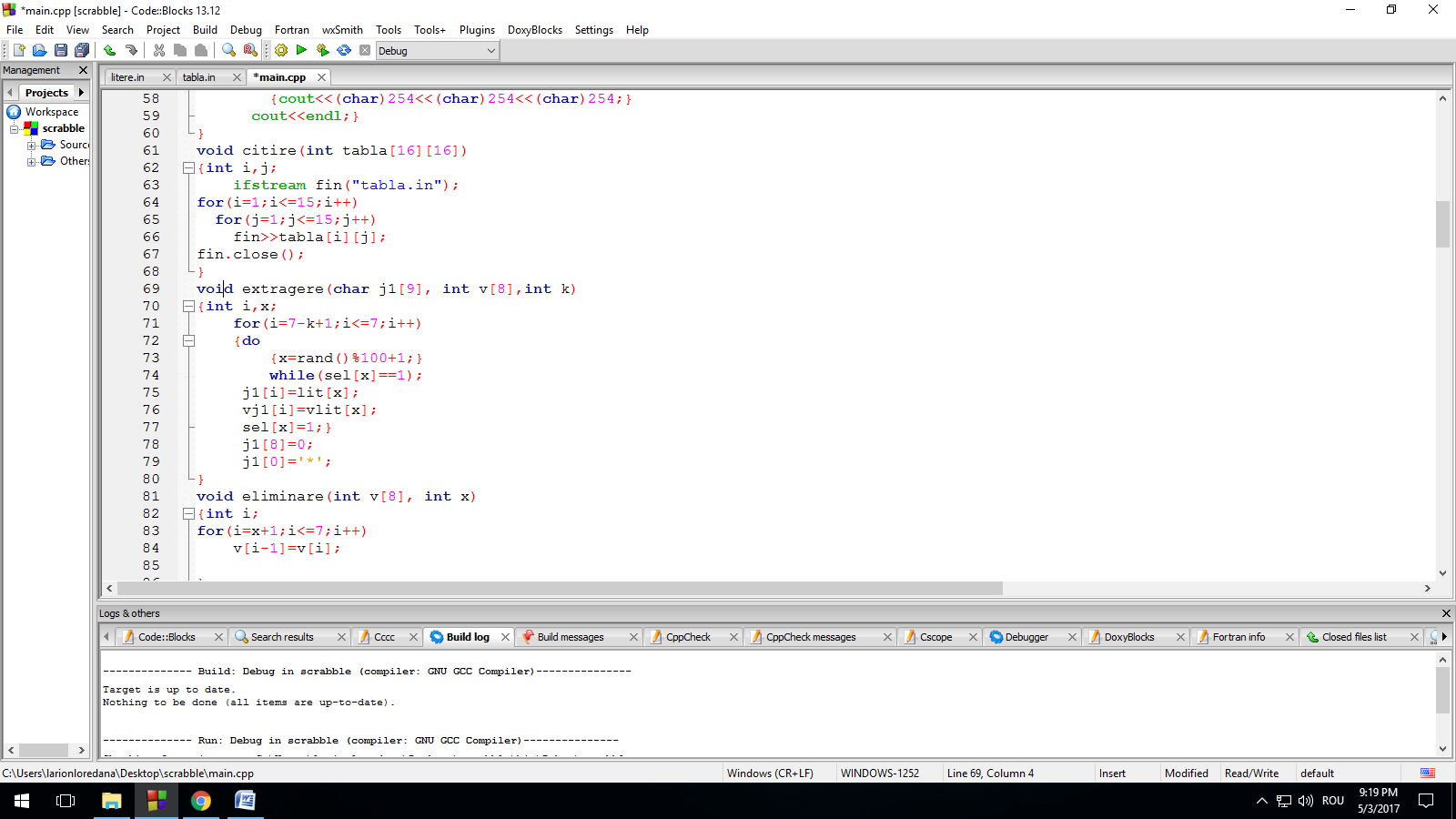


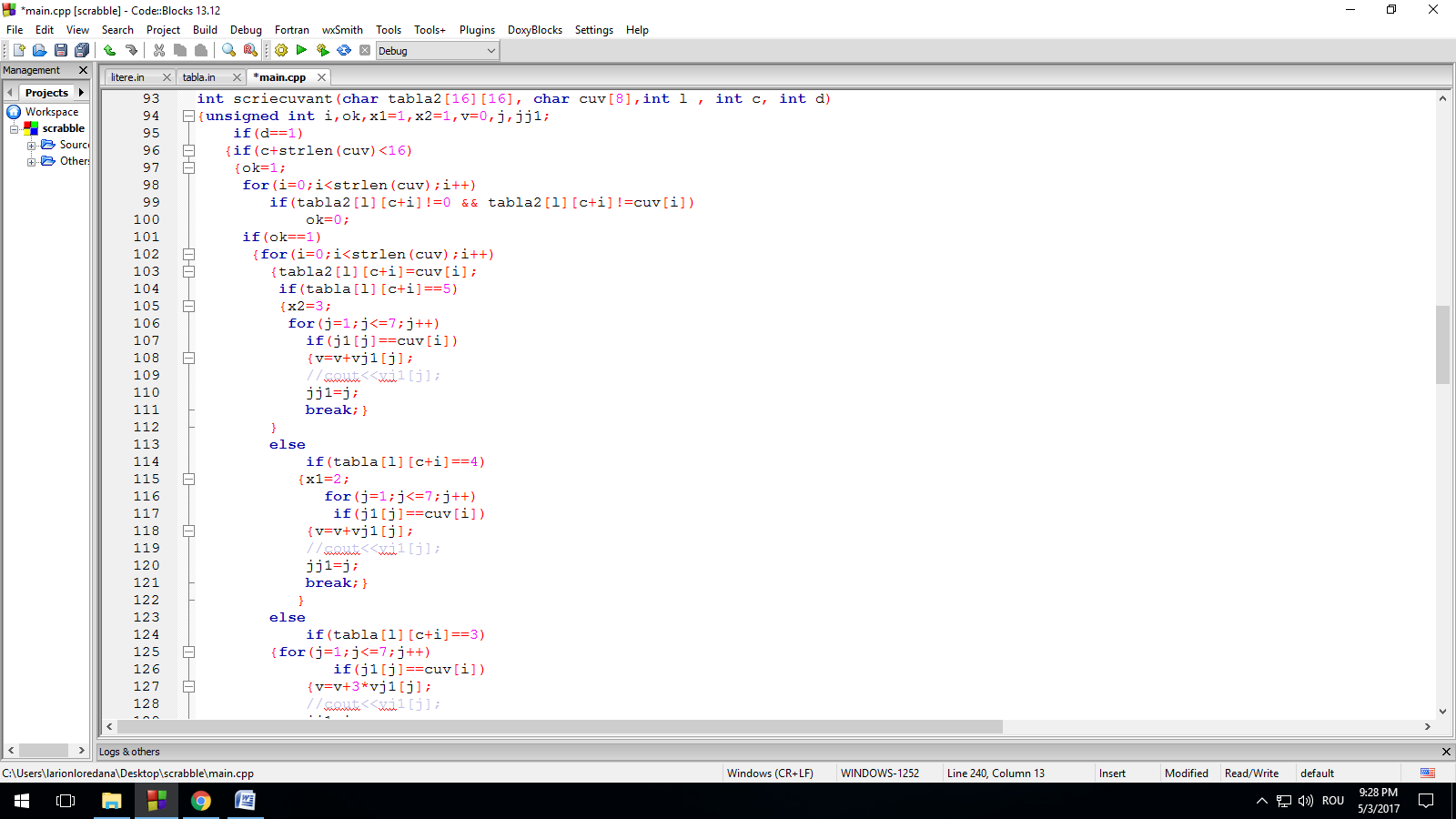
* Afișarea tablei,în conformitate cu forma ei reală, în loc de culori, am folosit diferite forme de pătrate, fiecare având câte o însemnătate în calcularea punctajului(tabla se citește din fișier):
* tabla[i][j]==1 ->nu se întâmplă nimic cu punctajul
* tabla[i][j]==2 ->se dublează valoarea literei
* tabla[i][j]==3 ->se triplează valoarea literei
* tabla[i][j]==4 ->se dublează valoarea cuvântului
* tabla[i][j]==5 ->se triplează valoarea cuvântului

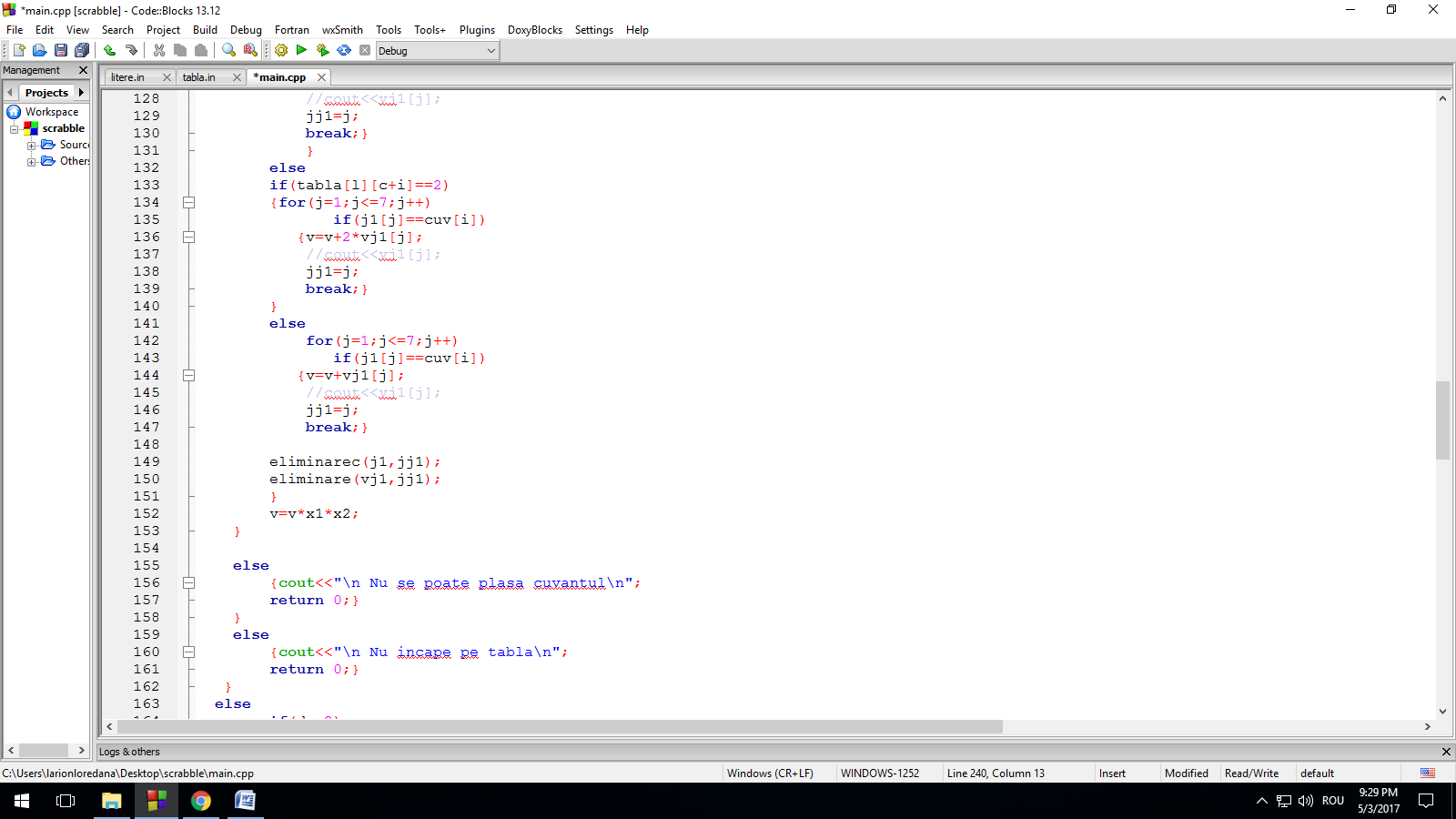


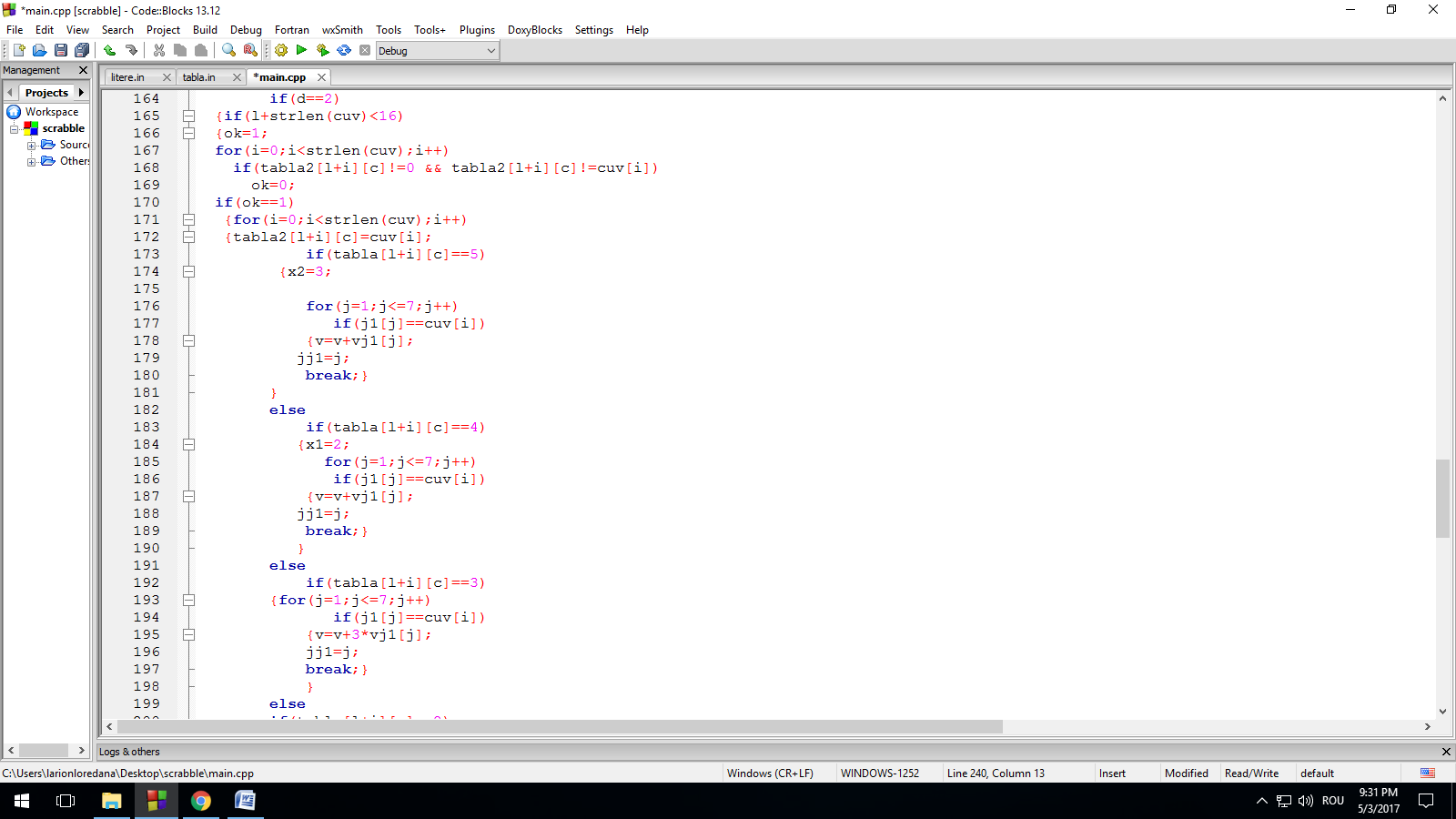


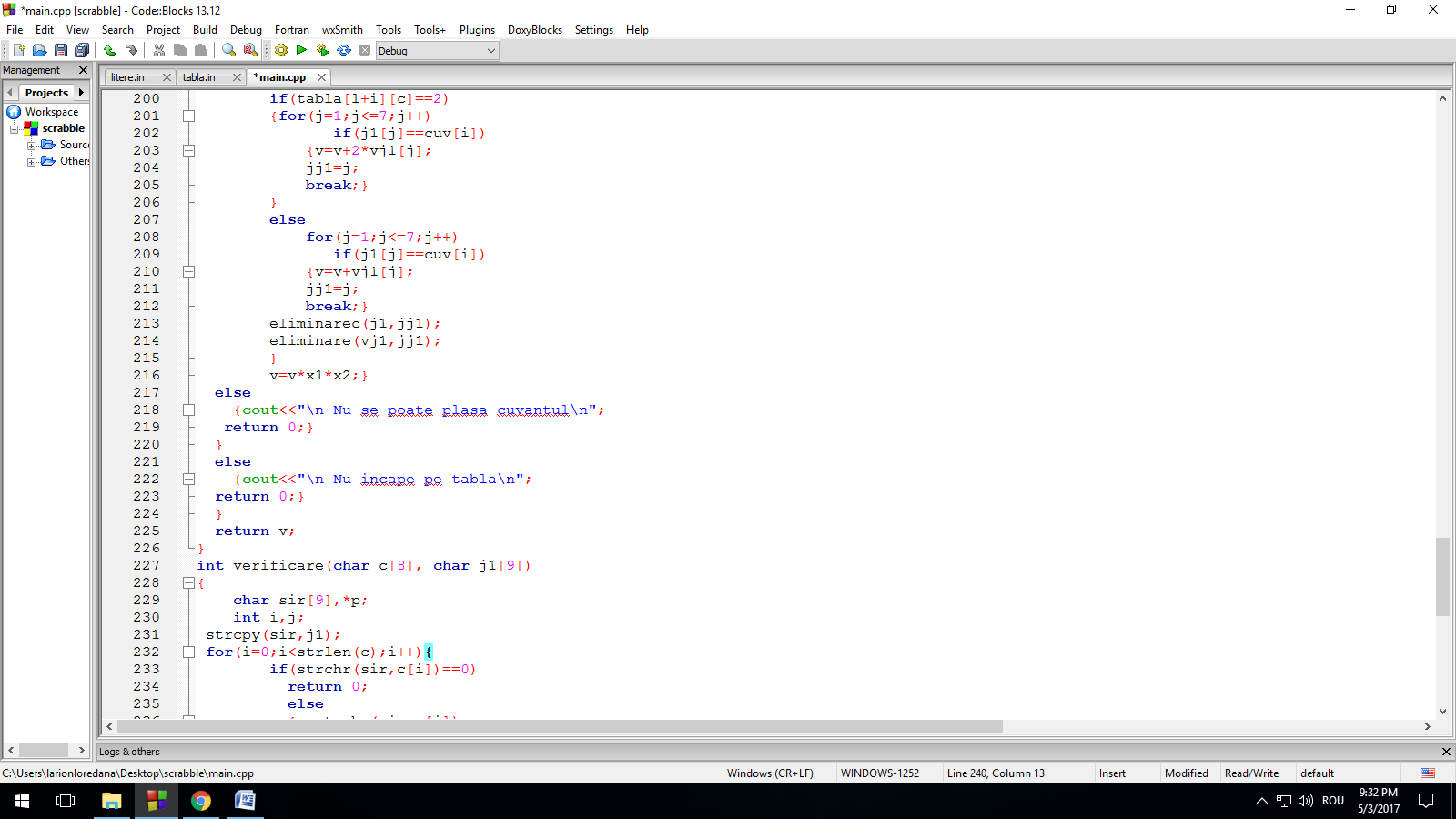
* Extragerea literelor:



* Subprogramul *scrierecuvant* îndeplinește mai multe funcții:
* Scrie cuvântul pe tablă
* Verifică daca cuvântul se încadrează pe tablă (afișează un mesaj corespunzător)
* Verifică dacă este format din literele puse la dispoziție,dacă nu îmi spune să introduc dinou cuvântul
* Verifică daca cuvintele se leagă între ele printr-o literă(dacă nu, afișează un mesaj corespunzător)
* Calculează punctajul în funcție de unde sunt plasate literele
* Generează următoarele litere (le șterge pe cele folosite deja în formarea cuvântului afișat pe tablă)







***3.DETALII TEHNICE DE IMPLEMENTARE***

#include <iostream>

#include<fstream>

#include<cstdlib>

#include<ctime>

#include<cstring>

using namespace std;

char lit[101],j1[9],tabla2[16][16];

int vlit[101],sel[101],vj1[8],mult[16][16],tabla[16][16];

int vlit2[23]={1,9,1,2,1,8,9,10,1,10,1,4,1,1,2,1,1,1,1,8,10,10,0};

void citlit(char lit[101], int vlit[101])

{int i;

ifstream fin("litere.in");

for(i=1;i<=100;i++)

fin>>lit[i]>>vlit[i];

fin.close();

}

void afisarelit(char lit[101], int vlit[101], int k)

{int i;

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

cout<<lit[i]<<" "<<vlit[i]<<"\t";

cout<<endl;

}

void afisaretabla(int tabla[16][16])

{int i,j;

cout<<" ";

for(i=1;i<=15;i++)

if(i<10)

cout<<" "<<i<<" ";

else

cout<<i<<" ";

cout<<endl;

for(i=1;i<=15;i++)

{if(i<10)

cout<<i<<" "<<" ";

else

cout<<i<<" ";

for(j=1;j<=15;j++)

if(tabla2[i][j]!=0)

cout<<" "<<tabla2[i][j]<<" ";

else

if(tabla[i][j]==1)

{ cout<<(char)176<<(char)176<<(char)176;

}

else

if(tabla[i][j]==2)

{cout<<(char)177<<(char)177<<(char)177;

}

else

if(tabla[i][j]==3)

{cout<<(char)178<<(char)178<<(char)178;

}

else

if(tabla[i][j]==4)

{cout<<(char)240<<(char)240<<(char)240;

}

else

{cout<<(char)254<<(char)254<<(char)254;}

cout<<endl;}

}

void citire(int tabla[16][16])

{int i,j;

ifstream fin("tabla.in");

for(i=1;i<=15;i++)

for(j=1;j<=15;j++)

fin>>tabla[i][j];

fin.close();

}

void extragere(char j1[9], int v[8],int k)

{int i,x;

for(i=7-k+1;i<=7;i++)

{do

{x=rand()%100+1;}

while(sel[x]==1);

j1[i]=lit[x];

vj1[i]=vlit[x];

sel[x]=1;}

j1[8]=0;

j1[0]='\*';

}

void eliminare(int v[8], int x)

{int i;

for(i=x+1;i<=7;i++)

v[i-1]=v[i];

}

void eliminarec(char v[8], int x)

{int i;

for(i=x+1;i<=7;i++)

v[i-1]=v[i];

}

int scriecuvant(char tabla2[16][16], char cuv[8],int l , int c, int d)

{unsigned int i,ok,x1=1,x2=1,v=0,j,jj1;

if(d==1)

{if(c+strlen(cuv)<16)

{ok=1;

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

if(tabla2[l][c+i]!=0 && tabla2[l][c+i]!=cuv[i])

ok=0;

if(ok==1)

{for(i=0;i<strlen(cuv);i++)

{tabla2[l][c+i]=cuv[i];

if(tabla[l][c+i]==5)

{x2=3;

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

//cout<<vj1[j];

jj1=j;

break;}

}

else

if(tabla[l][c+i]==4)

{x1=2;

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

//cout<<vj1[j];

jj1=j;

break;}

}

else

if(tabla[l][c+i]==3)

{for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+3\*vj1[j];

//cout<<vj1[j];

jj1=j;

break;}

}

else

if(tabla[l][c+i]==2)

{for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+2\*vj1[j];

//cout<<vj1[j];

jj1=j;

break;}

}

else

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

//cout<<vj1[j];

jj1=j;

break;}

eliminarec(j1,jj1);

eliminare(vj1,jj1);

}

v=v\*x1\*x2;

}

else

{cout<<"\n Nu se poate plasa cuvantul\n";

return 0;}

}

else

{cout<<"\n Nu incape pe tabla\n";

return 0;}

}

else

if(d==2)

{if(l+strlen(cuv)<16)

{ok=1;

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

if(tabla2[l+i][c]!=0 && tabla2[l+i][c]!=cuv[i])

ok=0;

if(ok==1)

{for(i=0;i<strlen(cuv);i++)

{tabla2[l+i][c]=cuv[i];

if(tabla[l+i][c]==5)

{x2=3;

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

jj1=j;

break;}

}

else

if(tabla[l+i][c]==4)

{x1=2;

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

jj1=j;

break;}

}

else

if(tabla[l+i][c]==3)

{for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+3\*vj1[j];

jj1=j;

break;}

}

else

if(tabla[l+i][c]==2)

{for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+2\*vj1[j];

jj1=j;

break;}

}

else

for(j=1;j<=7;j++)

if(j1[j]==cuv[i])

{v=v+vj1[j];

jj1=j;

break;}

eliminarec(j1,jj1);

eliminare(vj1,jj1);

}

v=v\*x1\*x2;}

else

{cout<<"\n Nu se poate plasa cuvantul\n";

return 0;}

}

else

{cout<<"\n Nu incape pe tabla\n";

return 0;}

}

return v;

}

int verificare(char c[8], char j1[9])

{

char sir[9],\*p;

int i,j;

strcpy(sir,j1);

for(i=0;i<strlen(c);i++){

if(strchr(sir,c[i])==0)

return 0;

else

{p=strchr(sir,c[i]);

j=p-sir;

strcpy(sir+j,sir+j+1);

}}

return 1;

}

int main()

{int l,c,d,joc1=0,lc,p;

char cuvant[8];

srand (time(NULL));

citlit(lit,vlit);

extragere(j1,vj1,7);

//afisarelit(lit,vlit);

afisarelit(j1,vj1,7);

citire(tabla);

afisaretabla(tabla);

do{

do{cout<<"Introduceti cuvantul";

cin>>cuvant;}while(verificare(cuvant,j1)==0);

lc=strlen(cuvant);

cout<<"introduceti pozitia primei litere (linie coloana)";

cin>>l>>c;

cout<<"introduceti directia (1.orizontala 2.verticala)";

cin>>d;

p=scriecuvant(tabla2,cuvant,l ,c, d);

if(p>0)

{joc1=joc1+p;

afisaretabla(tabla);

cout<<"\njucatorul 1 are "<<joc1<<" puncte\n";

extragere(j1,vj1,lc);

}

afisarelit(j1,vj1,7);

afisaretabla(tabla);

}while(1);

return 0;

}

***4.RESURSE HARD ŞI SOFT NECESARE***

Siteul a fost realizat cu ajutorul programului **Code::Blocks.**

Resurse minime :

1) Hardware :

* Procesor (CPU) : 500 MHz +
* Memorie (RAM): 64 MB
* Spaţiu liber pe hard-disk : 30 MB
* Monitor :1024\*768

2) Software :

- Sistem de operare : Windows 98/XP

Resurse recomandate:

1) Hardware :

* Procesor (CPU) : 1000 MHz +
* Memorie (RAM): 128 MB
* Spaţiu liber pe hard-disk : 30 MB
* Monitor : VGA 32-bit color (1024x768)

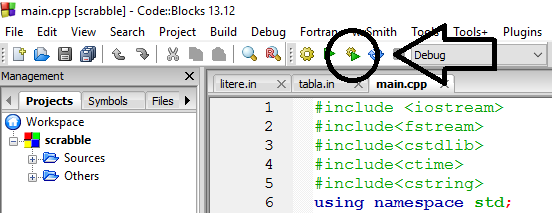
2) Software :

- Sistem de operare : Windows XP

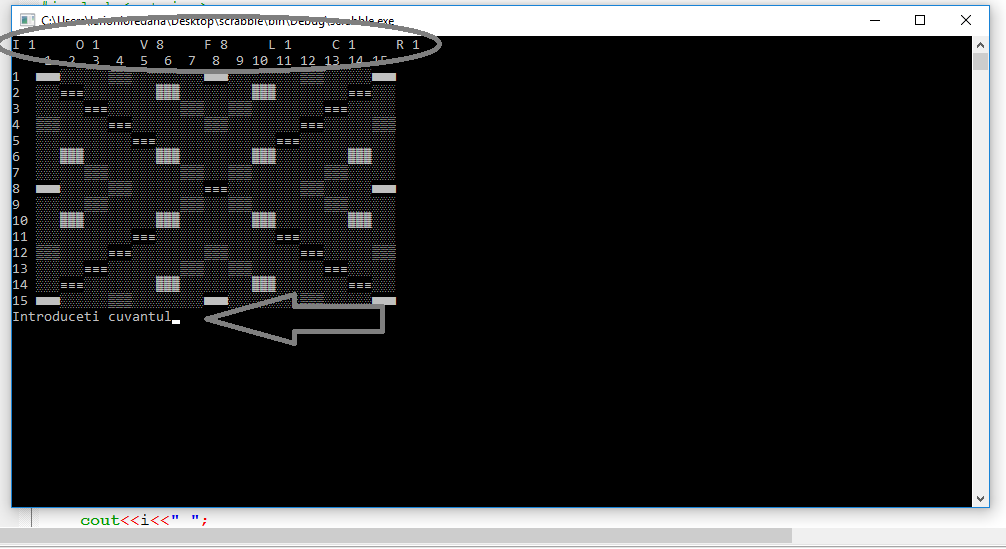
***5.MODALITĂŢI DE UTILIZARE***

Pentru a utiliza jocul avem nevoie de programul **Code::Blocks**.

Mai întâi rulăm programul apăsând butonul asociat acestei funcții:

******

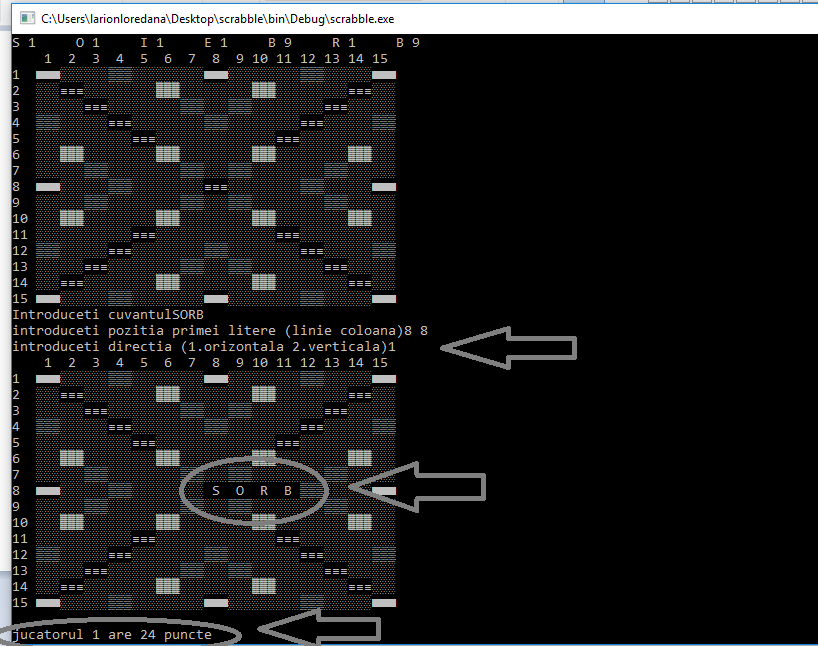
Introducem un cuvant format din 2-7 litere alese din cele procesate de program:



Alegem casuțele(linie,coloană) unde vrem să fie plasat



Apoi dacă sa fie plasat pe orizontală/verticală și în final ne plasează cuvantul în tabel și calculează punctajul jucătorului



***6.POSIBILITĂŢI DE DEZVOLTARE***

Jocul poate fii dezvoltat prin adăugarea a mai multor jucători, îmbunătățirea tablei (colorarea pătrățelelor) și în plus mai putem face și un tabel pentru reținerea punctajului fiecărui jucător.

***7.BIBLIOGRAFIE***

1. [www.csimpetus.ro](http://www.csimpetus.ro)
2. www.scribd.com